Learning Principal Directions: Integrated-Squared-Error Minimization

Jong-Hoon Ahn [a], Jong-Hoon Oh [a], Seungjin Choi [b,*]

[a] *Department of Physics*
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea

[b] *Department of Computer Science*
Pohang University of Science and Technology
San 31 Hyoja-dong, Nam-gu, Pohang 790-784, Korea

Abstract

A common derivation of principal component analysis (PCA) is based on the minimization of the squared-error between centered data and linear model, corresponding to the reconstruction error. In fact, minimizing the squared-error leads to principal subspace analysis where *scaled* and *rotated* principal axes of a set of observed data, are estimated. In this paper, we introduce and investigate an alternative error measure, *integrated-squared-error* (ISE), the minimization of which determines the exact principal axes (without rotational ambiguity) of a set of observed data. We show that exact principal directions emerge from the minimization of ISE. We present a simple EM algorithm, 'EM-ePCA', which is similar to EM-PCA [9], but finds exact principal directions without rotational ambiguity. In addition, we revisit the generalized Hebbian algorithm (GHA) and show that it emerges from the integrated-squared-error minimization in a single-layer linear feedforward neural network.

Key words: EM algorithm, Generalized Hebbian algorithm, Generative models, Probabilistic coupled models, Separable LS, PCA

* Corresponding author. Tel.: +82-54-279-2259; Fax: +82-54-279-2299
Email: jonghun@postech.ac.kr (J. -H. Ahn), jhoh@postech.ac.kr (J. -H. Oh), seungjin@postech.ac.kr (S. Choi)
URL: http://www.postech.ac.kr/∼seungjin (S. Choi)

1 Introduction

Principal component analysis (PCA) is a widely-used linear dimensionality reduction technique [4, 6]. A common derivation of PCA is in terms of a linear orthogonal projection that minimizes the squared reconstruction error. Principal axes of a set of observed variables can also be determined through maximum likelihood estimation of parameters in a latent variable model. Along this line, probabilistic PCA (PPCA) [11] and EM-PCA [9] were proposed. However, these methods find the scaled and *rotated* principal axes (principal subspace analysis rather than PCA), hence some post-processing is required to compute exact principal directions (which corresponds to the orthogonal eigenvectors of the data covariance matrix)

In this paper we introduce and analyze an alternative error measure, *integrated-squared-error* (ISE), where squared-errors are coupled in a certain way. The ISE was motivated from a coupled linear generative model that was introduced in our earlier work [2, 1] where ISE was referred to as generalized squared error and preliminary result was provided. In this paper we focus on ISE, rather than the coupled generative model and provide further analysis of ISE, showing that the minimization of ISE leads to exact principal directions of a set of observed data without rotational ambiguity, in contrast to PPCA and EM-PCA.

We also present a simple EM algorithm in the context of separable LS, referred to as 'EM-ePCA' where ePCA was used to emphasize the fact that the algorithm finds exact principal directions without rotational ambiguity. The EM-ePCA algorithm has updating rules which are similar to EM-PCA in [9], but upper and lower operators (that will be illustrated in Sec. 4) distinguish the behavior of EM-ePCA from EM-PCA, enforcing EM-ePCA to converge to exact principal directions. In addition, we revisit the generalized Hebbian algorithm (GHA) [10] and show that the minimization of the integrated-squared-error using the gradient descent method leads to the GHA.

The rest of this paper is organized as follows. Next section briefly overviews probabilistic PCA and EM-PCA, providing background on PCA to readers. Sec. 3 introduces and analyzes ISE, providing our main theorem where we prove that the minimization of ISE leads to exact principal directions of a set of observed data. A simple EM algorithm, EM-ePCA, is presented in Sec. 4 in the framework of a separable LS. A connection with the generalized Hebbian algorithm is also discussed. A link with a probabilistic coupled model is discussed in Sec. 5. Simple numerical experiments are shown in Sec. 6 and conclusion is drawn in Sec. 7

2 Probabilistic PCA and EM-PCA

The probabilistic PCA (PPCA) [12, 11] considers a linear generative model which assumes that the observed data $\boldsymbol{x} \in \mathbb{R}^m$ is generated by

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

where the parameter matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains the factor loadings, and the latent variables $\boldsymbol{s} \in \mathbb{R}^n$ have a unit isotropic Gaussian distribution with zero mean ($m \geq n$). The noise $\boldsymbol{v}$ is also isotropic Gaussian, $\boldsymbol{v} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$.

It was shown in [11] that the maximum likelihood estimator $\boldsymbol{A}_{ML}$ is the matrix whose columns are the *scaled and rotated principal eigenvectors* of the sample covariance matrix of the data, even when the covariance model is approximate. The maximum likelihood estimator $\boldsymbol{A}_{ML}$ is given by $\boldsymbol{A}_{ML} = \boldsymbol{U}_n(\boldsymbol{\Lambda}_n - \sigma^2 \boldsymbol{I})^{1/2}\boldsymbol{R}$, where $\boldsymbol{U}_n \in \mathbb{R}^{m \times n}$ contains n eigenvectors of the sample covariance matrix of the observed data with corresponding eigenvalues in the diagonal matrix $\boldsymbol{\Lambda}_n \in \mathbb{R}^{n \times n}$, $\boldsymbol{R} \in \mathbb{R}^{n \times n}$ is an arbitrary orthogonal rotation matrix, and $\boldsymbol{I}$ is the $m \times m$ identity matrix. The true principal axes can be recovered when the columns of $\boldsymbol{R}^T$ are equal to the eigenvectors of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ matrix.

PCA can be viewed as a limiting case of the linear Gaussian model (1) as the noise variance σ^2 becomes infinitesimally small. Along this line, the EM-PCA algorithm was derived by taking zero noise limit into account [9]. In the case of zero noise limit, the linear generative model (see Fig. 1) can be rewritten as $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$ where the centered data matrix $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ is defined by

$$\boldsymbol{X} = \left[\boldsymbol{x}_{(1)}, \ldots, \boldsymbol{x}_{(N)}\right],$$

and $\boldsymbol{S} \in \mathbb{R}^{n \times N}$ is the latent variable matrix. The row vectors of $\boldsymbol{X}$ are denoted by $\{\vec{\boldsymbol{x}}_i\}$, $i = 1, \ldots, m$, i.e.,

$$\boldsymbol{X} = \left[\vec{\boldsymbol{x}}_1^T, \ldots, \vec{\boldsymbol{x}}_m^T\right]^T. \tag{2}$$

The same notation is applied to the latent variable matrix $\boldsymbol{S}$.

Algorithm Outline: EM-PCA [9]

E-step

$$\boldsymbol{S} = \left(\boldsymbol{A}^T \boldsymbol{A}\right)^{-1} \boldsymbol{A}^T \boldsymbol{X}. \tag{3}$$

M-step

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}\boldsymbol{S}^T \left(\boldsymbol{S}\boldsymbol{S}^T\right)^{-1}. \tag{4}$$

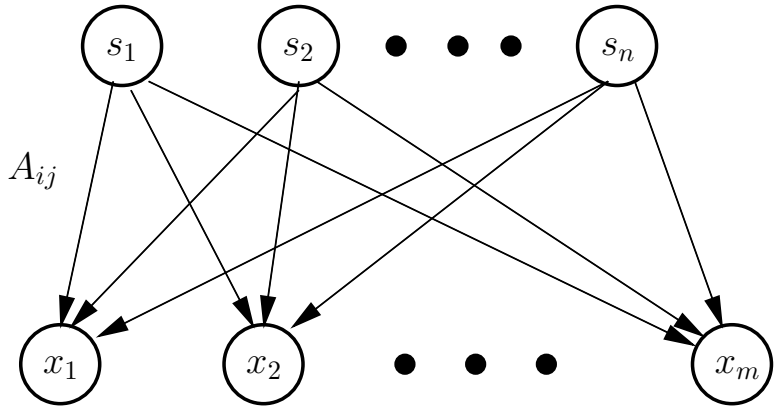

Fig. 1. Linear generative model for PPCA. Latent variables $\{s_i\}$ are assumed to be Gaussian and parameters A_{ij} are learned in such a way that the agreement between data $\{x_i\}$ and model $\left\{\sum_j A_{ij} s_j\right\}$.

