Wake-Sleep PCA

Seungjin Choi, *Member, IEEE*

***Abstract*— In this paper we introduce a coupled Helmholtz machine for principal component analysis (PCA), where sub-machines are related through sharing some latent variables and associated weights. We present a wake-sleep algorithm for PCA (referred to as WS-PCA), leading both generative and recognition weights to converge to principal eigenvectors of a data covariance matrix without rotational ambiguity, in contrast to probabilistic PCA and EM-PCA. Then we also present a kernerlized variation, i.e., a wake-sleep algorithm for kernel PCA (WS-KPCA). The coupled Helmholtz machine provides a unified view of principal component analysis, including various existing algorithms as its special cases. The validity of wake-sleep PCA and KPCA algorithms are confirmed by numerical experiments.**

I. Introduction

Spectral decomposition of a symmetric matrix (for example, data covariance matrix or kernel matrix) involves determining eigenvectors of the matrix. This task plays an important role in machine learning, pattern recognition, and signal processing. For instance, principal component analysis (PCA) or kernel PCA requires the calculation of first few principal eigenvectors of a data covariance matrix or a kernel matrix, respectively. Spectral embedding and clustering also rely on eigenvectors of a data-driven matrix (adjacency matrix or affinity matrix).

Singular value decomposition (SVD) is the most widely-used method for eigen-decomposition. A variety of methods have been developed for PCA (see [7], [8] and references therein). A common derivation of PCA is in terms of a linear (orthogonal) projection $\boldsymbol{W} = [\boldsymbol{w}_1 \cdots \boldsymbol{w}_n] \in \mathbb{R}^{m \times n}$ such that given a centered data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N] \in \mathbb{R}^{m \times N}$, the reconstruction error $\|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{X}\|_F^2$ is minimized, where $\|\cdot\|_F$ denotes the Frobenius norm (Euclidean norm). It is well known that the reconstruction error is blind to an arbitrary rotation. Thus, the minimization of the reconstruction error leads to $\boldsymbol{W} = \boldsymbol{U}_1 \boldsymbol{Q}$ where $\boldsymbol{Q} \in \mathbb{R}^{n \times n}$ is an arbitrary orthogonal matrix and the eigen-decomposition of the covariance matrix $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{X}^\top$ is given by

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\Lambda}_1 & 0 \\ 0 & \boldsymbol{\Lambda}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_1 & \boldsymbol{U}_2 \end{bmatrix}^\top, \qquad (1)$$

where $\boldsymbol{U}_1 \in \mathbb{R}^{m \times n}$ contains n largest eigenvectors, the rest of eigenvectors are in $\boldsymbol{U}_2 \in \mathbb{R}^{m \times (m-n)}$, and associated eigenvalues are in $\boldsymbol{\Lambda}_1,\ \boldsymbol{\Lambda}_2$ with $\lambda_1 > \lambda_2 > \cdots > \lambda_m$.

Probabilistic models and associated learning algorithms for PCA were developed, including probabilistic PCA (PPCA) [13] and EM-PCA [11]. As pointed out above, these methods require a post-processing to resolve the rotation ambiguity. Recently a linear coupled generative model [2] was proposed, showing that the model determines principal eigenvectors without any rotational ambiguity. This was further elaborated in [1], where the integrated squared error (ISE) was introduced, showing that an expectation maximization (EM) algorithm that iteratively minimizes ISE determines exact principal eigenvectors of a data covariance matrix. A constrained projection approximation algorithm [5] was also developed, which is a recognition model counterpart of [1].

Helmholtz machine [6] is a statistical inference engine where a recognition model is used to infer a probability distribution over underlying causes from the sensory input and a generative model is used to train the recognition model. The wake-sleep learning is a way of training the Helmholtz machine and the delta-rule wake-sleep learning was used for factor analysis [9].

