Transformations for Variational Factor Analysis to Speed up Learning

Jaakko Luttinen, Alexander Ilin, Tapani Raiko

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

Abstract. We propose simple transformation of the hidden states in variational Bayesian (VB) factor analysis models to speed up the learning procedure. The transformation basically performs centering and whitening of the hidden states taking into account the posterior uncertainties. The transformation is given a theoretical justification from optimisation of the VB cost function. We derive the transformation formulae for variational Bayesian principal component analysis and show experimentally that it can significantly improve the rate of convergence. Similar transformations can be applied to other variational Bayesian factor analysis models as well.

1 Introduction

Bayesian methods provide a principled way for learning models. Several basic models use the assumption that the observed data vectors $\boldsymbol{y}_n$ are constructed from hidden states $\boldsymbol{x}_n$ and they use the Gaussian distribution to construct the prior for the states $\boldsymbol{x}_n$. Examples include probabilistic principal component analysis (PCA) [1, 2] with

$$\boldsymbol{y}_n = \boldsymbol{W}\boldsymbol{x}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n \,, \tag{1}$$

as well as its generalisations to exponential family [3] and nonlinear mappings [4, 5]. We generalise these variants under the term factor analysis models.

In the fully Bayesian approach, the joint distribution over all the variables is modelled. This often leads to intractable integrals and high computational cost. These issues can be solved, for instance, using computationally efficient variational Bayesian (VB) methods, which often approximate the posterior distribution by factorising it with respect to groups of variables. In the VB EM algorithm, the approximate distributions are learned iteratively by updating one group at a time. However, the variables in the model (1) are strongly coupled, which causes the VB iteration to converge slowly.

Faster convergence can be achieved by using parameter expanded VB methods [6]. The general idea is to use some auxiliary variables to reduce the couplings between variables in the original model. In this paper, we demonstrate a similar idea for VB factor analysis models and show that it can sinificantly speed up convergence. We propose simple transformations of the hidden states $\boldsymbol{x}_n$ and parameters $\boldsymbol{W}$, $\boldsymbol{\mu}$ which are derived from the minimisation of the VB cost function. We use the VB PCA model [2] as an example but the same type of transformations can be applied to similar models as well.

2 Variational Bayesian PCA

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

$$p(\boldsymbol{Y}|\boldsymbol{W},\boldsymbol{X},\boldsymbol{\mu},\tau) = \prod_n \mathcal{N}\left(\boldsymbol{y}_n|\boldsymbol{W}\boldsymbol{x}_n + \boldsymbol{\mu}, \tau^{-1}\boldsymbol{I}\right) ,$$

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 $\mathbf{C}$, $\boldsymbol{W}$ is an $M \times D$ loading matrix, $\boldsymbol{\mu}$ is a bias term and τ^{-1} represents the precision parameter of the Gaussian isotropic noise.

The prior models for the unknown variables are

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

where the hyperparameters β, $a_{\boldsymbol{\alpha}}$, $b_{\boldsymbol{\alpha}}$, a_τ, and b_τ are fixed to proper (small) values, e.g., 10^{-3}.

In VB PCA, the joint posterior density function is approximated with a density function q which is often a product of the following factors [2]:

$$q(\boldsymbol{x}_n) = \mathcal{N}(\boldsymbol{x}_n|\overline{\boldsymbol{x}}_n, \boldsymbol{\Sigma}_{\boldsymbol{x}_n}), \quad n = 1,\ldots,N$$
$$q(\boldsymbol{w}_m) = \mathcal{N}(\boldsymbol{w}_m|\overline{\boldsymbol{w}}_m, \boldsymbol{\Sigma}_{\boldsymbol{w}_m}), \quad q(\mu_m) = \mathcal{N}(\mu_m|\overline{\mu}_m, \widetilde{\mu}_m), \quad m = 1,\ldots,M$$
$$q(\tau) = \mathcal{G}(\tau|\breve{a}_\tau, \breve{b}_\tau), \qquad q(\alpha_d) = \mathcal{G}(\alpha_d|\breve{a}_{\alpha_d}, \breve{b}_{\alpha_d}), \quad d = 1,\ldots,D ,$$

where $\boldsymbol{w}_m$ are the rows of $\boldsymbol{W}$, $\mathcal{G}(\chi|a,b)$ are Gamma density functions which have the expectations $\langle\chi\rangle = a/b$ and $\langle\log\chi\rangle = \psi(a) - \log(b)$, with $\psi(a)$ the digamma function. The factors of the approximate distribution q are found iteratively by maximising the lower bound of the marginal likelihood

$$\mathcal{L}(q) = \int q(\boldsymbol{\Theta}) \log \frac{p(\boldsymbol{Y},\boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \,\mathrm{d}\,\boldsymbol{\Theta} = \left\langle \log p(\boldsymbol{Y}|\boldsymbol{\Theta}) - \log \frac{q(\boldsymbol{\Theta})}{p(\boldsymbol{\Theta})} \right\rangle , \qquad (2)$$

where $\boldsymbol{\Theta}$ represents the set of all the variables and $\langle\cdot\rangle$ denotes the expectation over the q distribution.