As pointed out in [9], 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 $\boldsymbol{A}\boldsymbol{s}$. In such a case, ML estimation of both $\boldsymbol{A}$ and $\boldsymbol{s}$ becomes a separable LS minimization problem [3]. The LS estimates, $\boldsymbol{A}$ and $\boldsymbol{S}$ are computed by

$$\widehat{\boldsymbol{A}}, \widehat{\boldsymbol{S}} = \min_{\boldsymbol{A},\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F^2 . \tag{5}$$

The separable LS minimization is carried out in two steps. Minimizing (5) with respect to $\boldsymbol{A}$ with $\boldsymbol{S}$ being fixed, leads to the M-step updating (4). The estimate $\widehat{\boldsymbol{A}}$ is substituted back into (5), then we obtain a new criterion which is a function of $\boldsymbol{S}$ only

$$\min_{\boldsymbol{S}} \left\|\boldsymbol{X}\boldsymbol{P}_S^{\perp}\right\|_F^2 , \tag{6}$$

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

$$\boldsymbol{P}_S^{\perp} = \boldsymbol{I} - \boldsymbol{S}^T \left(\boldsymbol{S}\boldsymbol{S}^T \right)^{-1} \boldsymbol{S}. \tag{7}$$

The minimization of (6) leads to the E-step updating (3).

3 Integrated-Squared-Error

In this section we introduce ISE and present our main theorem, showing that the minimization of ISE leads to the exact principal directions (eigenvectors of the data covariance matrix) of a set of observed data.

Definition 1 (Integrated-Squared-Error) *Given matrices $\boldsymbol{X} \in \mathbb{R}^{m \times N}$, $\boldsymbol{S} \in \mathbb{R}^{n \times N}$, and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, the integrated-squared-error between $\boldsymbol{X}$ and $\boldsymbol{AS}$ is defined by a linear sum (with positive coefficients, $c_i > 0,\ i = 1, \ldots, n$) of squared errors $\mathcal{J}_i = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{I}_i\boldsymbol{S}\|^2$, i.e.,*

$$\mathcal{J}_{ISE}\left(\boldsymbol{A}, \boldsymbol{S}\right) = \sum_{i=1}^{n} c_i \mathcal{J}_i = \sum_{i=1}^{n} c_i \left\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{I}_i\boldsymbol{S}\right\|^2, \tag{8}$$

where $\boldsymbol{I}_i \in \mathbb{R}^{n \times n}$ is a diagonal matrix, so called, factor selection matrix *with $\boldsymbol{I}_i(j,j) = 1$ for $j = 1, \ldots, i$ and $\boldsymbol{I}_i(j,j) = 0$ for $j = i+1, \ldots, n$.*

Theorem 1 (Main Theorem) *The ISE $\mathcal{J}_{ISE}$ is minimized if and only if the column vectors of $\boldsymbol{A}$ and the row vectors of $\boldsymbol{S}$ have the form, $\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|} = \boldsymbol{\varphi}_i$ and $\frac{\vec{\boldsymbol{s}}_i}{\|\vec{\boldsymbol{s}}_i\|} = \boldsymbol{\xi}_i$ for $i = 1, \ldots, n$ where $\{\boldsymbol{\varphi}_i\}$ are the normalized eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and $\{\boldsymbol{\xi}_i\}$ are the normalized eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$ with associated eigenvalues of $\boldsymbol{X}\boldsymbol{X}^T$ (or $\boldsymbol{X}^T\boldsymbol{X}$), $\lambda_1 \geq \cdots \geq \lambda_n$.*

Proof. See Appendix.

Remarks

- It is known that the minimization of $\mathcal{J}_n = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|^2$ leads to the principal subspace spanned by the largest n eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$, i.e., $\boldsymbol{A}$ is the eigenvector matrix of $\boldsymbol{X}\boldsymbol{X}^T$ post-multiplied by an arbitrary orthogonal matrix.
- $\mathcal{J}_i$ represents the reconstruction error for i-dimensional principal subspace which completely includes $(i-1)$-dimensional principal subspace. Thus the minimization of the integrated-squared-error, implies that each $\mathcal{J}_i$ for $i = 1, \ldots, n$ is minimized. The $\boldsymbol{A}$ and $\boldsymbol{S}$ that minimize the integrated-squared-error, also minimize $\mathcal{J}_n = ||\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}|\,|^2$. However, the $\boldsymbol{A}$ and $\boldsymbol{S}$ which minimize $\mathcal{J}_n$, does not necessarily minimize the integrated-squared-error.

- The reconstruction error which is just squared error, $\|\boldsymbol{X}-\boldsymbol{A}\boldsymbol{S}\|^2$ is invariant to an orthogonal transform $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ because $\boldsymbol{A}\boldsymbol{R}^{-1}$ and $\boldsymbol{R}\boldsymbol{S}$ contributes the same reconstruction error as $\boldsymbol{A}$ and $\boldsymbol{S}$. In contrast, the integrated-squared-error is not invariant under an orthogonal transformation because $\boldsymbol{R}^{-1}\boldsymbol{I}_i\boldsymbol{R} \neq \boldsymbol{I}_i, \forall i \neq n$.
- The main point of Theorem 1 states that the minimization of $\mathcal{J}_{ISE}$ leads to n largest eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$, while the minimization of the conventional reconstruction error $\mathcal{J}_n$ leads to the rotated eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$.

4 EM-ePCA Algorithm

This section presents the EM-ePCA algorithm, its detailed derivation, and a link with the generalized Hebbian algorithm [10, 5]. The algorithm derivation is in the context of separable LS, with the same sprit as EM-PCA [9].

4.1 Algorithm Outline

The integrated-squared-error (8) is iteratively minimized by a simple EM algorithm, so called, EM-ePCA (exact principal directions are emphasized by a letter "e") which is summarized below:

Algorithm Outline: EM-ePCA

E-step

$$\boldsymbol{S} = \left[\mathsf{L}\left(\boldsymbol{A}^T\boldsymbol{A}\right)\right]^{-1}\boldsymbol{A}^T\boldsymbol{X}. \tag{9}$$

M-step

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}\boldsymbol{S}^T\left[\mathsf{U}\left(\boldsymbol{S}\boldsymbol{S}^T\right)\right]^{-1}. \tag{10}$$

The operator L is defined by

$$\mathsf{L}\left(Y_{ij}\right) = \begin{cases} Y_{ij} & \text{for } i \geq j \\ Y_{ij}\frac{\sum_{k=j}^{n} c_k}{\sum_{k=i}^{n} c_k} & \text{for } i < j \end{cases}, \tag{11}$$

for an arbitrary square matrix $\boldsymbol{Y} = [Y_{ij}]$ and $\mathsf{U}(\boldsymbol{Y}) = \left[\mathsf{L}\left(\boldsymbol{Y}^T\right)\right]^T$.