In this paper, we introduce a coupled Helmholtz machine for PCA, where sub-Helmholtz machines are related through sharing some latent variables as well as associated weights (see Fig. 1). We derive a wake-sleep learning algorithm, referred to as WS-PCA, that iteratively minimizes ISE in the coupled Helmholtz machine, showing that the algorithm indeed determines first few eigenvectors of a data covariance matrix without rotational ambiguity. We also present a kernelized version of WS-PCA, referred to as WS-KPCA, which iteratively determines kernel principal components, in the framework of the coupled Helmholtz machine. A brief idea on the coupled Helmholtz machine was recently reported in [4] and we here present detailed algorithm derivation and a kernelzied algorithm as well. In addition, we show that the coupled Helmholtz machine includes [1], [5] as its special cases.

II. Coupled Helmholtz Machine

Denote observed variables by $\boldsymbol{x} \in \mathbb{R}^m$, leading to a data matrix $\boldsymbol{X} = [\boldsymbol{x}_1 \cdots \boldsymbol{x}_N] \in \mathbb{R}^{m \times N}$ (assume that data is already centered). Latent variables are denoted by $\boldsymbol{Y} = [\boldsymbol{y}_1 \cdots \boldsymbol{y}_N] \in \mathbb{R}^{n \times N}$ ($n \leq m$) that correspond to principal components or factors.

The coupled Helmholtz machine is described by a set of generative models and recognition models (see Fig. 1 for example), where a set of n generative models has the form

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{E}_i \boldsymbol{y}, \quad i = 1, \ldots, n, \qquad (2)$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ contains generative weight vectors $\boldsymbol{a}_j \in \mathbb{R}^m$ in its columns and $\boldsymbol{E}_i \in \mathbb{R}^{n \times n}$ is a diagonal matrix, defined by

$$[\boldsymbol{E}_i]_{jj} = \begin{cases} 1 & \text{for } j = 1, \ldots, i, \\ 0 & \text{for } j = i+1, \ldots, n. \end{cases}$$

Seungjin Choi is with the Department of Computer Science, Pohang University of Science and Technology, Korea (phone: +82-54-279-2259; fax: +82-54-279-2299; email: seungjin@postech.ac.kr).

The set of generative models shares some latent variables y_i ($\boldsymbol{y} = [y_1 \cdots y_n]^\top$) as well as associated generative weights A_{ij}, in such a way that sub-model 2 shares y_1 with sub-model 1 and sub-model 3 shares y_1 and y_2 with sub-model 2 as well as y_1 with sub-model 1, and so on. The recognition model infers latent variables by

$$\boldsymbol{y} = \boldsymbol{W}^\top \boldsymbol{x}, \tag{3}$$

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

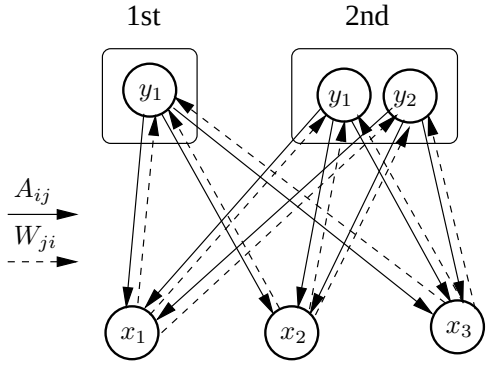


Fig. 1. A coupled Helmholtz machine for $m = 3$ and $n = 2$ is shown. The generative model assumes that observed data x_i is generated by $x_i = A_{i1} y_1$ for the 1st model and $x_i = \sum_{j=1}^{2} A_{ij} y_j$ for the 2nd model, $i = 1, 2, 3$, while these two models share the same latent variable y_1, forcing A_{i1} to be the same weights. The recognition model estimates latent variables y_i by $y_i = \sum_{j=1}^{3} W_{ji} x_j$, $i = 1, 2$ and $j = 1, 2, 3$.

