Transformations for variational factor analysis to speed up learning

Jaakko Luttinen*, Alexander Ilin

Department of Information and Computer Science, Helsinki University of Technology TKK, P.O. Box 5400, FI-02015 TKK, Finland

Abstract

We propose simple transformation of the hidden states in variational Bayesian factor analysis models to speed up the learning procedure. The speed-up is achieved by using proper parameterization of the posterior approximation which allows joint optimization of its individual factors, thus the transformation is theoretically justified. We derive the transformation formulae for variational Bayesian factor analysis and show experimentally that it can significantly improve the rate of convergence. The proposed transformation basically performs centering and whitening of the hidden factors taking into account the posterior uncertainties. Similar transformations can be applied to other variational Bayesian factor analysis models as well.

Key words:
principal component analysis, factor analysis, variational Bayesian learning, subspace rotation, speed-up

1. Introduction

Probabilistic latent variable models is a powerful tool of unsupervised data analysis which can efficiently be used for data compression, feature extraction and dynamical modeling. In this article, we consider a latent variable model called *factor analysis* [3, 4], in which observed data vectors $\mathbf{y}_n$ are assumed to be constructed from variables $\mathbf{x}_n$ using a linear mapping:

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n\,, \quad n = 1, \ldots, N\,, \tag{1}$$

where matrix $\mathbf{W}$ and vector $\boldsymbol{\mu}$ are adaptive parameters and $\boldsymbol{\epsilon}_n$ is a noise term. The latent latent variables $\{\mathbf{x}_n\}$ are modelled to be zero-mean Gaussian with uncorrelated and unit-variance components, while the noise term $\boldsymbol{\epsilon}_n$ is also a zero-mean Gaussian with a diagonal covariance matrix.

Factor analysis (FA) can be seen as a basic latent variable model which has been extended in many ways. Probabilistic principal component analysis (PCA) [24] is a FA model with isotropic noise $\boldsymbol{\epsilon}_n$ [6]. Linear state-space models (see, e.g., [11]) use the linear generative model (1) but include dynamics into the prior model for the latent variables. Mixtures of factor analyzers allow different local FA models in different regions of the input space [10]. Non-Gaussian factors yield a noisy independent component analysis model [2]. A nonlinear mapping from the hidden factors $\mathbf{x}_n$ to observations $\mathbf{y}_n$ is assumed in nonlinear factor analysis models [12, 15, 16], exponential family PCA [21] or nonlinear state-space models [25].

Bayesian methods provide a principled way for learning latent variable models. The main advantages of Bayesian techniques include easy handling of missing data, resistance to overfitting and natural ways of model comparison. In Bayesian inference, both the adaptive parameters, including $\mathbf{W}$ and $\boldsymbol{\mu}$, and the latent variables $\{\mathbf{x}_n\}$ are assigned priors which express our modeling assumptions. Then, the goal is to evaluate the joint distribution over all the unknown variables given the observations $\{\mathbf{y}_n\}$.

Variational Bayesian (VB) methods have been widely used in FA and its extensions (see, e.g., [5, 8, 15]). Perhaps the main argument for using VB methods is a typically large number of unknown parameters in latent variable models, which can make sampling methods computationally prohibitive. VB methods approximate the true posterior probability density function (pdf) $p(\{\mathbf{x}_n\}, \boldsymbol{\mu}, \mathbf{W}, \boldsymbol{\alpha}, ...|\{\mathbf{y}_n\})$ of the unknown variables using a simpler pdf which is factorized with respect to groups of variables (see, e.g., [7]).

The variational approximation usually assumes that the hidden factors $\{\mathbf{x}_n\}$ and the rows of the loading matrix $\mathbf{W}$ are inpendent a posteriori, which is done mainly for computational convenience. However, all the variables in the original FA model are strongly coupled. This often causes slow convergence of VB methods in practice.

Parameter-expanded VB (PX-VB) methods were proposed recently to address the slow convergence problem [23]. The general idea is to use auxiliary parameters in the original model to reduce the effect of strong couplings between different variables. The auxiliary parameters are optimized during learning, which corresponds to *joint* optimization of different components of the variational approximation of the true posterior. In this way strong functional couplings between the components are reduced, which facilitates faster convergence. One of the main challenges for applying the PX-VB methodology is to use proper reparameterization of the original model.

In this paper, we present a very similar idea in the context of learning VB factor analysis (VBFA) models. Similarly to PX-VB, we propose to use auxiliary variables which are optimized in conjunction with the variational approximation. In

*Corresponding author
Email addresses: jaakko.luttinen@tkk.fi (Jaakko Luttinen), alexander.ilin@tkk.fi (Alexander Ilin)

our presentation, the auxiliary variables reparameterize the approximating posterior pdf rather than the original model. Thus, the adaptation of the auxiliary variables corresponds to transformations of the latent variables in our terminology rather than to reparameterization of the original model in the PX-VB terminology. Similarly to examples of PX-VB [23], our proposed family of possible transformations may look ad-hoc but we provide a rigorous proof that the optimal transformation (within the proposed family) always maximizes the lower bound of the marginal likelihood. Thus, the propsed transformations facilitate faster convergence.

In the experimental part, we consider the VB PCA model [8] and show that the proposed methodology can lead to significantly faster convergence. The preliminary results of this work were presented in our conference paper [20].

2. Transformations for variational Bayesian factor analysis

2.1. Learning VBFA model by variational EM algorithm

Let us denote by $\{\mathbf{y}_n\}_{n=1}^N$ a set of M-dimensional observations $\mathbf{y}_n$. The data are assumed to be generated from hidden D-dimensional states $\{\mathbf{x}_n\}_{n=1}^N$:

$$p(\mathbf{Y}|\mathbf{W},\mathbf{X},\boldsymbol{\mu},\boldsymbol{\tau}) = \prod_{n=1}^{N} \mathcal{N}\left(\mathbf{y}_n|\mathbf{W}\mathbf{x}_n+\boldsymbol{\mu},\mathrm{diag}(\boldsymbol{\tau})^{-1}\right), \quad (2)$$

where $\mathcal{N}(\mathbf{a}|\mathbf{b},\mathbf{C})$ denotes a Gaussian probability density function over $\mathbf{a}$ with mean $\mathbf{b}$ and covariance matrix $\mathbf{C}$, $\mathbf{W}$ is an $M\times D$ loading matrix, $\boldsymbol{\mu}$ is a bias term and $\mathrm{diag}(\boldsymbol{\tau})$ is a diagonal precision matrix with elements τ_m on the diagonal.

The prior models for the unknown variables are

$$p(\mathbf{X}) = \prod_{n=1}^{N}\prod_{d=1}^{D}\mathcal{N}(x_{dn}|0,1), \qquad p(\boldsymbol{\mu}) = \prod_{m=1}^{M}\mathcal{N}(\mu_m|0,\beta^{-1}),$$
$$p(\mathbf{W}|\boldsymbol{\alpha}) = \prod_{m=1}^{M}\prod_{d=1}^{D}\mathcal{N}(w_{md}|0,\alpha_d^{-1}), \quad p(\boldsymbol{\alpha}) = \prod_{d=1}^{D}\mathcal{G}(\alpha_d|a_{\boldsymbol{\alpha}},b_{\boldsymbol{\alpha}}),$$
$$p(\boldsymbol{\tau}) = \prod_{m=1}^{M}\mathcal{G}(\tau_m|a_{\boldsymbol{\tau}},b_{\boldsymbol{\tau}}),$$

where $\mathcal{G}(\chi|a,b)$ is a Gamma density function which has the expectations $\langle\chi\rangle = a/b$ and $\langle\log\chi\rangle = \psi(a) - \log(b)$, with $\psi(a)$ being the digamma function, and the hyperparameters β, $a_{\boldsymbol{\alpha}}$, $b_{\boldsymbol{\alpha}}$, $a_{\boldsymbol{\tau}}$, and $b_{\boldsymbol{\tau}}$ are fixed to small values (e.g., 10^{-5}) resulting in broad priors.