Remarks: We consider two limiting cases:

- In the limit of $\frac{c_{i+1}}{c_i} \to 0,\ i = 1, \ldots, n-1$, the operators L and U become usual lower/upper triangularization operators L_T and U_T where

$$\mathsf{L}_T\left(Y_{ij}\right) = \begin{cases} Y_{ij} \text{ for } i \geq j \\ 0 \quad \text{for } i < j \end{cases}. \tag{12}$$

  The EM-updates (9) and (10) are further simplified as (**EM-ePCA (limiting case)**)

$$\boldsymbol{S} = \left[\mathsf{L}_T\left(\boldsymbol{A}^T\boldsymbol{A}\right)\right]^{-1}\boldsymbol{A}^T\boldsymbol{X}, \tag{13}$$

$$\widehat{\boldsymbol{A}} = \boldsymbol{X}\boldsymbol{S}^T\left[\mathsf{U}_T\left(\boldsymbol{S}\boldsymbol{S}^T\right)\right]^{-1}. \tag{14}$$

  Note that EM-ePCA (limiting case) algorithm is involved with the triangular matrix inversion, hence, computational complexity is greatly reduced, especially for the case of high-dimensional data.
- The EM-PCA algorithm [9] is a special limiting case of our model as $c_i \to 0,\ i = 1, \ldots, n-1$. Under this limit, the inference in (9) reduces to simple least squares projection. The M-step update (10) becomes Wiener filtering.

4.2 Algorithm Derivation

The ISE in (8) has two arguments $\boldsymbol{A}$ and $\boldsymbol{S}$ and its minimization can be carried out using the separable LS. Maximum likelihood estimation in the context of a coupled probabilistic model will be illustrated in Sec. 5, where the EM optimization is used.

The updating rule (10) can be easily derived by solving $\frac{\partial \mathcal{J}_{ISE}}{\partial \boldsymbol{A}} = 0$ with $\boldsymbol{S}$ fixed. A simple calculus leads us to

$$\frac{\partial \mathcal{J}_{ISE}}{\partial \boldsymbol{A}} = 2\sum_{i=1}^{n} c_i \left[\boldsymbol{A}\boldsymbol{I}_i\boldsymbol{S}\boldsymbol{S}^T\boldsymbol{I}_i - \boldsymbol{X}\boldsymbol{S}^T\boldsymbol{I}_i\right] = 0. \tag{15}$$

We define

$$\boldsymbol{\Gamma} = \begin{bmatrix} \sum_{i=1}^{n} c_i & \sum_{i=2}^{n} c_i & \sum_{i=3}^{n} c_i & \cdots & c_n \\ \sum_{i=2}^{n} c_i & \sum_{i=2}^{n} c_i & \sum_{i=3}^{n} c_i & \cdots & c_n \\ \sum_{i=3}^{n} c_i & \sum_{i=3}^{n} c_i & \sum_{i=3}^{n} c_i & \cdots & c_n \\ \vdots & \vdots & \vdots & & \vdots \\ c_n & c_n & c_n & \cdots & c_n \end{bmatrix}, \tag{16}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix} \sum_{i=1}^{n} c_i & 0 & 0 & \cdots & 0 \\ 0 & \sum_{i=2}^{n} c_i & 0 & \cdots & 0 \\ & & \ddots & & \\ 0 & 0 & \cdots & 0 & c_n \end{bmatrix}. \tag{17}$$

With these definitions, one can easily show that

$$\boldsymbol{\Gamma}\boldsymbol{\Lambda}^{-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \frac{\sum_{i=2}^{n} c_i}{\sum_{i=1}^{n} c_i} & 1 & 1 & \cdots & 1 \\ \frac{\sum_{i=3}^{n} c_i}{\sum_{i=1}^{n} c_i} & \frac{\sum_{i=3}^{n} c_i}{\sum_{i=2}^{n} c_i} & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{c_n}{\sum_{i=1}^{n} c_i} & \frac{c_n}{\sum_{i=2}^{n} c_i} & \frac{c_n}{\sum_{i=3}^{n} c_i} & \cdots & 1 \end{bmatrix}. \tag{18}$$

It follows from (15) that we have

$$\begin{aligned} \boldsymbol{A} &= \left[\boldsymbol{X}\boldsymbol{S}^T\right] \left[\sum_{i=1}^{n} c_i \boldsymbol{I}_i\right] \left[\sum_{i=1}^{n} c_i \boldsymbol{I}_i \boldsymbol{S}\boldsymbol{S}^T \boldsymbol{I}_i\right]^{-1} \\ &= \left[\boldsymbol{X}\boldsymbol{S}^T\right] \boldsymbol{\Lambda} \left[\left(\boldsymbol{S}\boldsymbol{S}^T\right) \odot \boldsymbol{\Gamma}\right]^{-1} \\ &= \left[\boldsymbol{X}\boldsymbol{S}^T\right] \left[\left(\boldsymbol{S}\boldsymbol{S}^T\right) \odot \left(\boldsymbol{\Gamma}\boldsymbol{\Lambda}^{-1}\right)\right]^{-1} \\ &= \boldsymbol{X}\boldsymbol{S}^T \left[\mathsf{U}\left(\boldsymbol{S}\boldsymbol{S}^T\right)\right]^{-1}, \end{aligned} \tag{19}$$

where $\odot$ is the Hadamard product (element-wise product) and $\mathsf{U}(\boldsymbol{Y}) = \left[\mathsf{L}\left(\boldsymbol{Y}^T\right)\right]^T$ where $\mathsf{L}(\boldsymbol{Y})$ is defined in (11).

In a similar manner, we can derive the updating rule (9). Solving $\frac{\partial \mathcal{J}_{ISE}}{\partial \boldsymbol{S}} = 0$ with $\boldsymbol{A}$ fixed, leads to

$$
\begin{aligned}
\boldsymbol{S} &= \left[\sum_{i=1}^{n} c_i \boldsymbol{I}_i \boldsymbol{A}^T \boldsymbol{A} \boldsymbol{I}_i\right]^{-1} \left[\sum_{i=1}^{n} c_i \boldsymbol{I}_i\right] \left[\boldsymbol{A}^T \boldsymbol{X}\right] \\
&= \left[\left(\boldsymbol{A}^T \boldsymbol{A}\right) \odot \boldsymbol{\Gamma}\right]^{-1} \boldsymbol{\Lambda} \left[\boldsymbol{A}^T \boldsymbol{X}\right] \\
&= \left[\left(\boldsymbol{\Lambda}^{-1} \boldsymbol{\Gamma}\right) \odot \left(\boldsymbol{A}^T \boldsymbol{A}\right)\right]^{-1} \left[\boldsymbol{A}^T \boldsymbol{X}\right] \\
&= \left[\mathsf{L}\left(\boldsymbol{A}^T \boldsymbol{A}\right)\right]^{-1} \boldsymbol{A}^T \boldsymbol{X}. \qquad (20)
\end{aligned}
$$

4.3 Link with Generalized Hebbian Algorithm

The generalized Hebbian algorithm [10] is one of well-known PCA neural nets which can extract principal components in an unsupervised manner. Although the convergence behavior of GHA was well studied, an optimality criterion is not clear yet. Here we show that the minimal integrated-squared-error in a single layer linear feedforward net leads to the GHA by equalizing the weights in the recognition model to the weights in the generative model. Under this, hidden variables $\boldsymbol{s}$ are estimated by $\boldsymbol{s} = \boldsymbol{A}^T \boldsymbol{x}$. The gradient descent method (for integrated-squared-error minimization) gives the updating rule for $\boldsymbol{A}^T$ which has the form

$$
\boldsymbol{A}^T \leftarrow \boldsymbol{A}^T + \eta \left(\sum_{i=1}^{n} 2 c_i \boldsymbol{I}_i\right) \left\{\boldsymbol{S}\boldsymbol{X}^T - \mathsf{L}\left(\boldsymbol{S}\boldsymbol{S}^T\right) \boldsymbol{A}^T\right\}. \qquad (21)
$$

In order for each row vector of $\boldsymbol{A}^T$ to be updated with identical learning rate, we take a learning rate η as

$$
\eta = \eta_0 \left(\sum_{i=1}^{n} 2 c_i \boldsymbol{I}_i\right)^{-1}, \qquad (22)
$$

to obtain an updating rule for $\boldsymbol{A}^T$:

$$
\boldsymbol{A}^T \leftarrow \boldsymbol{A}^T + \eta_0 \left\{\boldsymbol{S}\boldsymbol{X}^T - \mathsf{L}\left(\boldsymbol{S}\boldsymbol{S}^T\right) \boldsymbol{A}^T\right\}. \qquad (23)
$$