Neglecting the recognition weights ($\boldsymbol{W} = 0$), the coupled Helmholtz machine simply becomes a coupled linear generative model. It was shown in [1] that the minimization of the integrated squared error $\sum_{i=1}^{n} \alpha_i \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{E}_i\boldsymbol{Y}\|_F^2$ ($\alpha_i > 0$ are positive coefficients) in the coupled linear generative model, yields principal eigenvectors of $\boldsymbol{X}\boldsymbol{X}^\top$, i.e., $\boldsymbol{A} = \boldsymbol{U}_1$ without rotational ambiguity. A Gaussian probabilistic coupled model and an EM algorithm, were presented in [1]. In a similar way, we may consider the integrated reconstruction error [5], $\sum_{i=1}^{n} \alpha_i \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{E}_i\boldsymbol{W}^\top\boldsymbol{X}\|_F^2$. This error measure is reminiscent of the deflation method which extracts principal components one by one.

In this paper we consider the following integrated squared error in the coupled Helmholtz machine, that has the form

$$\mathcal{J} = \sum_{i=1}^{n} \alpha_i \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{E}_i\boldsymbol{W}^\top\boldsymbol{X}\|_F^2, \tag{4}$$

Note that this integrated squared error includes the the generative model-based integrated squared error [1] and the integrated reconstruction error [5], as its special cases. In contrast to the generative model-based method, inferring latent variables can be done by the recognition model easily Learned generative models are used to train the recognition model, following the core spirit of the Helmholtz machine.

III. Wake-Sleep PCA

We derive a wake-sleep-like algorithm which iteratively finds the minimum of (4). To this end, we employ the alternating minimization method.

In the sleep phase, we fix $\boldsymbol{A}$ and solve $\frac{\partial \mathcal{J}}{\partial \boldsymbol{W}} = 0$ for $\boldsymbol{W}$, to derive an updating algorithm for the recognition weight matrix $\boldsymbol{W}$. The gradient of (4) with respect to $\boldsymbol{W}$ is given by

$$\frac{\partial \mathcal{J}}{\partial \boldsymbol{W}} = -\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{A}\boldsymbol{\Sigma} + \boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{W}\left[\boldsymbol{A}^\top\boldsymbol{A} \odot \boldsymbol{\Pi}\right], \tag{5}$$

where

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sum_{i=1}^{n}\alpha_i & 0 & 0 & \cdots & 0 \\ 0 & \sum_{i=2}^{n}\alpha_i & 0 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n \end{bmatrix},$$

$$\boldsymbol{\Pi} = \begin{bmatrix} \sum_{i=1}^{n}\alpha_i & \sum_{i=2}^{n}\alpha_i & \sum_{i=3}^{n}\alpha_i & \cdots & \alpha_n \\ \sum_{i=2}^{n}\alpha_i & \sum_{i=2}^{n}\alpha_i & \sum_{i=3}^{n}\alpha_i & \cdots & \alpha_n \\ \sum_{i=3}^{n}\alpha_i & \sum_{i=3}^{n}\alpha_i & \sum_{i=3}^{n}\alpha_i & \cdots & \alpha_n \\ \vdots & & & \ddots & \vdots \\ \alpha_n & \alpha_n & \alpha_n & \cdots & \alpha_n \end{bmatrix},$$

and $\odot$ is the Hadamard product (element-wise product).

With these definitions, it follows from $\frac{\partial \mathcal{J}}{\partial \boldsymbol{W}} = 0$ that

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{A}\boldsymbol{\Sigma}\left[\boldsymbol{A}^\top\boldsymbol{A} \odot \boldsymbol{\Pi}\right]^{-1} \\ &= \boldsymbol{A}\left[\left(\boldsymbol{A}^\top\boldsymbol{A}\right) \odot \left(\boldsymbol{\Pi}\boldsymbol{\Sigma}^{-1}\right)\right]^{-1} \\ &= \boldsymbol{A}\left[\mathsf{U}\left(\boldsymbol{A}^\top\boldsymbol{A}\right)\right]^{-1}, \end{aligned} \tag{6}$$

where $\mathsf{U}(\boldsymbol{Z})$ is an element-wise operator, whose arguments Z_{ij} are transformed by

$$\mathsf{U}(Z_{ij}) = \begin{cases} Z_{ij}\frac{\sum_{l=i}^{n}\alpha_l}{\sum_{l=j}^{n}\alpha_l} & \text{if } i > j, \\ Z_{ij} & \text{if } i \le j. \end{cases} \tag{7}$$