In VBFA, the joint posterior probability density function (pdf) of the unknown variables $\boldsymbol{\Theta} = \{\mathbf{X},\boldsymbol{\mu},\mathbf{W},\boldsymbol{\alpha},\boldsymbol{\tau}\}$ is approximated with a suitable pdf $q(\boldsymbol{\Theta})$. The approximate pdf is often chosen to factorize with respect to the variables as $q(\boldsymbol{\Theta}) = q(\mathbf{X})q(\boldsymbol{\mu})q(\mathbf{W})q(\boldsymbol{\alpha})q(\boldsymbol{\tau})$. The approximate distribution $q(\boldsymbol{\Theta})$ is found by maximizing the lower bound of the marginal log-likelihood

$$\log p(\mathbf{Y}) \geq \mathcal{L}(q) = \int q(\boldsymbol{\Theta})\log\frac{p(\mathbf{Y},\boldsymbol{\Theta})}{q(\boldsymbol{\Theta})}d\boldsymbol{\Theta}$$
$$= \langle\log p(\mathbf{Y}|\boldsymbol{\Theta})\rangle - \left\langle\log\frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta})}\right\rangle, \quad (3)$$

where $\langle\cdot\rangle$ denotes the expectation over the q distribution. This optimization results in the following forms for the factors [8]:

$$q(\mathbf{X}) = \prod_{n=1}^{N}\mathcal{N}(\mathbf{x}_n|\overline{\mathbf{x}}_n,\boldsymbol{\Sigma}_{\mathbf{x}_n}), \qquad q(\boldsymbol{\mu}) = \prod_{m=1}^{M}\mathcal{N}(\mu_m|\overline{\mu}_m,\widetilde{\mu}_m), \quad (4)$$
$$q(\mathbf{W}) = \prod_{m=1}^{M}\mathcal{N}(\mathbf{w}_m|\overline{\mathbf{w}}_m,\boldsymbol{\Sigma}_{\mathbf{w}_m}), \quad q(\boldsymbol{\alpha}) = \prod_{d=1}^{D}\mathcal{G}(\alpha_d|\breve{a}_{\alpha_d},\breve{b}_{\alpha_d}), \quad (5)$$
$$q(\boldsymbol{\tau}) = \prod_{m=1}^{M}\mathcal{G}(\tau_m|\breve{a}_{\tau_m},\breve{b}_{\tau_m}), \quad (6)$$

where $\mathbf{w}_m^{\mathrm{T}}$ is the m-th row of $\mathbf{W}$. In the variational EM algorithm, the optimization is done by alternate updates of the individual factors in (4)-(6) while keeping the rest of the factors fixed. The update rules for the parameters are as follows:

$$\boldsymbol{\Sigma}_{\mathbf{x}_n}^{-1} = \mathbf{I} + \sum_{m\in O_{mn}}\langle\tau_m\rangle\left\langle\mathbf{w}_m\mathbf{w}_m^{\mathrm{T}}\right\rangle,$$
$$\overline{\mathbf{x}}_n = \boldsymbol{\Sigma}_{\mathbf{x}_n}\sum_{m\in O_{mn}}\langle\tau_m\rangle\langle\mathbf{w}_m\rangle(y_{mn}-\langle\mu_m\rangle),$$
$$\boldsymbol{\Sigma}_{\mathbf{w}_m}^{-1} = \mathrm{diag}\,\langle\boldsymbol{\alpha}\rangle + \langle\tau_m\rangle\sum_{n\in O_{mn}}\left\langle\mathbf{x}_n\mathbf{x}_n^{\mathrm{T}}\right\rangle,$$
$$\overline{\mathbf{w}}_m = \boldsymbol{\Sigma}_{\mathbf{w}_m}\langle\tau_m\rangle\sum_{n\in O_{mn}}\langle\mathbf{x}_n\rangle(y_{mn}-\langle\mu_m\rangle),$$
$$\widetilde{\mu}_m^{-1} = \beta + N_m\langle\tau_m\rangle,$$
$$\overline{\mu}_m = \widetilde{\mu}_m\langle\tau_m\rangle\sum_{n\in O_{mn}}\left(y_{mn}-\langle\mathbf{w}_m\rangle^{\mathrm{T}}\langle\mathbf{x}_n\rangle\right),$$
$$\breve{a}_{\alpha_d} = a_{\boldsymbol{\alpha}} + \frac{1}{2}M,$$
$$\breve{b}_{\alpha_d} = b_{\boldsymbol{\alpha}} + \frac{1}{2}\sum_{m=1}^{M}\left\langle w_{md}^2\right\rangle,$$
$$\breve{a}_{\tau_m} = a_{\boldsymbol{\tau}} + \frac{1}{2}N_m,$$
$$\breve{b}_{\tau_m} = b_{\boldsymbol{\tau}} + \frac{1}{2}\sum_{n\in O_{mn}}\left\langle\left(y_{mn}-\mathbf{w}_m^{\mathrm{T}}\mathbf{x}_n-\mu_m\right)^2\right\rangle,$$

where O_{mn} is the set of indeces (m,n) for which the corresponding y_{mn} is not missing, and N_m is the number of non-missing observations in the m-th row of $\mathbf{Y}$.

2.2. Transformations of the posterior distributions

The two terms of $\mathcal{L}(q)$ in (3) suggest that the optimal approximation $q(\boldsymbol{\Theta})$ should provide good explanation of data, which is expressed in the likelihood term $\langle\log p(\mathbf{Y}|\boldsymbol{\Theta})\rangle$. It should also reflect our prior model because the second term is simply the negative of the Kullback-Leibler divergence between the prior distribution and the posterior approximation.

However, the maximization of $\mathcal{L}(q)$ can be quite slow because discarding posterior correlations between many variables in the posterior approximation often leads to zigzagging of the

update trajectories in the parameter space. This effect can be reduced by transformations of the model parameters, as we propose in the following. Such transformations can be performed after each iteration step of the EM learning algorithm or less frequently.

The additional computational cost is rather small because the transformations are performed in the lower-dimensional subspace of $\mathbf{x}_n$. One cycle of the VB updates has a computational cost of $O(DNM)$ without missing values and $O(D^2NM)$ in the presence of missing values, but the presented transformations do not even exceed $O(D^2N + D^2M)$ although they speed up the convergence significantly.