Now we consider two limiting cases of (23)

Case 1: $c_i \to 0$ for $i = 1, \ldots, n-1$

Only single squared error $\| \boldsymbol{X} - \boldsymbol{A}\boldsymbol{S} \|^2$ is considered. In this case, the algorithm (23) reduces to Oja's subspace rule [8]:

$$
\boldsymbol{A}^T \leftarrow \boldsymbol{A}^T + \eta_0 \boldsymbol{S} \left(\boldsymbol{X}^T - \boldsymbol{S}^T \boldsymbol{A}\right). \qquad (24)
$$

Case 2: $c_{i+1}/c_i \to 0$ for $i = 1, \ldots, n-1$

The operator L becomes L_T. Hence the algorithm (23) reduces to the GHA [10]:

$$\boldsymbol{A}^T \leftarrow \boldsymbol{A}^T + \eta_0 \left\{ \boldsymbol{S}\boldsymbol{X}^T - \mathsf{L}_T \left(\boldsymbol{S}\boldsymbol{S}^T \right) \boldsymbol{A}^T \right\}. \tag{25}$$

It seems that this case simply treats a single squared error $\mathcal{J}_1$ due to $c_{i+1}/c_i \to 0$. However, note that we use a normalization factor in the learning rate matrix (22).

The converged weights $\boldsymbol{A}^T$ minimizes the integrated-squared-error under the constraints $\boldsymbol{S} = \boldsymbol{A}^T\boldsymbol{X}$:

$$\mathcal{J} = \sum_{i=1}^{n} c_i \parallel \boldsymbol{X} - \boldsymbol{A}\boldsymbol{I}_i\boldsymbol{A}^T\boldsymbol{X} \parallel^2 \tag{26}$$

Reversely, the weights $\boldsymbol{A}^T$ that minimizes the integrated-squared-error satisfy $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}$ and $\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A} = \mathsf{U}(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A})$. The error function really gives the normalized principal axes of $\boldsymbol{X}\boldsymbol{X}^T$.

The derivation of the GHA has been already treated in [7]. They proposed a criterion to be maximized in the generalization of variance maximization. It uses the recognition model and the weights are constrained to be orthogonal via the Lagrange multipliers. In our method, however, the orthogonality emerges from the minimal integrated-squared-error without orthogonality constraint and we uses the alternating model of recognition and generation with the same weights.

5 Link with Probabilistic Coupled Generative Model

A main motivation of the integrated-squared-error (8) came from the coupled linear generative model [2] where a set of linear Gaussian model shares the same latent variables $\boldsymbol{s} \in \mathbb{R}^n$ and parameters $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with different factor selection matrices $\{\boldsymbol{I}_i\}$. The n-coupled generative model is described by

$$\begin{cases} \boldsymbol{x}_1 = \boldsymbol{A}\boldsymbol{I}_1\boldsymbol{s} + \boldsymbol{v}_1, \\ \boldsymbol{x}_2 = \boldsymbol{A}\boldsymbol{I}_2\boldsymbol{s} + \boldsymbol{v}_2, \\ \quad \vdots \quad \vdots \\ \boldsymbol{x}_n = \boldsymbol{A}\boldsymbol{I}_n\boldsymbol{s} + \boldsymbol{v}_n, \end{cases} \tag{27}$$

where $\boldsymbol{I}_i \in \mathbb{R}^{n\times n}$ is a diagonal matrix with $\boldsymbol{I}_i(j,j) = 1$ for $j = 1, \ldots, i$ and $\boldsymbol{I}_i(j,j) = 0$ for $j = i+1, \ldots, n$.

The coupled linear generative model (see Fig. 2) shares the same latent variables $\boldsymbol{s}$ and factor loading matrix $\boldsymbol{A}$, but takes different isotropic Gaussian noise models $\{\boldsymbol{v}_i \sim \mathcal{N}(0, \sigma_i^2\boldsymbol{I})\}$ and factor selection matrices $\{\boldsymbol{I}_i\}$. The factor

selection matrix $\boldsymbol{I}_i$ is designed in such a way that first i principal directions are selected when each model observes the same data, i.e., $\boldsymbol{x}_1 = \cdots = \boldsymbol{x}_n = \boldsymbol{x}$.

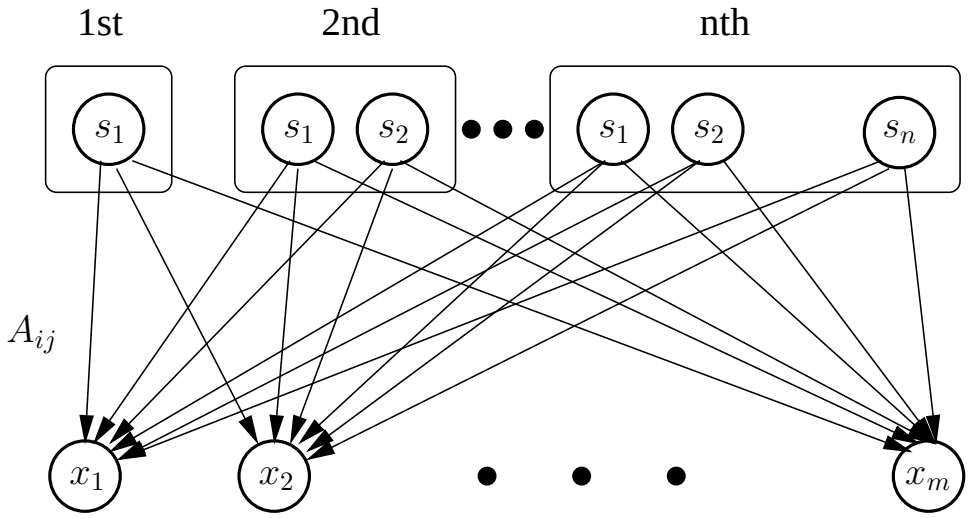

Fig. 2. A coupled linear generative model where n number of models share some common latent variables $\{s_i\}$ in such a way that the 1st model has only s_1, the 2nd model has $s_1,\ s_2$, and the nth model has factors $s_1, \ldots, s_n$. Connections leaving from identical factor nodes in 1st through nth models, are forced to have the same weights. Agreements (in Euclidean sense) between each model and the same observed data, leads to squared-errors $\mathcal{J}_i$. Parameters A_{ij} are learned in such a way that the sum of squared-errors are minimized.

For mutually independent isotropic Gaussian noise models, the joint probability distribution $p(\boldsymbol{x}_1 = \boldsymbol{x}, \cdots, \boldsymbol{x}_n = \boldsymbol{x}|\boldsymbol{s})$ over q-coupled $\boldsymbol{x}$ spaces, conditioned on latent variables $\boldsymbol{s}$ is factorized as

$$\prod_{i=1}^{n} p(\boldsymbol{x}_i = \boldsymbol{x}|\boldsymbol{s}; i) = \prod_{i=1}^{n} (2\pi\sigma_i^2)^{-m/2} \exp\left\{-\frac{1}{2\sigma_i^2} \parallel \boldsymbol{x} - \boldsymbol{A}\boldsymbol{I}_i\boldsymbol{s} \parallel^2\right\}, \tag{28}$$

where $p(\boldsymbol{x}_i = \boldsymbol{x}|\boldsymbol{s}; i)$ is the conditional density for the ith generative model and $\int p(\boldsymbol{x}, \cdots, \boldsymbol{x}|\boldsymbol{s})d\boldsymbol{x} \neq 1$. With unit isotropic Gaussian latent variables $\boldsymbol{s}$, we obtain the marginal distribution $p(\boldsymbol{x}_1 = \boldsymbol{x}, \boldsymbol{x}_2 = \boldsymbol{x}, \cdots, \boldsymbol{x}_n = \boldsymbol{x})$ over the n-coupled $\boldsymbol{x}$ spaces:

$$\int p(\boldsymbol{x}, \cdots, \boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})d\boldsymbol{s} = \prod_{i=1}^{n} (2\pi\sigma_i^2)^{-m/2} |\boldsymbol{M}|^{-1/2} \exp\left\{-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{C}^{-1}\boldsymbol{x}\right\}, \tag{29}$$

where the model covariance $\boldsymbol{C} \in \mathbb{R}^{m\times m}$, the posterior model covariance matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$, and the matrix $\boldsymbol{Q}$ are defined by

$$\boldsymbol{C} = \left[\sum_{i=1}^{n} \boldsymbol{I}/\sigma_i^2 - \boldsymbol{A}\boldsymbol{Q}\boldsymbol{M}^{-1}\boldsymbol{Q}^T\boldsymbol{A}^T\right]^{-1}, \tag{30}$$

$$\boldsymbol{M} = \sum_{i=1}^{n} \boldsymbol{I}_i^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{I}_i/\sigma_i^2 + \boldsymbol{I}, \tag{31}$$

$$\boldsymbol{Q} = \sum_{i=1}^{n} \boldsymbol{I}_i/\sigma_i^2. \tag{32}$$

It follows from Bayes' rule that the posterior distribution $p(\boldsymbol{s}|\boldsymbol{x}, \cdots, \boldsymbol{x})$ over the latent variables $\boldsymbol{s}$ is computed as

$$\begin{aligned} &p(\boldsymbol{s}|\boldsymbol{x}, \cdots, \boldsymbol{x}) \\ &= (2\pi)^{-n/2}|\boldsymbol{M}|^{1/2} \exp\left\{-\frac{1}{2}\left[\boldsymbol{s} - \boldsymbol{M}^{-1}\boldsymbol{Q}^T\boldsymbol{A}^T\boldsymbol{x}\right]^T \boldsymbol{M}\left[\boldsymbol{s} - \boldsymbol{M}^{-1}\boldsymbol{Q}^T\boldsymbol{A}^T\boldsymbol{x}\right]\right\}. \end{aligned} \tag{33}$$

The log-likelihood of observing the complete data under this model is

$$\mathcal{L}_C = \sum_{t=1}^{N} \log\left\{p\left(\boldsymbol{x}_{(t)}, \cdots, \boldsymbol{x}_{(t)}, \boldsymbol{s}_{(t)}\right)\right\}. \tag{34}$$

The expected complete-data log-likelihood $\langle\mathcal{L}_C\rangle$ is given by

$$\begin{aligned} \langle\mathcal{L}_C\rangle = -\sum_{t=1}^{N}\sum_{i=1}^{n}\Bigg\{&\frac{m}{2}\log\sigma_i^2 + \frac{1}{2n}\mathrm{tr}\left(\left\langle\boldsymbol{s}_{(t)}\boldsymbol{s}_{(t)}^T\right\rangle\right) + \frac{1}{2\sigma_i^2}\parallel\boldsymbol{x}_{(t)}\parallel^2 \\ &-\frac{1}{\sigma_i^2}\left\langle\boldsymbol{s}_{(t)}\right\rangle^T\boldsymbol{I}_i^T\boldsymbol{A}^T\boldsymbol{x}_{(t)} + \frac{1}{2\sigma_i^2}\mathrm{tr}\left(\boldsymbol{I}_i^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{I}_i\left\langle\boldsymbol{s}_{(t)}\boldsymbol{s}_{(t)}^T\right\rangle\right)\Bigg\}, \end{aligned} \tag{35}$$

where $\mathrm{tr}(\cdot)$ denotes the trace operator and $\langle\cdot\rangle$ denotes the statistical expectation taken with respect to

$$p\left(\boldsymbol{s}_{(t)}|\boldsymbol{x}_{(t)}, \cdots, \boldsymbol{x}_{(t)}; \boldsymbol{A}, \sigma_i^2\right).$$

The terms irrelevant to parameters were left out in Eq. (35).

In E-step, sufficient statistics are computed:

$$\begin{aligned} \left\langle\boldsymbol{s}_{(t)}\right\rangle &= \boldsymbol{M}^{-1}\boldsymbol{Q}^T\boldsymbol{A}^T\boldsymbol{x}_{(t)} \\ \left\langle\boldsymbol{s}_{(t)}\boldsymbol{s}_{(t)}^T\right\rangle &= \boldsymbol{M}^{-1} + \left\langle\boldsymbol{s}_{(t)}\right\rangle\left\langle\boldsymbol{s}_{(t)}\right\rangle^T. \end{aligned} \tag{36}$$

In M-step, parameters $\{\boldsymbol{A}, \sigma_i^2\}$ are updated by

$$\widehat{\boldsymbol{A}} = \left[\sum_{t=1}^{N} \boldsymbol{x}_{(t)} \left\langle \boldsymbol{s}_{(t)} \right\rangle^T\right] \boldsymbol{Q}^T \left[\sum_{i=1}^{n}\sum_{t=1}^{N} \boldsymbol{I}_i \left\langle \boldsymbol{s}_{(t)} \boldsymbol{s}_{(t)}^T \right\rangle \boldsymbol{I}_i^T / \sigma_i^2\right]^{-1},$$

$$\sigma_i^2 = \frac{1}{Nm}\sum_{t=1}^{N}\left\{ \| \boldsymbol{x}_{(t)} \|^2 - 2\left\langle \boldsymbol{s}_{(t)} \right\rangle^T \boldsymbol{I}_i^T \widehat{\boldsymbol{A}}^T \boldsymbol{x}_{(t)} \right.$$
$$\left. + \mathrm{tr}\left(\left\langle \boldsymbol{s}_{(t)} \boldsymbol{s}_{(t)}^T \right\rangle \boldsymbol{I}_i^T \widehat{\boldsymbol{A}}^T \widehat{\boldsymbol{A}} \boldsymbol{I}_i \right) \right\}. \tag{37}$$

Now we consider a limiting case of the coupled linear generative model (27) as

$$\sigma_n^2 \to 0, \quad \sigma_n^2/\sigma_i^2 = c_i, \quad i = 1, \ldots, n. \tag{38}$$

In this case, maximizing the log-likelihood is practically identical to minimizing the integrated squared error. This can be also confirmed by computing $\lim_{\{\sigma_i^2 \to 0\}} \sigma_n^2 \langle \mathcal{L}_C \rangle$ and omitting constants. The EM-updates (36) and (37) reduce to the EM-ePCA algorithm described in (9) and (10).

6 Numerical Experiments

We investigate the convergence behavior and the performance of our EM algorithms: (1) EM-ePCA given in Eqs. (9) and (10); (2) EM-ePCA (limiting case) given in Eqs. (13) and (14), compared to EM-PCA algorithm given in Eqs. (3) and (4). These three algorithms were tested using three different data sets, including USPS handwritten digit data, face image data, and toy data.

The USPS handwritten digit data contains 16×16 handwritten numeral images for $0, 1, \ldots, 9$, with 400 samples for each numeral. Each image is converted to 256-dimensional vector, to compute 256×256 covariance matrix using 4000 samples. Eigenvalues of the covariance matrix in the case of USPS data, are shown in Fig. 3. The face image data set consists of 1608 266-dimensional vectors and the eigenvalues of the associated covariance matrix are shown in Fig. 4.

Figs. 5 and 6 show the convergence behavior of all these three algorithms, in the case of USPS data and face image data, respectively. In terms of only squared error $\mathcal{J}_n$, it takes almost the same number of iterations for all three algorithms to achieve the final convergence. However, our EM algorithms find exact principal directions (without rotational ambiguity), whereas EM-PCA finds the principal subspace.