The operator $\mathsf{U}(\boldsymbol{Z})$ results from the structure of $\boldsymbol{\Pi}\boldsymbol{\Sigma}^{-1}$ given by

$$\boldsymbol{\Pi}\boldsymbol{\Sigma}^{-1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ \frac{\sum_{i=2}^{n}\alpha_i}{\sum_{i=1}^{n}\alpha_i} & 1 & 1 & \cdots & 1 \\ \frac{\sum_{i=3}^{n}\alpha_i}{\sum_{i=1}^{n}\alpha_i} & \frac{\sum_{i=3}^{n}\alpha_i}{\sum_{i=2}^{n}\alpha_i} & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\alpha_n}{\sum_{i=1}^{n}\alpha_i} & \frac{\alpha_n}{\sum_{i=2}^{n}\alpha_i} & \frac{\alpha_n}{\sum_{i=3}^{n}\alpha_i} & \cdots & 1 \end{bmatrix}.$$

Next, in the wake phase, we fix $\boldsymbol{W}$ and solve $\frac{\partial \mathcal{J}}{\partial \boldsymbol{A}} = 0$ for $\boldsymbol{A}$, to derive an updating algorithm for the generative weight matrix $\boldsymbol{A}$. It follows from $\frac{\partial \mathcal{J}}{\partial \boldsymbol{A}} = 0$ that

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{X}\boldsymbol{Y}^\top\boldsymbol{\Sigma}\left[\boldsymbol{Y}\boldsymbol{Y} \odot \boldsymbol{\Pi}\right]^{-1} \\ &= \boldsymbol{X}\boldsymbol{Y}^\top\left[\mathsf{U}\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)\right]^{-1}. \end{aligned} \tag{8}$$

The updating algorithms in (6) and (8) are referred to as *wake-sleep PCA* (WS-PCA). As in [1], [2], we also consider the limiting case where $\frac{\alpha_{i+1}}{\alpha_i} \to 0$ for $i = 1, \ldots, n-1$, that is, weighting α_i's are rapidly diminishing as i increases.

In such a case, the operator $\mathsf{U}(\cdot)$ becomes the conventional upper-triangularization operator U_T which is given by

$$\mathsf{U}_T(Z_{ij}) = \begin{cases} 0 & \text{if } i > j, \\ Z_{ij} & \text{if } i \leq j. \end{cases} \tag{9}$$

Replacing the operator U in WS-PCA by the upper-triangularization operator U_T, WS-PCA is referred to as WS-PCA (limiting case). Note that the limiting case where $\frac{\alpha_{i+1}}{\alpha_i} \rightarrow 0$ for $i = 1, \ldots, n-1$, is totally different from the case for $\alpha_l = 0$ ($l = 2, \ldots, n$). In a way similar to [3], one can prove that both $\boldsymbol{A}$ and $\boldsymbol{W}$ that are minimizers of (4), correspond to principal eigenvectors of a data covariance matrix with rotation ambiguity. As will be shown in numerical experiments, both generative weights $\boldsymbol{A}$ and recognition weights $\boldsymbol{W}$ in WS-PCA, converge to exact eigenvectors of a data covariance matrix ($\boldsymbol{a}_1$ converges to the largest eigenvector and $\boldsymbol{a}_2$ converges to the second largest eigenvector, and so on). Regardless of values of coefficients α_i, both weights converge to exact eigenvectors, however, the convergence behavior of WS-PCA is slightly different, depending on the ratio, $\frac{\alpha_{i+1}}{\alpha_i}$ for $i = 1, \ldots, n-1$.