2.2.1. Removing the bias from $\mathbf{X}$

We note that one can move a constant bias term between $\mathbf{X}$ and $\boldsymbol{\mu}$ as in

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu} = \mathbf{W}(\mathbf{x}_n - \mathbf{b}) + (\mathbf{W}\mathbf{b} + \boldsymbol{\mu}) = \mathbf{W}\mathbf{x}_{n*} + \boldsymbol{\mu}_*\,, \quad (7)$$

where $\mathbf{b}$ is a $D \times 1$ bias vector. This motivates a transformation of the approximating posterior pdfs to

$$q_*(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\overline{\mathbf{x}}_n - \mathbf{b}, \boldsymbol{\Sigma}_{\mathbf{x}_n}),$$
$$q_*(\boldsymbol{\mu}) = \prod_{m=1}^{M} \mathcal{N}(\mu_m|\overline{\mu}_m + \overline{\mathbf{w}}_m^{\mathrm{T}}\mathbf{b}, \widetilde{\mu}_m),$$

where the bias $\mathbf{b}$ parameterizes the transformed distributions. Note that the original distributions are recovered by setting $\mathbf{b} = \mathbf{0}$.

The lower bound (3) to be maximized with respect to $\mathbf{b}$ becomes

$$\langle \log p(\mathbf{Y}|\boldsymbol{\Theta})\rangle_* - \left\langle \log \frac{q_*(\mathbf{X})}{p(\mathbf{X})} \right\rangle_* - \left\langle \log \frac{q_*(\boldsymbol{\mu})}{p(\boldsymbol{\mu})} \right\rangle_* + \text{const}\,, \quad (8)$$

where the expectation is taken over the transformed q_* distributions and the constant term represents the terms independent of $\mathbf{b}$.

As we show in the Appendix A, the bias term $\mathbf{b}$ that maximizes (8) can be reasonably approximated by

$$\mathbf{b} \approx \frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{x}}_n\,.$$

Thus, the expected mean of the latent variables $\mathbf{x}_n$ should be transformed to zero. In our experiments, we use this approximate translation. However, if a very large portion of the data $\mathbf{Y}$ is missing, one may obtain better performace with the exact formulae (18) as discussed in the Appendix A.

2.2.2. Rotation of the latent subspace

Similarly, one can rotate the latent variables $\mathbf{X}$ with a proper rotation of the loading matrix $\mathbf{W}$ such that the likelihood term $\langle \log p(\mathbf{Y}|\boldsymbol{\Theta})\rangle$ remains constant:

$$\mathbf{y}_n = \mathbf{W}\mathbf{x}_n + \boldsymbol{\mu} = (\mathbf{W}\mathbf{R})(\mathbf{R}^{-1}\mathbf{x}_n) + \boldsymbol{\mu} = \mathbf{W}_*\mathbf{x}_{n*} + \boldsymbol{\mu}\,.$$

This yields the transformed distributions parameterized with the $D \times D$ rotation matrix $\mathbf{R}$:

$$q_*(\mathbf{W}) = \prod_{m=1}^{M} \mathcal{N}(\mathbf{w}_m|\mathbf{R}^{\mathrm{T}}\overline{\mathbf{w}}_m, \mathbf{R}^{\mathrm{T}}\boldsymbol{\Sigma}_{\mathbf{w}_m}\mathbf{R})\,,$$
$$q_*(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_n|\mathbf{R}^{-1}\overline{\mathbf{x}}_n, \mathbf{R}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}_n}\mathbf{R}^{-\mathrm{T}})\,,$$
$$q_*(\boldsymbol{\alpha}) = \prod_{d=1}^{D} \mathcal{G}(\alpha_d|a_{\boldsymbol{\alpha}} + \frac{1}{2}M, b_{\boldsymbol{\alpha}} + \frac{1}{2}\mathbf{r}_d^{\mathrm{T}}\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W}\right\rangle \mathbf{r}_d)\,,$$

where $\mathbf{r}_d$ is the d-th column of $\mathbf{R}$. The transformed distribution $q_*(\boldsymbol{\alpha})$ is motivated by the update rule of $q(\boldsymbol{\alpha})$. Again, the original distributions can be recovered by setting $\mathbf{R} = \mathbf{I}$.

The lower bound (3) to be maximized can be written as a function of $\mathbf{R}$ as

$$-\left\langle \log \frac{q_*(\mathbf{X})}{p(\mathbf{X})} \right\rangle_* - \left\langle \log \frac{q_*(\mathbf{W})}{p(\mathbf{W}|\boldsymbol{\alpha})} \right\rangle_* - \left\langle \log \frac{q_*(\boldsymbol{\alpha})}{p(\boldsymbol{\alpha})} \right\rangle_* + \text{const}, \quad (9)$$

where the expectations are taken over the transformed q_* distributions. The constant term represents the terms independent of the rotation parameter $\mathbf{R}$.

As we show in Appendix B, the transformation matrix $\mathbf{R}$ which maximizes (9) can be found from the requirements

$$\frac{1}{N}\left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}}\right\rangle_* = \mathbf{I} \quad (10)$$
$$\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W}\right\rangle_* = \text{diagonal matrix.} \quad (11)$$

The resulting rotation matrix is formed as $\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}$ with orthogonal matrix $\mathbf{U}$ and diagonal matrix $\boldsymbol{\Lambda}$ found from the eigendecomposition

$$\mathbf{U}\boldsymbol{\Lambda}^2\mathbf{U}^{\mathrm{T}} = \frac{1}{N}\left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}}\right\rangle \quad (12)$$

and $\mathbf{V}$ is an orthogonal matrix computed from the eigendecomposition

$$\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}}\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W}\right\rangle \mathbf{U}\boldsymbol{\Lambda} = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}\,. \quad (13)$$

Thus, the transformation basically whitens the hidden states $\mathbf{x}_n$ and orthogonalizes the columns of the loading matrix $\mathbf{W}$ while taking into account the posterior uncertainties.

2.2.3. Relation to parameter expanded VB

Our speed-up methodology is closely related to the general framework of parameter expanded variational Bayesian (PX-VB) methods [23], which are similar to the parameter expansions for MCMC [18, 26] and EM algorithm [17]. The main difference is that we use proper parameterization of the approximate posterior distribution $q(\boldsymbol{\Theta})$ in order to optimize jointly the individual factors of the variational approximation. In contrast, in PX-VB one parameterizes the prior distributions and the likelihood function with the auxiliary variables which are then optimized to decrease the VB cost function.

Despite the differences, our methodology and PX-VB lead to the same transformations. The bias removal can be seen as

augmenting the model with auxiliary variable $\mathbf{b}$ by exchanging the original variables as

$$\begin{aligned}\mathbf{x}_n &= \mathbf{x}_{n*} - \mathbf{b},\\ \boldsymbol{\mu} &= \boldsymbol{\mu}_* + \langle \mathbf{W} \rangle \mathbf{b},\end{aligned}$$

resulting in a new prior and likelihood

$$\begin{aligned}p(\mathbf{X}_*|\mathbf{b}) &= \prod_{n=1}^{N} \mathcal{N}(\mathbf{x}_{n*}|\mathbf{b}, \mathbf{I}),\\ p(\boldsymbol{\mu}_*|\mathbf{b}) &= \mathcal{N}\left(\boldsymbol{\mu}_* \left| -\langle \mathbf{W} \rangle \mathbf{b}, \beta^{-1}\mathbf{I}\right.\right),\\ p(\mathbf{Y}|\boldsymbol{\Theta}) &= \prod_{(m,n)\in\mathcal{O}_{mn}} \mathcal{N}\left(y_{mn} \left| \mathbf{w}_m^{\mathrm{T}}(\mathbf{x}_{n*} - \mathbf{b}) + \boldsymbol{\mu}_* + \left\langle \mathbf{w}_m^{\mathrm{T}} \right\rangle \mathbf{b}, \mathrm{diag}(\boldsymbol{\tau})^{-1}\right.\right).\end{aligned}$$

First, the approximate posterior $q(\mathbf{W}, \mathbf{X}_*, \boldsymbol{\mu}_*, \tau, \boldsymbol{\alpha})$ is evaluated with the original model (i.e., $\mathbf{b} = 0$). Next, $\mathrm{KL}(q(\mathbf{W}, \mathbf{X}_*, \boldsymbol{\mu}_*, \tau, \boldsymbol{\alpha}) \| p(\mathbf{Y}, \mathbf{W}, \mathbf{X}_*, \boldsymbol{\mu}_*, \tau, \boldsymbol{\alpha}, \mathbf{b}))$ is minimized with respect to the auxiliary variable $\mathbf{b}$. It can be shown that this cost function is actually identical to (8), thus resulting in the same optimal translation.