Fig. 7 shows the time evolution of the angle between first two principal directions estimated by our EM-ePCA algorithm. The convergence is not always monotonic, but, the orthogonality is always guaranteed. We also applied our

EM-ePCA algorithm to a non-Gaussian data (see Fig. 8) in order to show that our algorithm does not get stuck in a local minimum even for the non-Gaussian data.

7 Conclusions

We have introduced a new error measure, the integrated-squared-error, as an alterative to conventional reconstruction error for PCA. We have shown that exact principal directions of a set of observed data emerged through integrated-squared-error minimization and have presented simple but efficient EM algorithms. In fact our EM-ePCA algorithm and its limiting case become more efficient when the extraction of a few principal components from very high-dimensional data is required. We have also revisited GHA, showing that it could be derived using gradient descent method by minimizing the integrated-squared-error.

Appendix : Proof of Main Theorem

We prove the main theorem in an induction-like manner, using Lemma 1. Note that $\mathcal{J}_i$ represents the reconstruction error for i-dimensional principal subspace which completely includes $(i-1)$-dimensional principal subspace. Thus the minimization of the integrated-squared-error, implies that each $\mathcal{J}_i$ for $i = 1, \ldots, n$ is minimized.

Lemma 1 (SVD) *A matrix $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ has the following decomposition:*

$$\boldsymbol{X} = \sum_{i=1}^{m} \lambda_i^{\frac{1}{2}} \boldsymbol{\varphi}_i \boldsymbol{\xi}_i^T, \tag{39}$$

where $\boldsymbol{\varphi}_i$ and $\boldsymbol{\xi}_i$ are the ith eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$, respectively, and λ_i is the associated nonzero eigenvalue of $\boldsymbol{X}\boldsymbol{X}^T$ or $\boldsymbol{X}^T\boldsymbol{X}$ $(\lambda_1 \geq \cdots \geq \lambda_m)$.

Proof of Main Theorem. It is straightforward to prove the necessity by showing $\frac{\partial \mathcal{J}}{\partial \boldsymbol{A}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{S}} = 0$ at such conditions. The sufficiency is proved below. First we show that $\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{A}} = 0$ and $\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{S}} = 0$ implies that $\frac{\boldsymbol{a}_1}{\|\boldsymbol{a}_1\|} = \boldsymbol{\varphi}_1$ and $\frac{\vec{\boldsymbol{s}}_1}{\|\vec{\boldsymbol{s}}_1\|} = \boldsymbol{\xi}_1$. From $\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{A}} = 0$ and $\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{S}} = 0$, we have

$$\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}_1 = \boldsymbol{A}_1 \left(\boldsymbol{S}_1 \boldsymbol{S}_1^T \boldsymbol{A}_1^T \boldsymbol{A}_1 \right), \tag{40}$$

or

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{S}_1^T = \boldsymbol{S}_1^T\left(\boldsymbol{S}_1\boldsymbol{S}_1^T\boldsymbol{A}_1^T\boldsymbol{A}_1\right), \tag{41}$$

where $\boldsymbol{A}_i = [\boldsymbol{a}_1 \ \cdots \ \boldsymbol{a}_i] \in \mathbb{R}^{m\times i}$ and $\boldsymbol{S}_i = [\vec{\boldsymbol{s}}_1 \ \cdots \ \vec{\boldsymbol{s}}_i]^T \in \mathbb{R}^{i\times N}$.

It follows from (40) and (41) that $\boldsymbol{A}_1(\boldsymbol{A}_1^T\boldsymbol{A}_1)^{-1/2}$ and $\boldsymbol{S}_1^T\ (\boldsymbol{S}_1\boldsymbol{S}_1^T)^{-1/2}$ are the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$, respectively, and $\boldsymbol{S}_1\boldsymbol{S}_1^T\boldsymbol{A}_1^T\boldsymbol{A}_1$ is the corresponding eigenvalue.

Suppose that $\boldsymbol{A}_1\left(\boldsymbol{A}_1^T\boldsymbol{A}_1\right)^{-1/2} = \boldsymbol{\varphi}_k$ and $\left(\boldsymbol{S}_1\boldsymbol{S}_1^T\right)^{-1/2}\boldsymbol{S}_1 = \boldsymbol{\xi}_k^T$. Then, using the Lemma 1, we can write the 1st squared error $\mathcal{J}_1$ as

$$\|\boldsymbol{X} - \boldsymbol{A}_1\boldsymbol{S}_1\|^2 = \left\|\sum_{i=1}^{m}\lambda_i^{1/2}\boldsymbol{\varphi}_i\boldsymbol{\xi}_i^T - \lambda_k^{1/2}\boldsymbol{\varphi}_k\boldsymbol{\xi}_k^T\right\|^2 = \sum_{i\neq k}\lambda_i.$$

Since $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, the squared-error $\mathcal{J}_1$ is minimized when $k = 1$.

Next, as in an induction method, we assume that $\frac{\boldsymbol{a}_l}{\|\boldsymbol{a}_l\|} = \boldsymbol{\varphi}_l$ and $\frac{\vec{\boldsymbol{s}}_l}{\|\vec{\boldsymbol{s}}_l\|} = \boldsymbol{\xi}_l$ for $l = 1, \ldots, i$ from the minimization of $c_1\mathcal{J}_1 + \cdots + c_i\mathcal{J}_i$. Under this assumption, we show that the minimization of $\mathcal{J}_{i+1}$ leads to $\frac{\boldsymbol{a}_{i+1}}{\|\boldsymbol{a}_{i+1}\|} = \boldsymbol{\varphi}_{i+1}$ and $\frac{\vec{\boldsymbol{s}}_{i+1}}{\|\vec{\boldsymbol{s}}_{i+1}\|} = \boldsymbol{\xi}_{i+1}$.

The stationary points of $\mathcal{J}_{i+1}$ satisfy

$$\begin{cases} \boldsymbol{A}_{i+1}^T\boldsymbol{X} = \boldsymbol{A}_{i+1}^T\boldsymbol{A}_{i+1}\boldsymbol{S}_{i+1} \\ \boldsymbol{X}\boldsymbol{S}_{i+1}^T = \boldsymbol{A}_{i+1}\boldsymbol{S}_{i+1}\boldsymbol{S}_{i+1}^T. \end{cases} \tag{42}$$

With these relations, we can rewrite (42) as

$$\begin{bmatrix} \boldsymbol{A}_i^T\boldsymbol{X} \\ \boldsymbol{a}_{i+1}^T\boldsymbol{X} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{S}_i + \boldsymbol{A}_i^T\boldsymbol{a}_{i+1}\vec{\boldsymbol{s}}_{i+1}^T \\ \boldsymbol{a}_{i+1}^T\boldsymbol{A}_i\boldsymbol{S}_i + \boldsymbol{a}_{i+1}^T\boldsymbol{a}_{i+1}\vec{\boldsymbol{s}}_{i+1}^T \end{bmatrix},$$

and

$$[\boldsymbol{X}\boldsymbol{S}_i^T \quad \boldsymbol{X}\vec{\boldsymbol{s}}_{i+1}] = [\boldsymbol{A}_i\boldsymbol{S}_i\boldsymbol{S}_i^T + \boldsymbol{a}_{i+1}\vec{\boldsymbol{s}}_{i+1}^T\boldsymbol{S}_i^T \quad \boldsymbol{A}_i\boldsymbol{S}_i\vec{\boldsymbol{s}}_{i+1} + \boldsymbol{a}_{i+1}\vec{\boldsymbol{s}}_{i+1}^T\vec{\boldsymbol{s}}_{i+1}].$$