Algorithm Outline: WS-PCA

Sleep phase

$$\boldsymbol{W} = \boldsymbol{A}\left[\mathsf{U}\left(\boldsymbol{A}^\top \boldsymbol{A}\right)\right]^{-1}, \tag{10}$$

$$\boldsymbol{Y} = \boldsymbol{W}^\top \boldsymbol{X}. \tag{11}$$

Wake phase

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{Y}^\top \left[\mathsf{U}\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)\right]^{-1}. \tag{12}$$

IV. Wake-Sleep Kernel PCA

Kernel PCA is a symmetric eigenvalue problem of the covariance matrix in a feature space $\mathcal{F}$ that is constructed by a nonlinear mapping $\varphi : \mathbb{R}^m \rightarrow \mathcal{F}$. It seeks an eigen-decomposition

$$\boldsymbol{C}_\varphi = \boldsymbol{U}_\varphi \boldsymbol{\Lambda}_\varphi \boldsymbol{U}_\varphi^\top, \tag{13}$$

where $\boldsymbol{C}_\varphi = \frac{1}{N}\sum_{t=1}^N \varphi(\boldsymbol{x}_t)\varphi(\boldsymbol{x}_t)^\top$ is the covariance matrix in the feature space $\mathcal{F}$. $\boldsymbol{U}_\varphi$ and $\boldsymbol{\Lambda}_\varphi$ are eigenvector and eigenvalue matrices for $\boldsymbol{C}_\varphi$.

All solutions $\boldsymbol{U}_\varphi$ with nonzero eigenvalues lie in the span of $\varphi(\boldsymbol{x}_1), \ldots, \varphi(\boldsymbol{x}_N)$, leading to

$$\boldsymbol{U}_\varphi = \boldsymbol{\Phi}\boldsymbol{\Gamma}, \tag{14}$$

where $\boldsymbol{\Phi} = [\varphi(\boldsymbol{x}_1) \cdots \varphi(\boldsymbol{x}_N)]$ and $\boldsymbol{\Gamma}$ contains associated coefficients.

Invoking (14), kernel PCA [12] involves the eigen-decomposition of the kernel matrix $\boldsymbol{K} = \boldsymbol{\Phi}^\top\boldsymbol{\Phi}$,

$$\frac{1}{N}\boldsymbol{K} = \boldsymbol{\Gamma}\boldsymbol{\Lambda}\boldsymbol{\Gamma}^\top, \tag{15}$$

instead of the eigen-decomposition of $\boldsymbol{C}_\varphi$ which requires the knowledge of $\varphi(\cdot)$. The projection of a test data point $\varphi(\boldsymbol{x})$ onto the eigenvectors $\boldsymbol{U}_\varphi$, is given by

$$\boldsymbol{U}_\varphi^\top \varphi(\boldsymbol{x}) = \boldsymbol{\Gamma}^\top \boldsymbol{\Phi}^\top \varphi(\boldsymbol{x}). \tag{16}$$

Kernel PCA requires the diagonalization of large matrices (kernel matrices whose dimension is growing according to the number of data points). In order to alleviate this numerical limitation, an EM algorithm was proposed in [10] where the kernelization of EM-PCA [11] was presented. As pointed out in the beginning of the paper, EM-PCA and EM-KPCA require a post-processing to resolve the rotational ambiguity. Here we present the kernerlized version of WS-PCA, referred to as WS-KPCA.

The sleep phase of WS-KPCA emerges by means of setting $\boldsymbol{W} = \boldsymbol{\Phi}\widetilde{\boldsymbol{\Gamma}}$ and $\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Gamma}$ in (10), leading to (17). Then, we have

$$\boldsymbol{Y} = \boldsymbol{W}^\top\boldsymbol{\Phi} = \widetilde{\boldsymbol{\Gamma}}^\top \boldsymbol{K}.$$

The wake-phase can be easily derived, by replacing $\boldsymbol{A}$ and $\boldsymbol{X}$ by $\boldsymbol{\Phi}\boldsymbol{\Gamma}$ and $\boldsymbol{\Phi}$, respectively in (12), leading to (19).