Similar relation holds for the rotational transformation as the rotation can be interpreted as augmenting the model with an auxiliary variable $\mathbf{R}$ by exchanging the original variables as

$$\begin{aligned}\mathbf{W} &= \mathbf{W}_*\mathbf{R},\\ \mathbf{X} &= \mathbf{R}^{-1}\mathbf{X}_*,\\ \alpha_d^{-1} &= \mathbf{r}_d^{\mathrm{T}} \,\mathrm{diag}(\boldsymbol{\alpha}_*)^{-1}\mathbf{r}_d.\end{aligned}$$

Updating the prior and the likelihood appropriately, the minimization of $\mathrm{KL}\left(q(\mathbf{W}_*, \mathbf{X}_*, \boldsymbol{\mu}, \tau, \boldsymbol{\alpha}_*) \| p(\mathbf{Y}, \mathbf{W}_*, \mathbf{X}_*, \boldsymbol{\mu}, \tau, \boldsymbol{\alpha}_*, \mathbf{R})\right)$ with respect to $\mathbf{R}$ is identical to the maximization of (9).

The difference between PX-VB and our methodology is in the viewpoint rather than in the resulting transformations. The transformations to the posterior distribution can be seen as a model augmentation. However, we suggest that it can be more convenient to parameterize the approximate posterior distribution directly rather than indirectly through the prior and the likelihood. In some cases, it might be difficult to see what model augmentation corresponds to a particular parameterization. Instead of auxiliary variables, the transformations can be interpreted as parameterized joint optimization of multiple factors in $q(\boldsymbol{\Theta})$. For instance, the speeding up of VB with pattern searches can be seen as one special case of doing parameterized joint optimization [13].

3. The PCA solution for probabilistic PCA models

The results reported in the previous section suggest that the solution found by VBPCA always satisfies the requirements that the principal components are zero-mean and mutually uncorrelated and the loading matrix $\mathbf{W}$ has mutually orthogonal columns:

$$\frac{1}{N}\sum_{n=1}^{N} \overline{\mathbf{x}}_{n*} = 0 \tag{14}$$

$$\frac{1}{N}\left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle = \mathbf{I} \tag{15}$$

$$\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle = \mathrm{diag}(\mathbf{s}) \tag{16}$$

$$s_k \geq s_l\,, \quad k < l\,, \tag{17}$$

where $\mathrm{diag}(\mathbf{s})$ denotes a diagonal matrix with diagonal elements s_k. Here, we ordered the columns of the loading matrix $\mathbf{W}$ according to their expected norms in order to remove the remaining ambiguity.

Thus, the VBPCA solution allows intuitive interpretation similar to standard PCA: The principal components are ordered according to the amount of data variance they explain, which is estimated using the computed posterior approximations. In this case, the normalized columns of $\mathbf{W}$ and their squared norms s_k play the role of the eigenvectors and eigenvalues of classical PCA.

Similarly to this, we can define the PCA basis for related probabilistic models which converge to some basis in the principal subspace and therefore have rotational ambiguity. For example, it is easy to show (see Appendix C) that the solution provided by probabilistic PCA [24] always satisfies at least (14) and (15). We can use the additional requirements (16)-(17) in order to define a practically unique solution. Again, this solution allows interpretation similar to standard PCA.

One can perform *explicitly* a transformation of the solution provided by [24] such that the requirements (14)–(17) are fulfiled. The only difference to the transformations discussed in the previous section would be using point-estimated $\mathbf{W}$, which yields, for example, $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ instead of $\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle$ in (13). Performing such a transformation can also speed up learning, which is motivated by the variational view of the EM-algorithm, as we discuss in Appendix C.

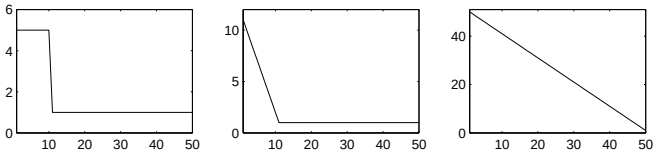
Similar transformations to the PCA basis are straightforward for principal subspace methods which compute point estimates for parameters $\{\mathbf{W}, \mathbf{x}_n, \boldsymbol{\mu}\}$ of model (1) and which use the only assumption that the noise term $\boldsymbol{\epsilon}_n$ is Gaussian with fixed variance. One simply needs to replace $\left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle$ and $\left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle$ with point estimates $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ and $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ in (12) and (13).

The formulation of the PCA solution in terms of (14)–(17) allows to extend the idea of the PCA basis to the case when data vectors $\mathbf{y}_n$ have missing values. There is an important difference, though: The c-dimensional PCA basis found for complete data has the property that the first $k < c$ columns of $\mathbf{W}$ always correspond to the k-dimensional PCA solution. This property does not generally hold for incomplete data: The principal subspace estimated with the same algorithm using fewer components may differ from the leading directions of the PCA basis found in the subspace with more components.

4. Experiments

4.1. Artificial data

In this section, we use an artificial experiment to illustrate the effect of the transformations on the speed of convergence of VBPCA. We generated three different datasets with $N = 200$ data points from the multivariate Gaussian distribution with



(a) First dataset (b) Second dataset (c) Third dataset

Figure 1: The square roots of the eigenvalues of the covariance matrices which were used to generate the three datasets.

$M = 50$ dimensions with the means drawn from a Gaussian distribution with zero mean and unit variance. The covariance matrix for the first dataset contained ten eigenvalues of 5^2 and the rest of the eigenvalues equaled unity (see Fig. 1). The covariance matrix for the second dataset had ten larger eigenvalues $2^2, \ldots, 11^2$ and the remaining eigenvalues equaled unity. The third covariance matrix had eigenvalues $1^2, \ldots, 50^2$ thus having no prominent low-dimensional subspace. 20% of the values were removed from the datasets. Those data points were used for validation of trained models in the problem of missing value reconstruction.