Take into account that stationary points of $\mathcal{J}_i$ satisfy

$$\begin{cases} \boldsymbol{A}_i^T\boldsymbol{X} = \boldsymbol{A}_i^T\boldsymbol{A}_i\boldsymbol{S}_i \\ \boldsymbol{X}\boldsymbol{S}_i^T = \boldsymbol{A}_i\boldsymbol{S}_i\boldsymbol{S}_i^T, \end{cases}$$

to obtain orthogonality

$$\begin{cases} \boldsymbol{A}_i^T \boldsymbol{a}_{i+1} = 0 \\ \vec{\boldsymbol{s}}_{i+1}^T \boldsymbol{S}_i^T = 0, \end{cases} \tag{43}$$

Hence we have the relations

$$\begin{cases} \boldsymbol{a}_{i+1}^T \boldsymbol{X} = \boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1} \vec{\boldsymbol{s}}_{i+1}^T \\ \boldsymbol{X} \vec{\boldsymbol{s}}_{i+1} = \boldsymbol{a}_{i+1} \vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1}. \end{cases} \tag{44}$$

The equations (44) can be expressed as

$$\boldsymbol{X}\boldsymbol{X}^T \boldsymbol{a}_{i+1} = \boldsymbol{a}_{i+1} (\vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1} \boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1}), \tag{45}$$

or

$$\boldsymbol{X}^T \boldsymbol{X} \vec{\boldsymbol{s}}_{i+1} = \vec{\boldsymbol{s}}_{i+1} (\vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1} \boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1}). \tag{46}$$

The equations (45) and (46) mean that $\boldsymbol{a}_{i+1}(\boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1})^{-1/2}$ and $\vec{\boldsymbol{s}}_{i+1}(\vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1})^{-1/2}$ are eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$ and $\boldsymbol{X}^T\boldsymbol{X}$, respectively. The $\vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1} a_{i+1}^T a_{i+1}$ is the corresponding eigenvalue. Similarly, if they are the kth eigenvectors and eigenvalue, we can know that $\boldsymbol{\varphi}_k = \boldsymbol{a}_{i+1}(\boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1})^{-1/2}$, $\boldsymbol{\xi}_k^T = (\vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1})^{-1/2}\ \vec{\boldsymbol{s}}_{i+1}^T$ and $\lambda_k = \vec{\boldsymbol{s}}_{i+1}^T \vec{\boldsymbol{s}}_{i+1} \boldsymbol{a}_{i+1}^T \boldsymbol{a}_{i+1}$. Using the Lemma 1, we can also expand the $(i+1)$th squared-error: $\mathcal{J}_{i+1} = \sum_{j \neq k} \lambda_j$. Because $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ and the $(i+1)$th vectors should be orthogonal to the first i vectors in the equation (43), the squared-error $\mathcal{J}_{i+1}$ is minimized at $k = i+1$. ■

Acknowledgment

This work was supported by Korea Ministry of Commerce, Industry, and Energy under Brain Neuroinformatics Program and by KOSEF International Cooperative Research Program.

References

[1] J. -H. Ahn, S. Choi, and J. -H. Oh. A new way of PCA: Integrated-squared-error and EM algorithms. In *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, Montreal, Canada, 2004.

[2] J. -H. Ahn and J. -H. Oh. A constrained EM algorithm for principal component analysis. *Neural Computation*, 15(1):57–65, 2003.

[3] S. Choi. Sequential EM learning for subspace analysis. *Pattern Recognition Letters*, 25(14):1559–1567, 2004.

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

[5] S. Haykin. *Neural Networks:A Comprehensive Foundation*. Prentice-Hall, 2 edition, 1999.

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

[7] J. Karhunen and J. Joutsensalo. Generalization of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562, 1995.

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

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

[10] T. D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989.

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

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

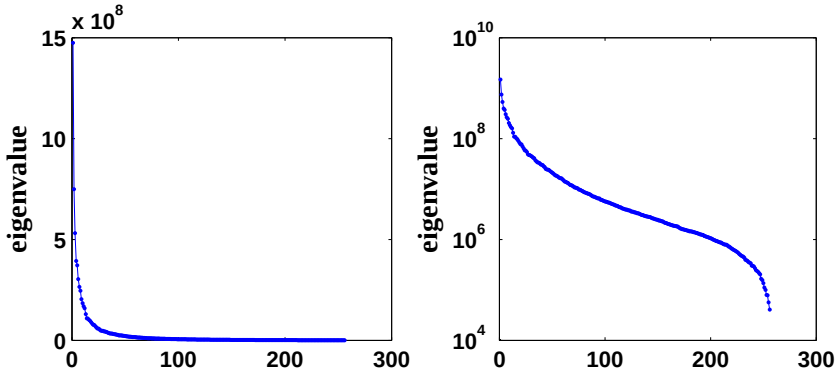

Fig. 3. In the case of USPS handwritten digita data, 256 eigenvalues of the data covariance matrix, are shown, where first 70 eigenvalues are dominant. The right plot is the log-scaled version of the left one.

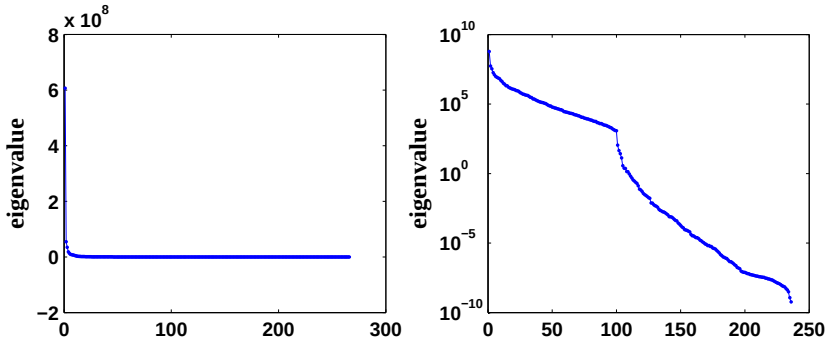

Fig. 4. Eigenvalues of the data covariance matrix in the case of face image data set, are shown. The right plot is the log-scaled version of the left one.