Algorithm Outline: WS-KPCA

Sleep phase

$$\widetilde{\boldsymbol{\Gamma}} = \boldsymbol{\Gamma}\left[\mathsf{U}\left(\boldsymbol{\Gamma}^\top\boldsymbol{K}\boldsymbol{\Gamma}\right)\right]^{-1}, \tag{17}$$

$$\boldsymbol{Y} = \widetilde{\boldsymbol{\Gamma}}^\top \boldsymbol{K}. \tag{18}$$

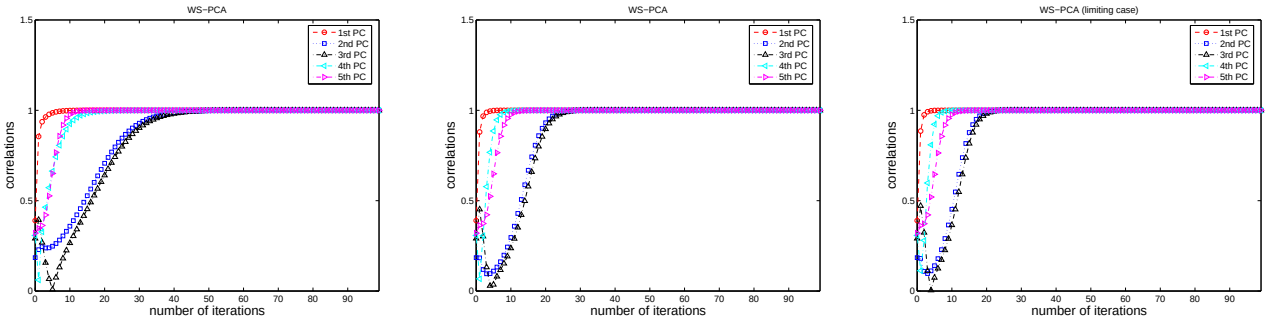
Wake phase

$$\boldsymbol{\Gamma} = \boldsymbol{Y}^\top\left[\mathsf{U}\left(\boldsymbol{Y}\boldsymbol{Y}^\top\right)\right]^{-1}. \tag{19}$$

V. Numerical Experiments

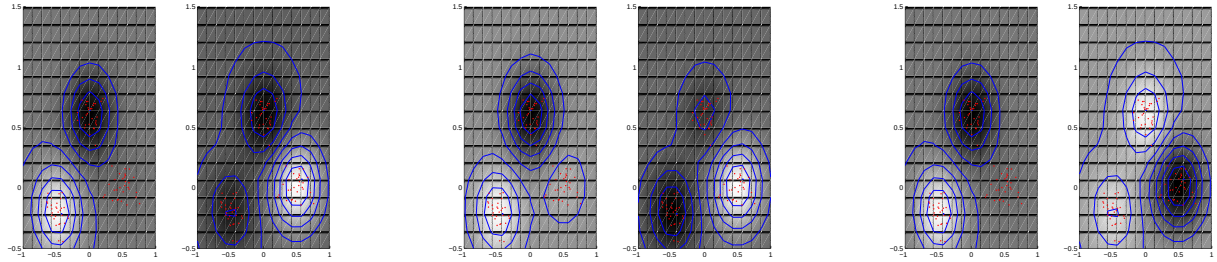
In the first example, 10-dimensional data vectors of length 1000, $\boldsymbol{X} \in \mathbb{R}^{10\times 1000}$, were generated by linearly-transforming 5-dimensional Gaussian random vectors (i.e., $m = 10$ and $n = 5$). Fig 2 shows the convergence behavior of the WS-PCA algorithm (and its limiting case) with different choice of α_i. Regardless of values of α_i, generative weights (or recognition weights) converge to true eigenvectors. However, the convergence behavior of the WS-PCA algorithm is slightly different, especially according to the ratio $\frac{\alpha_{i+1}}{\alpha_i}$ for $i = 1, \ldots, n-1$ (see Fig. 2). The WS-PCA achieves the faster convergence, as the ratio, $\frac{\alpha_{i+1}}{\alpha_i}$ for $i = 1, \ldots, n-1$, becomes smaller. In fact, the limiting case of WS-PCA (where U_T is used instead of U) shows the fastest convergence (see Fig. 2).