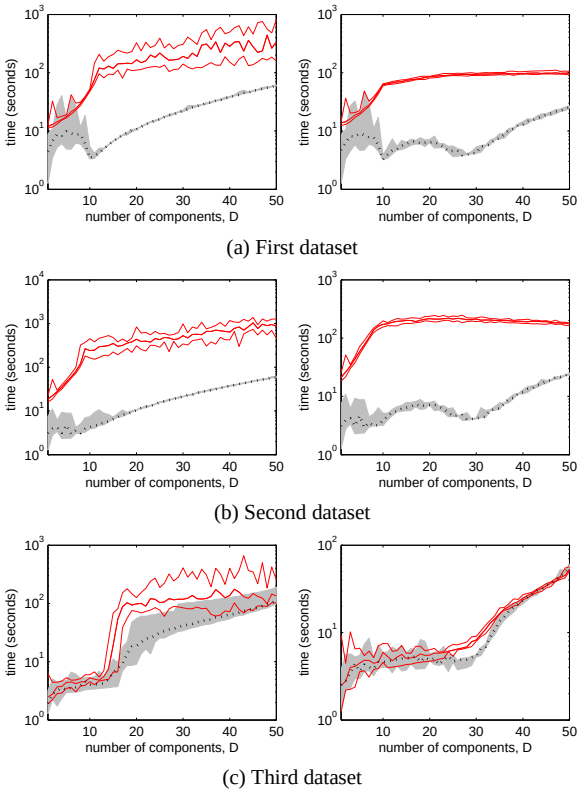
The VBPCA model was trained using the variational EM algorithm presented in [8]. We ran the experiments for two variants of the prior for $\mathbf{W}$: 1) the hierarchical prior explained in Section 2 and 2) the broad prior which was achieved by fixing hyperparameters α_d to small values.[1] For each dataset, we learned the model with all possible dimensionalities for the latent subspace, ranging in $D = 1, \ldots, 50$. We measured the running time until convergence, which was the point when the relative relative difference to the converged value of the log-likelihood lower bound was less than 10^{-3}.

In all experiments, we initialized the hyperparameter τ defining the inverse variance of the noise in (2) to a large value. This was done to avoid underfitting when the noise variance is estimated to be too large and many components are estimated to have zero variance.

Fig. 2 shows the results for the three datasets and for ten experiments with and without applying the proposed transformations after each iteration of the variational EM-algorithm. For the first two datasets, the convergence of the algorithm when the transformations were used was approximately ten times faster depending on the dimensionality of the latent subspace. However, the transformations had small or almost no effect in the third experiment when the data had no prominent latent subspace. We also see that the transformations become more important for models with a larger number of latent components.

We also observed a greater significance of the transformations for higher-dimensional datasets. For 200-dimensional Gaussian data, the convergence with the transformations was 1000 times faster than without them. In that experiment, the covariance matrix had eigenvalues $21^2, 20^2, \ldots, 2^2, 1, 1, \ldots, 1$, the number of samples was $N = 2000$ and the number of estimated components was $D = 50$.

[1] The optimal transformation in case of broad priors is somewhat different from the one presented in Section 2, as we explain in Section 5.



(a) First dataset

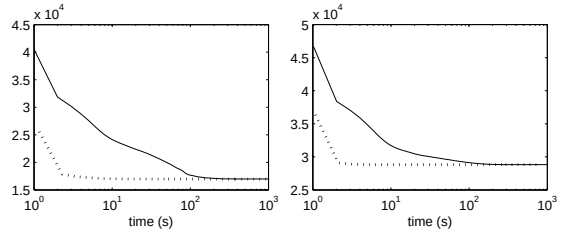
(b) Second dataset

(c) Third dataset

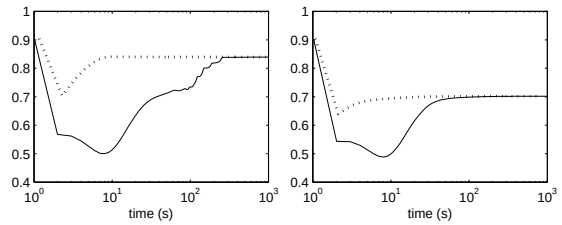
Figure 2: Convergence time for three different artificial datasets using hierarchichal prior (left) and fixed broad prior (right) for $\mathbf{W}$. The x-axis corresponds to different number of latent components. The black dotted curve with the shaded area and the three solid red curves show the worst, the best and the median convergence times out of ten experiments with and without the transformations respectively.

More detailed analysis of the results suggests that performing the transformation can reduce the overfitting effect during learning. Fig. 3 shows the VB cost (i.e., the negative of (3)) and the root mean squared error (RMSE) $(\sum_{mn}(y_{mn} - \overline{\mathbf{w}}_m^{\mathrm{T}}\overline{\mathbf{x}}_n - \overline{\mu}_m)^2)^{\frac{1}{2}}$ for the training set during the learning. RMSE is estimated to be too small at the beginning and it is later increased to recover from an overfitted solution in order to decrease the RMSE for the test set in Fig. 3c. This overfitting effect is smaller when the proposed transformations are used. For both hierarchical and broad priors, the convergence was 10–100 faster with the transformations.

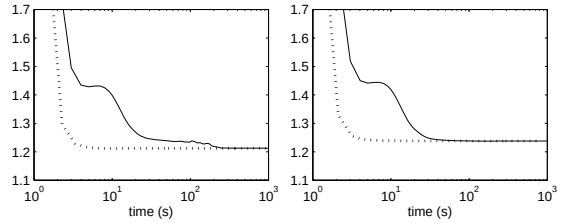
When the noise variance was initialized to be large, that is, a small value was used for τ in (2), the improvement of convergence obtained using the transformations can be mild. However, in our experiments, this type of initialization typically led to slower convergence compared to the initialization with small noise variance. Also, the risk of converging to a bad local optimum was increased: Initialization with large variance of observation noise may lead to pruning out some of the components. In this case, using the transformations may speed up the pruning process, which can be a negative effect. Therefore, initialization



(a) VB cost

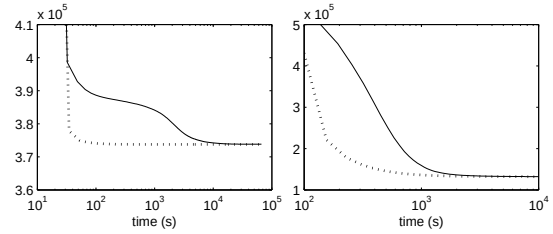


(b) Training RMSE

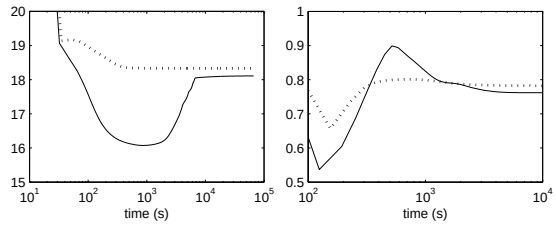


(c) Test RMSE

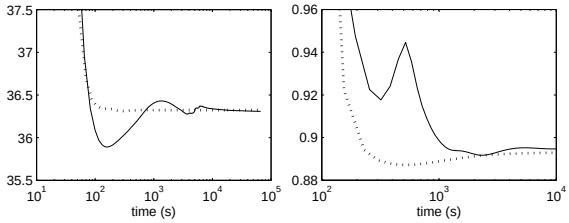
Figure 3: Experimental results obtained for an artificial dataset when using hierarchichal prior (left) or fixed broad prior (right) for $\mathbf{W}$. The dotted and solid curves represent the results with and without the transformations respectively. The VB cost is minus lower bound of the log marginal likelihood (3), that is, the VB cost is minimized during learning.



(a) VB cost



(b) Training RMSE



(c) Test RMSE

Figure 4: Experimental results obtained for MNIST (left) and MovieLens (right) datasets. The dotted and solid curves represent the results with and without the transformations respectively. The VB cost is minus lower bound of the log marginal likelihood, that is, the VB cost is minimized during learning.

of noise variance with small values seems to be a more robust approach. Although it is possible to converge to a poorer local optimum with the transformations, this happened very rarely in our experiments. If this appears to be a problem, it is possible to start using the transformations after a few cyclic updates have been completed.