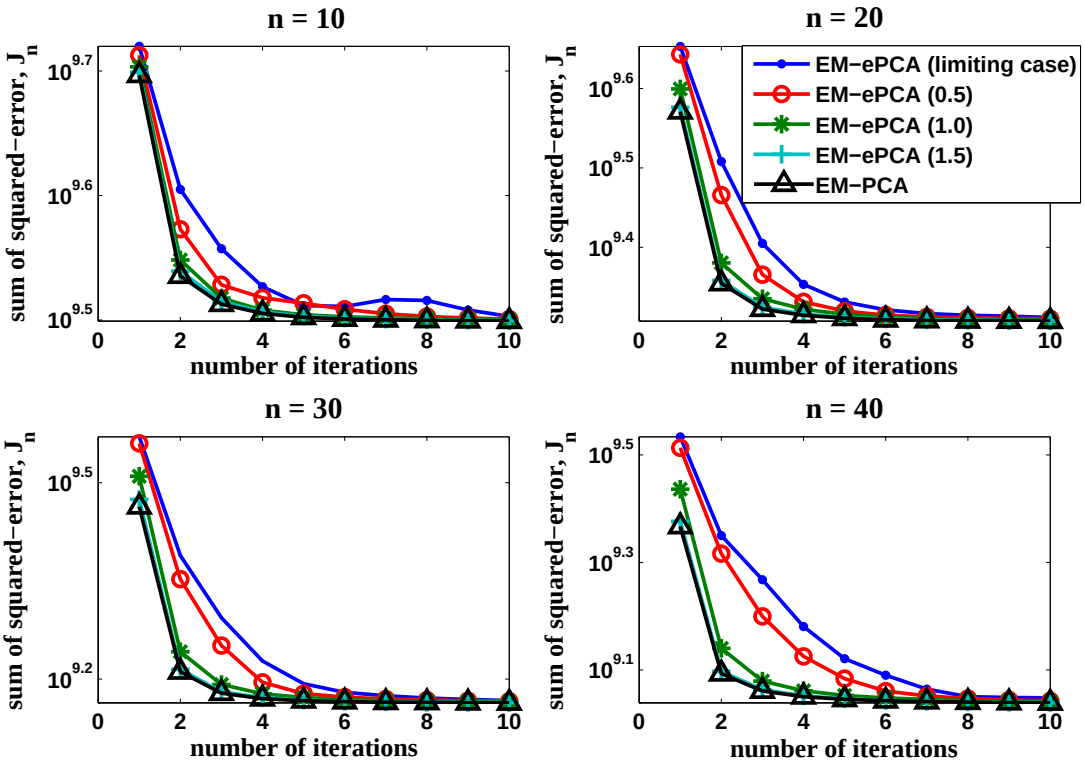

Fig. 5. USPS data: Our proposed EM algorithms, EM-ePCA and its limiting case, show a slightly different convergence behavior in terms of only squared error $\mathcal{J}_n$, compared to the EM-PCA algorithm. It seems that our EM algorithms are slightly slower than the EM-PCA algorithm in a first few iterations, since our EM algorithms tries to minimize the integrated-squared-error rather than just single squared error $\mathcal{J}_n$. However, it takes almost same number of iterations for all the algorithms to achieve the final convergence. In this simulation, $\frac{c_{i+1}}{c_i} = \{0, 0.5, 1.0, 1.5, \infty\}$ were used.

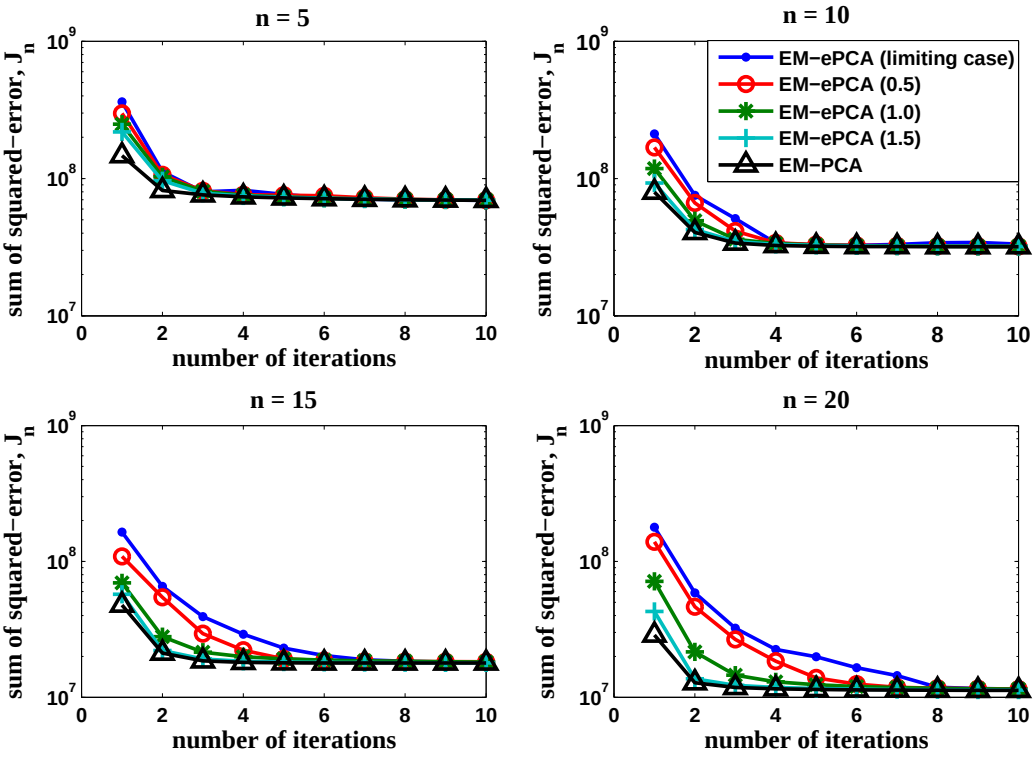

Fig. 6. Face data: Our proposed EM algorithms, EM-ePCA and its limiting case, show a slightly different convergence behavior in terms of only squared error $\mathcal{J}_n$, compared to the EM-PCA algorithm. It seems that our EM algorithms are slightly slower than the EM-PCA algorithm in a first few iterations, since our EM algorithms tries to minimize the integrated-squared-error rather than just single squared error $\mathcal{J}_n$. However, it takes almost same number of iterations for all the algorithms to achieve the final convergence. In this simulation, $\frac{c_{i+1}}{c_i} = \{0, 0.5, 1.0, 1.5, \infty\}$ were used.

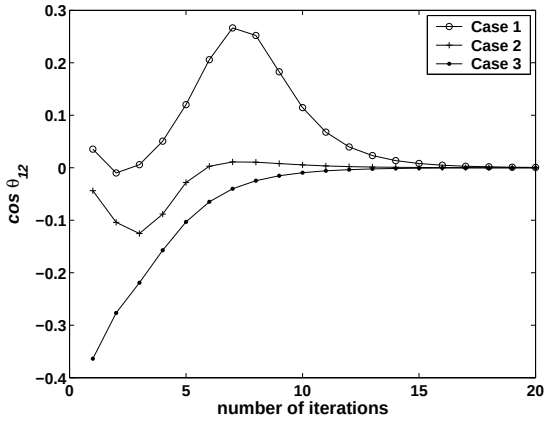

Fig. 7. Two dimensional toy data: Our EM-ePCA algorithm estimates the orthogonal eigenvectors of the data covariance matrix. Cosine of angle between the first and second principal directions are plotted with respect the number of iterations. The convergence is not monotonic, however, after some iterations our EM-ePCA algorithm finds exact two principal directions which are orthogonal each other. Here three different realizations of data were considered.

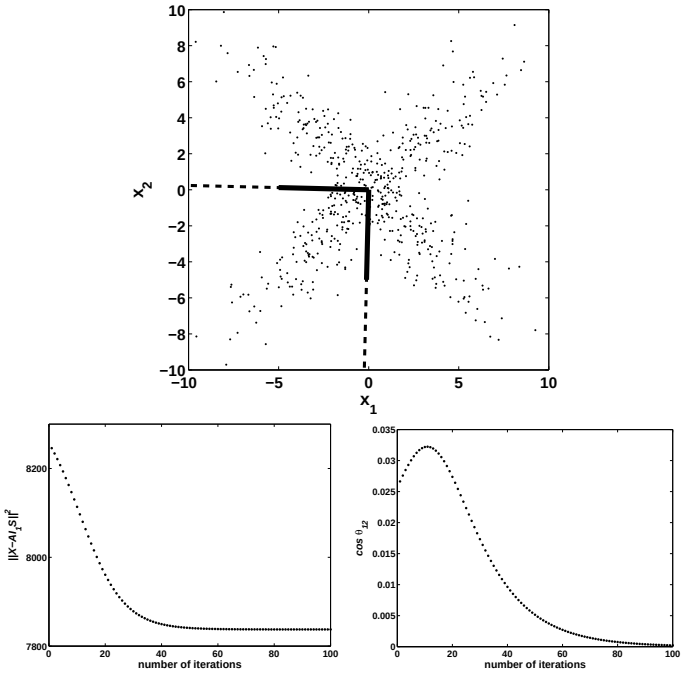

Fig. 8. Two dimensional toy data: A scatter plot of some exemplary non-Gaussian data is shown in the upper panel. The dashed lines indicate the directions of the two leading eigenvectors of the sample covariance matrix whose diagonal components are very close to each other. In the lower pannel, the convergence in terms of $\mathcal{J}_1$ (left pannel) and the angle between first two principal directions (right pannel) is shown. Notice that the difficult learning does not get stuck in a local minimum, although it does take more than 100 iterations to converge, which is unusual for Gaussian data.[9]