For kernel PCA, we use the same toy example used in [10], [12]. Three 2-dimensional Gaussian clusters (see Fig. 3) with means $[-0.5, -0.2; 0.0, 0.6; 0.5, 0.0]$ and common variance 0.1, are generated. Each cluster consists of 30 data points. In this example, we use a RBF kernel, $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{0.1}\right\}$. Fig.3 shows first two nonlinear principal components determined by KPCA (a direct diagonalization based on SVD), EM-KPCA [10], and WS-KPCA. In all those



(a) (b) (c)

Fig. 2. Evolution of generative weight vectors, is shown in terms of the absolute value of the inner produce between a weight vector and a true eigenvector (computed by SVD): (a) WS-PCA with $\frac{\alpha_{i+1}}{\alpha_i} = 0.5$ and $\alpha_1 = 1$; (b) WS-PCA with $\frac{\alpha_{i+1}}{\alpha_i} = 0.1$ and $\alpha_1 = 1$; (c) WS-PCA (limiting case)



(a) (b) (c)

Fig. 3. First two nonlinear principal components are shown, in the case of the toy example where three 2-dimensional Gaussian clusters are considered: (a) SVD-based KPCA; (b) EM-KPCA; (c) WS-KPCA. Contours represent lines of constant principal component value.

methods, first two principal components nicely separate the three clusters. However, SVD-based KPCA and WS-KPCA shows the exactly same contours (associated with constant principal component values), whereas slightly different contours are shown in the case of EM-KPCA.

VI. Conclusions

We have introduced a coupled Helmholtz machine where latent variables as well as associated weights are shared by a set of sub-Helmholtz machines. We have presented wake-sleep algorithms for PCA and KPCA in the framework of the coupled Helmholtz machine, showing that algorithms indeed determine exact principal eigenvectors without rotational ambiguity. Wake-sleep algorithms for PCA and KPCA, are useful in applications where first few principal components need to be extracted, in the case of high-dimensional data or large sample size.

Acknowledgments: This work was supported by Korea MCIE under Brain Neuroinformatics Program, and National Core Research Center for Systems Bio-Dynamics.

References

[1] J. H. Ahn, S. Choi, and J. H. Oh, "A new way of PCA: Integrated-squared-error and EM algorithms," in *Proceedings of IEEE International Conference on 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*, vol. 15, no. 1, pp. 57–65, 2003.

[3] J. H. Ahn, J. H. Oh, and S. Choi, "Learning principal directions: Integrated-squared-error minimization," *Neurocomputing*, vol. 70, pp. 1372–1381, 2007.

[4] S. Choi, "Coupled Helmholtz machine for PCA," *Electronics Letters*, vol. 42, no. 16, pp. 936–937, 2006.

[5] S. Choi, J. H. Ahn, and A. Cichocki, "Constrained projection approximation algorithms for principal component analysis," *Neural Processing Letters*, vol. 24, no. 1, pp. 53–65, 2006.

[6] P. Dayan, G. E. Hinton, R. M. Neal, and R. S. Zemel, "The Helmholtz machine," *Neural Computation*, vol. 7, pp. 1022–1037, 1995.

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

[8] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Spinger-Verlag, 2002.

[9] R. M. Neal and P. Dayan, "Factor analysis using delta-rule wake-sleep learning," *Neural Computation*, vol. 9, pp. 1781–1803, 1997.

[10] R. Rosipal and M. Girolami, "An expectation-maximization approach to nonlinear component analysis," *Neural Computation*, vol. 13, pp. 505–510, 2001.

[11] S. T. Roweis, "EM algorithms for PCA and SPCA," in *Advances in Neural Information Processing Systems*, vol. 10. MIT Press, 1998, pp. 626–632.

[12] B. Schölkopf, A. J. Smola, and K. R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[13] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Computation*, vol. 11, no. 2, pp. 443–482, 1999.