4.2. Real-world data

We also tested the effect of the transformations on two real-world datasets. The first dataset consisted of images of handwritten digits, extracted from the MNIST database[2]. We chose $N = 100$ images of digit 5 as the training data. Images are in grayscale and have a size of 28×28 pixels, resulting in dimensionality $M = 784$. The second dataset consisted of movie ratings from a set of users extracted from the MovieLens database[3]. We used the smallest dataset, which had 100,000 ratings for $N = 1682$ movies by $M = 943$ users, resulting in an extremely sparse matrix.

[2]Available online at http://yann.lecun.com/exdb/mnist/.

[3]Available online at http://www.grouplens.org/node/73.

For both datasets, we discarded 20% of the data and used that as a test set. We used $D = 50$ and $D = 100$ components for MNIST and MovieLens datasets respectively. In both experiments, we used a hierarchical prior for $\mathbf{W}$, as presented in Section 2.

Fig. 4 shows the results for both datasets obtained with and without the transformations. In both cases, the transformations made the convergence faster and more stable. Learning was 100 and 10 times faster for MNIST and MovieLens respectively. One can also notice that the algorithm converged to two different solutions for the MNIST dataset and the solution obtained without the transformations was slightly better. However, this result should not be interpreted as a drawback of using the transformations.

Note also that the convergence of the runs without the transformations can get stuck in a difficult region of the optimized parameter space (see the l.h.s. plots in Fig. 4). This introduces a risk of too early stopping when an automatic stopping criterion is used. On the contrary, the runs with the transformations seem to avoid this problem: in all our experiments they converged very fast to the vicinity of a local optimum.

5. Conclusions and discussion

In this paper, we showed how simple transformations of the latent space can speed up leaning of variational Bayesian factor analysis models. The presented approach resembles the more general idea of using auxiliary parameters in VB learning [23]. We gave theoretical justification and showed experimentally that the proposed transformations can significantly improve the rate of convergence. The transformations become more significant for large-scale datasets and larger number of components. Fast convergence is especially important when one needs the computed lower bound of the log marginal likelihood to do model comparison. Another possible approach to speeding up learning of variational PCA is explicitly incorporating the orthogonality restrictions into the model [27].

Apart from a possible speed-up of learning, the proposed transformation produces an intuitive representation of the latent space, similarly to standard PCA. This allows, for example, to use the computed means $\overline{\mathbf{x}}_n$ as the analogue of the principal components in algorithms which require preprocessing with whitening (see, e.g. [14]).

Similar transformations can improve the algorithms for other related variational Bayesian latent variable models. The exact formulae for other models can be derived using the presented methodology. For example, when the model in Section 2 is restricted to have fixed broad prior for $\mathbf{W}$, that is, hyperparameters α_d are fixed to some small values, the optimal rotation can be shown to yield $\frac{1}{N-M}\left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle_* = \mathbf{I}$, with $N > M$, instead of (10).

Transformations can easily be derived for robust PCA models which use heavy-tailed Student-t distribution for the observation noise [19] and for latent components $\mathbf{X}$ as well [1]. When the Gaussian prior is used to describe $\mathbf{X}$ [19], the rotation defined by (10) and (11) remains optimal. When the prior model for the hidden components is described using the multivariate Student-t distribution, one can use a hierarchical prior model

$$p(\mathbf{X}) = \prod_{n=1}^{N} \mathcal{S}(\mathbf{x}_n|0, \mathbf{I}, \nu) = \prod_{n=1}^{N} \int \mathcal{N}\left(\mathbf{x}_n|0, u_n^{-1}\mathbf{I}\right) \mathcal{G}\left(u_n|\tfrac{\nu}{2}, \tfrac{\nu}{2}\right) du_n,$$

where ν denotes the degrees of freedom and u_n are auxiliary latent variables. Integrating out variables u_n results in the Student-t prior for $\mathbf{X}$. Then, the optimal rotation can be shown to yield

$$\frac{1}{N}\left\langle \mathbf{X}\,\mathrm{diag}(\mathbf{u})\mathbf{X}^{\mathrm{T}} \right\rangle_* = \mathbf{I}\,,$$

which should be used instead of (10). Here, diag($\mathbf{u}$) is a diagonal matrix with elements u_n on its diagonal.

The presented ideas might be extended to VB learning of other factor analysis models in which the latent variables appear in the data model in the form $\mathbf{WX}$, as in (2). For example, a relevant transformation for blind source separation methods based on dynamical generative models [9] might be applying fast separation algorithms [28] during learning. Similar ideas might also be used in nonlinear and mixture models as well [5, 10, 12, 21, 25].

Acknowledgements

This work was supported in part by the Academy of Finland under the Centers for Excellence in Research Program and Alexander Ilin's posdoctoral research project and the IST Program of the European Community, under the PASCAL2 Network of Excellence.

References

[1] Archambeau, C., Delannay, N., Verleysen, M., 2006. Robust probabilistic projections. In: Proceedings of the 23rd International Conference on Machine Learning (ICML'06). pp. 33–40.

[2] Attias, H., 1999. Independent factor analysis. Neural Computation 11 (4), 803–851.

[3] Bartholomew, D. J., 1987. Latent Variable Models and Factor Analysis. Charles Griffin & Co. Ltd., London.

[4] Basilevsky, A., 1994. Statistical Factor Analysis and Related Methods. Wiley, New York.

[5] Beal, M. J., May 2003. Variational algorithms for approximate Bayesian inference. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

[6] Bishop, C., 1999. Latent variable models. In: Jordan, M. (Ed.), Learning in Graphical Models. The MIT Press, Cambridge, MA, USA, pp. 371–403.

[7] Bishop, C., 2006. Pattern Recognition and Machince Learning. Springer, Cambridge.

[8] Bishop, C. M., 1999. Variational principal components. In: Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99). pp. 509–514.

[9] Cichocki, A., Thawonmas, R., 2000. On-line algorithm for blind signal extraction of arbitrarily distributed, but temporally correlated sources using second order statistics. Neural Processing Letters 12, 91–98.

[10] Ghahramani, Z., Hinton, G. E., 1996. The EM algorithm for mixtures of factor analyzers. Tech. Rep. CRG-TR-96-1, University of Toronto, Toronto, Canada.

[11] Ghahramani, Z., Hinton, G. E., 1996. Parameter estimation for linear dynamical systems. Tech. Rep. CRG-TR-96-2, University of Toronto, Toronto, Canada.

[12] Honkela, A., Valpola, H., 2005. Unsupervised variational Bayesian learning of nonlinear models. In: Saul, L. K., Weiss, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA, pp. 593–600.

[13] Honkela, A., Valpola, H., Karhunen, J., 2003. Accelerating cyclic update algorithms for parameter estimation by pattern searches. Neural Processing Letters 17 (2), 191–203.

[14] Hyvärinen, A., Karhunen, J., Oja, E., 2001. Independent Component Analysis. J. Wiley.

[15] Lappalainen, H., Honkela, A., 2000. Bayesian nonlinear independent component analysis by multi-layer perceptrons. In: Girolami, M. (Ed.), Advances in Independent Component Analysis. Springer-Verlag, Berlin, pp. 93–121.

[16] Lawrence, N., November 2005. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. Journal of Machine Learning Research 6, 1783–1816.

[17] Liu, C., Rubin, D. B., Wu, Y. N., 1998. Parameter expansion to accelerate EM: the PX-EM algorithm. Biometrika 85, 755–770.

[18] Liu, J. S., Wu, Y. N., 1999. Parameter expansion for data augmentation. Journal of the American Statistical Association 94, 1264–1274.

[19] Luttinen, J., Ilin, A., Karhunen, J., 2009. Bayesian robust PCA for incomplete data. In: Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation (ICA'2009). Springer-Verlag, pp. 66–73.

[20] Luttinen, J., Ilin, A., Raiko, T., 2009. Transformations for variational factor analysis to speed up learning. In: Verleysen, M. (Ed.), Proceedings of the 17th European Symposium on Artificial Neural Networks (ESANN'2009). d-side, pp. 77–82.

[21] Mohamed, S., Heller, K., Ghahramani, Z., 2009. Bayesian exponential family PCA. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (Eds.), Advances in Neural Information Processing Systems 21. MIT Press, Cambridge, MA, pp. 1089–1096.

[22] Neal, R. M., Hinton, G. E., 1999. A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M. I. (Ed.), Learning in Graphical Models. The MIT Press, Cambridge, MA, USA, pp. 355–368.
[23] Qi, Y. A., Jaakkola, T. S., 2007. Parameter expanded variational Bayesian methods. In: Schölkopf, B., Platt, J., Hoffman, T. (Eds.), Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA, pp. 1097–1104.
[24] Tipping, M. E., Bishop, C. M., 1999. Probabilistic principal component analysis. Journal of the Royal Statistical Society Series B 61 (3), 611–622.
[25] Valpola, H., Karhunen, J., 2002. An unsupervised ensemble learning method for nonlinear dynamic state-space models. Neural Computation 14 (11), 2647–2692.
[26] van Dyk, D. A., Meng, X.-L., 2001. The art of data augmentation (with discussion). Journal of Computational and Graphical Statistics 10, 1–111.
[27] Šmidl, V., Quinn, A., September 2003. Fast variational PCA for functional analysis of dynamic image sequences. In: Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPA'2003). Vol. 1. pp. 555–560.
[28] Ziehe, A., Müller, K.-R., 1998. TDSEP — an effective algorithm for blind separation using time structure. In: Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN '98). Skövde, Sweden, pp. 675–680.

A. Derivation of the translation

In this section, we derive the optimal translation $\mathbf{b}$ which maximizes the log-likelihood lower bound in (8). If $\boldsymbol{\mu}$ has a broad prior, the term $\langle \log p(\boldsymbol{\mu}) \rangle_*$ is constant. The terms $-\langle \log q(\mathbf{X}) \rangle_*$ and $-\langle \log q(\boldsymbol{\mu}) \rangle_*$ are entropies of Gaussian distributions, which are in general constant with respect to the mean parameter. Therefore these terms are also constant with respect to $\mathbf{b}$. Removing the constant terms, the following remaining terms form the non-constant part of the lower bound:

$$\langle \log p(\mathbf{Y}|\boldsymbol{\Theta}) \rangle_* = -\frac{\langle \tau \rangle}{2} \sum_{n=1}^{N} (\overline{\mathbf{x}}_n - \mathbf{b})^{\mathrm{T}} \sum_{m \in O_{mn}} \boldsymbol{\Sigma}_{\mathbf{w}_m} (\overline{\mathbf{x}}_n - \mathbf{b}) + \text{const},$$

$$\langle \log p(\mathbf{X}) \rangle_* = -\frac{1}{2} \sum_{n=1}^{N} (\overline{\mathbf{x}}_n - \mathbf{b})^{\mathrm{T}} (\overline{\mathbf{x}}_n - \mathbf{b}) + \text{const},$$

where O_{mn} is the set of indices m for which y_{mn} is observed, that is, missing values are ignored. Taking the derivative with respect to $\mathbf{b}$ and equating the result to zero yields the optimal transformation

$$\mathbf{b} = \left(\sum_{n=1}^{N} \boldsymbol{\Psi}_n \right)^{-1} \left(\sum_{n=1}^{N} \boldsymbol{\Psi}_n \overline{\mathbf{x}}_n \right), \tag{18}$$

where $\boldsymbol{\Psi}_n = \mathbf{I} + \sum_{m \in O_{mn}} \langle \tau \rangle \boldsymbol{\Sigma}_{\mathbf{w}_m}$. This formula can be approximated in order to reduce the computational cost. Assuming that $\boldsymbol{\Psi}_n$ is approximately constant with respect to n, that is there are few missing values or the posterior covariance $\boldsymbol{\Sigma}_{\mathbf{w}_m}$ is small, $\boldsymbol{\Psi}_n$ is cancelled out and the formula reduces to

$$\mathbf{b} \approx \frac{1}{N} \sum_{n=1}^{N} \overline{\mathbf{x}}_n .$$

B. Derivation of the rotation

In this section, we derive the optimal rotation $\mathbf{R}$ which maximizes the log-likelihood lower bound in (9). Assuming broad prior for $\boldsymbol{\alpha}$ (i.e., $a_{\boldsymbol{\alpha}} \to 0$ and $b_{\boldsymbol{\alpha}} \to 0$), the fourth term in (9) is constant. Thus, the following terms remain non-constant:

$$\begin{aligned}
\langle \log p(\mathbf{X}) \rangle_* &= -\frac{1}{2} \operatorname{tr}\left(\mathbf{R}^{-1} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle \mathbf{R}^{-\mathrm{T}} \right) + \text{const}, \\
-\langle \log q(\mathbf{X}) \rangle_* &= \frac{1}{2} \sum_{n=1}^{N} \log \left| \mathbf{R}^{-1} \boldsymbol{\Sigma}_{\mathbf{x}_n} \mathbf{R}^{-\mathrm{T}} \right| + \text{const} \\
&= -N \log |\mathbf{R}| + \text{const}, \\
\langle \log p(\mathbf{W}|\boldsymbol{\alpha}) \rangle_* &= \frac{M}{2} \sum_{d=1}^{D} \langle \log \alpha_d \rangle_* - \\
&\quad \frac{1}{2} \operatorname{tr}\left(\operatorname{diag} \langle \boldsymbol{\alpha} \rangle_* \mathbf{R}^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{R} \right) + \text{const}, \\
-\langle \log q(\mathbf{W}) \rangle_* &= \frac{1}{2} \sum_{m=1}^{M} \log \left| \mathbf{R}^{\mathrm{T}} \boldsymbol{\Sigma}_{\mathbf{w}_m} \mathbf{R} \right| + \text{const} \\
&= M \log |\mathbf{R}| + \text{const}.
\end{aligned}$$

The broad prior for $\boldsymbol{\alpha}$ yields $\langle \alpha_d \rangle_* \approx M / \left(\mathbf{r}_d^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{r}_d \right)$, and therefore the latter term in $\langle \log p(\mathbf{W}|\boldsymbol{\alpha}) \rangle_*$ is a constant:

$$\begin{aligned}
&-\frac{1}{2} \operatorname{tr}\left(\operatorname{diag} \langle \boldsymbol{\alpha} \rangle_* \mathbf{R}^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{R} \right) \\
&= -\frac{1}{2} \sum_{d=1}^{D} \langle \alpha_d \rangle_* \mathbf{r}_d^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{r}_d = -\frac{1}{2} \sum_{d=1}^{D} M = \text{const},
\end{aligned}$$

and can be discarded.

We represent $\mathbf{R}$ using its singular value decomposition as $\mathbf{R} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\boldsymbol{\Lambda}$ is diagonal. Matrix $\mathbf{V}$ cancels out in most of the terms affecting only the term

$$\frac{M}{2} \sum_{d=1}^{D} \langle \log \alpha_d \rangle_* \approx -\frac{M}{2} \log \prod_{d=1}^{D} \frac{1}{2} \mathbf{r}_d^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{r}_d + \text{const}.$$

Thus, $\mathbf{V}$ can be found by maximizing this term, which is equivalent to minimization of the product of the diagonal elements of $\mathbf{V}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}$. For a positive definite and symmetric matrix, the product of the diagonal elements is bounded below by the determinant[4] and it equals the determinant if the matrix is diagonal. Since an orthogonal rotation does not change the determinant, the optimal $\mathbf{V}$ is obtained when

$$\mathbf{V}^{\mathrm{T}}\boldsymbol{\Lambda}\mathbf{U}^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{U}\boldsymbol{\Lambda}\mathbf{V} = \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle = \text{diagonal matrix}.$$

Now, applying the result that the product of the diagonal elements equals the determinant, we obtain

$$\begin{aligned}
\frac{M}{2} \sum_{d=1}^{D} \langle \log \alpha_d \rangle_* &\approx -\frac{M}{2} \log \left| \mathbf{R}^{\mathrm{T}} \left\langle \mathbf{W}^{\mathrm{T}}\mathbf{W} \right\rangle \mathbf{R} \right| + \text{const} \\
&= -M \log |\mathbf{R}| + \text{const},
\end{aligned}$$

[4] This can be seen using the Cholesky decomposition of a positive definite and symmetric matrix $\mathbf{C} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$, where $\mathbf{L}$ is lower triangular, and therefore $|\mathbf{L}\mathbf{L}^{\mathrm{T}}| = |\mathbf{L}|^2 = \prod_{d=1}^{D} l_{dd}^2 \leq \prod_{d=1}^{D} \sum_{i=1}^{d} l_{id}^2$ which is the product of the diagonal elements of $\mathbf{C}$.

which cancels out $-\langle \log q(\mathbf{W}) \rangle_*$. Therefore the lower bound maximized w.r.t. $\mathbf{U}$ and $\mathbf{\Lambda}$ simplifies to

$$\begin{aligned} &\langle \log p(\mathbf{X}) \rangle_* - \langle \log q(\mathbf{X}) \rangle_* + \text{const} \\ &= -\frac{1}{2} \operatorname{tr}\left((\mathbf{U}\mathbf{\Lambda})^{-1} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle (\mathbf{U}\mathbf{\Lambda})^{-\mathrm{T}} \right) - N \log |\mathbf{U}\mathbf{\Lambda}| + \text{const}\,. \end{aligned}$$

Equating the derivative w.r.t. $\mathbf{U}\mathbf{\Lambda}$ to zero yields the following requirements for the matrices $\mathbf{U}$ and $\mathbf{\Lambda}$

$$\mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^{\mathrm{T}} = \frac{1}{N} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle$$

or equivalently

$$\frac{1}{N} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle_* = \mathbf{V}^{\mathrm{T}}\mathbf{\Lambda}^{-1}\mathbf{U}^{\mathrm{T}} \frac{1}{N} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle \mathbf{U}\mathbf{\Lambda}\mathbf{V} = \mathbf{I}\,.$$

C. Rotation to the PCA basis for Probabilistic PCA

The variational view of the EM algorithm [22] allows for an interpretation of the learning algorithm for probabilistic PCA [24] in which the following function

$$\mathcal{F}(\mathbf{W}, \boldsymbol{\mu}, \tau, q(\mathbf{X})) = \langle \log p(\mathbf{Y}|\mathbf{W}, \mathbf{X}, \boldsymbol{\mu}, \tau) \rangle - \left\langle \log \frac{q(\mathbf{X})}{p(\mathbf{X})} \right\rangle, \quad (19)$$

is maximized w.r.t. to the model parameters $\mathbf{W}$, $\boldsymbol{\mu}$, τ and the pdf $q(\mathbf{X})$ which is defined as in (4).

We consider the same transformations (7), (2.2.2) which do not change the first term in (19). The second term in (19) is minus Kullback-Leibler divergence between $q(\mathbf{X})$ and $p(\mathbf{X})$:

$$\begin{aligned} D = \left\langle \log \frac{q(\mathbf{X})}{p(\mathbf{X})} \right\rangle &= \sum_{n=1}^{N} \int q(\mathbf{x}_n) \log \frac{q(\mathbf{x}_n)}{p(\mathbf{x}_n)} d\mathbf{x}_n \\ &= \frac{1}{2} \sum_{n=1}^{N} \left[\operatorname{tr}(\mathbf{\Sigma}_{\mathbf{x}_n}) + \overline{\mathbf{x}}_n^{\mathrm{T}} \overline{\mathbf{x}}_n - \log |\mathbf{\Sigma}_{\mathbf{x}_n}| \right] \end{aligned} \quad (20)$$

because $p(\mathbf{x}_n) = \mathcal{N}(\mathbf{x}_n|0, \mathbf{I})$. Transformation (7) changes (20) to

$$D = \frac{1}{2} \sum_{n=1}^{N} \left[\operatorname{tr}(\mathbf{\Sigma}_{\mathbf{x}_n}) + (\overline{\mathbf{x}}_n - \mathbf{b})^{\mathrm{T}} (\overline{\mathbf{x}}_n - \mathbf{b}) - \log |\mathbf{\Sigma}_{\mathbf{x}_n}| \right].$$

Now taking the derivative w.r.t. $\mathbf{b}$ and equating it to zero yields

$$\mathbf{b} = \frac{1}{N} \sum_{n=1}^{N} \overline{\mathbf{x}}_n\,.$$

Similarly, transformation (2.2.2) with $\mathbf{A} = \mathbf{R}^{-1}$ gives

$$\begin{aligned} D &= \frac{1}{2} \sum_{n=1}^{N} \left[\operatorname{tr}(\mathbf{A}\mathbf{\Sigma}_{\mathbf{x}_n}\mathbf{A}^{\mathrm{T}}) + \overline{\mathbf{x}}_n^{\mathrm{T}} \mathbf{A}^{\mathrm{T}} \mathbf{A} \overline{\mathbf{x}}_n - \log \left| \mathbf{A}\mathbf{\Sigma}_{\mathbf{x}_n}\mathbf{A}^{\mathrm{T}} \right| \right] \\ &= \frac{1}{2} \operatorname{tr}(\mathbf{A} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle \mathbf{A}^{\mathrm{T}}) - \frac{1}{2} \sum_{n=1}^{N} \log \left| \mathbf{A}\mathbf{\Sigma}_{\mathbf{x}_n}\mathbf{A}^{\mathrm{T}} \right|. \end{aligned}$$

Taking the derivative w.r.t. $\mathbf{A}$ gives

$$\mathbf{A} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle - N(\mathbf{A}^{\mathrm{T}})^{-1} = 0\,,$$

which implies that the following holds for optimal $\mathbf{A}$:

$$\mathbf{A} \frac{1}{N} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle \mathbf{A}^{\mathrm{T}} = \frac{1}{N} \left\langle \mathbf{X}\mathbf{X}^{\mathrm{T}} \right\rangle = \mathbf{I}\,.$$