3 Transformations for speeding up VB PCA

The maximisation of the function (2) can be seen to consist of two parts: making the data more likely and minimising the Kullback-Leibler divergence between the prior and approximate posterior distributions. Thus, the lower bound can be improved by transforming the q distributions closer to the prior distributions if

this does not affect the reconstruction of the data. This gives an insight to the transformations presented in the following. The proposed transformations can be performed iteratively during learning. The additional computational cost is rather small because the transformations are performed in the lower-dimensional subspace of $\boldsymbol{x}_n$.

3.1 Translation of $\boldsymbol{X}$ and $\boldsymbol{\mu}$

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

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

Motivated by this relation, we consider the following transformation:

$$\begin{aligned} q(\boldsymbol{X}_*) &= \prod_n \mathcal{N}(\boldsymbol{x}_{n*} | \overline{\boldsymbol{x}}_n - \boldsymbol{b}, \boldsymbol{\Sigma}_{\boldsymbol{x}_n}) \,, \\ q(\boldsymbol{\mu}_*) &= \prod_m \mathcal{N}(\mu_{m*} | \overline{\mu}_m + \overline{\boldsymbol{w}}_m^{\mathrm{T}} \boldsymbol{b}, \widetilde{\mu}_m) \,. \end{aligned}$$

The lower bound of the loglikelihood to be maximised is

$$\left\langle \log p(\boldsymbol{Y} | \boldsymbol{W}, \boldsymbol{X}_*, \boldsymbol{\mu}_*, \tau) - \frac{\log q(\boldsymbol{X}_*)}{\log p(\boldsymbol{X}_*)} - \frac{\log q(\boldsymbol{\mu}_*)}{\log p(\boldsymbol{\mu}_*)} \right\rangle + \mathrm{const} \,,$$

where the expectation is taken over the transformed q distributions and const is used to represent terms that are constant with respect to the translation. If $\boldsymbol{\mu}$ has flat prior (i.e., $\beta \to 0$), the third term is constant. Then taking the derivative with respect to $\boldsymbol{b}$ and equating the result to zero yields

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

which implies that the expected mean of the latent variables $\boldsymbol{x}_n$ should be transformed to zero.

3.2 Rotation of $\boldsymbol{X}$ and $\boldsymbol{W}$

The loading matrix $\boldsymbol{W}$ can be rotated arbitrarily by compensating it in $\boldsymbol{X}$ as

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

Thus, we consider the following transformation:

$$\begin{aligned} q(\boldsymbol{W}_*) &= \prod_m \mathcal{N}(\boldsymbol{w}_{m*} | \boldsymbol{R}^{\mathrm{T}} \overline{\boldsymbol{w}}_m, \boldsymbol{R}^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{w}_m} \boldsymbol{R}) \,, \\ q(\boldsymbol{X}_*) &= \prod_n \mathcal{N}(\boldsymbol{x}_{n*} | \boldsymbol{R}^{-1} \overline{\boldsymbol{x}}_n, \boldsymbol{R}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{x}_n} \boldsymbol{R}^{-\,\mathrm{T}}) \,, \\ q(\boldsymbol{\alpha}_*) &= \prod_d \mathcal{G}(\alpha_{d*} | a_{\boldsymbol{\alpha}} + \frac{1}{2} M, b_{\boldsymbol{\alpha}} + \frac{1}{2} \boldsymbol{r}_d^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{r}_d) \,, \end{aligned}$$

where $\boldsymbol{r}_d$ is the d-th column of $\boldsymbol{R}$, $q(\boldsymbol{\alpha}_*)$ is motivated by the update rule of $q(\boldsymbol{\alpha})$ in [2]. The lower bound to be maximised as a function of $\boldsymbol{R}$ is

$$\left\langle \log p(\boldsymbol{Y}|\boldsymbol{W}_*, \boldsymbol{X}_*, \boldsymbol{\mu}_*, \tau) - \frac{\log q(\boldsymbol{X}_*)}{\log p(\boldsymbol{X}_*)} - \frac{\log q(\boldsymbol{W}_*)}{\log p(\boldsymbol{W}_*|\boldsymbol{\alpha}_*)} - \frac{\log q(\boldsymbol{\alpha}_*)}{\log p(\boldsymbol{\alpha}_*)} \right\rangle + \text{const} \tag{3}$$

where the expectation is taken over the transformed q distributions. The derivation of the result is shown in the appendix. To summarise, the rotation matrix is formed as $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, and $\boldsymbol{\Lambda}$ is a diagonal matrix. Matrices $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ can be found from the requirement

$$\boldsymbol{U}\boldsymbol{\Lambda}^2\boldsymbol{U}^{\mathrm{T}} = \frac{1}{N}\langle \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \rangle \quad \text{or equivalently} \quad \frac{1}{N}\langle \boldsymbol{X}_*\boldsymbol{X}_*^{\mathrm{T}} \rangle = \boldsymbol{I} \tag{4}$$

and matrix $\boldsymbol{V}$ should satisfy the following condition

$$\boldsymbol{V}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathrm{T}}\langle \boldsymbol{W}^{\mathrm{T}}\boldsymbol{W} \rangle \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V} = \langle \boldsymbol{W}_*^{\mathrm{T}}\boldsymbol{W}_* \rangle = \text{diagonal matrix.} \tag{5}$$

The matrices can be found using eigendecompositions. Thus, the transformation basically whitens the hidden states $\boldsymbol{x}_n$ and orthogonalises the columns of the loading matrix $\boldsymbol{W}$.

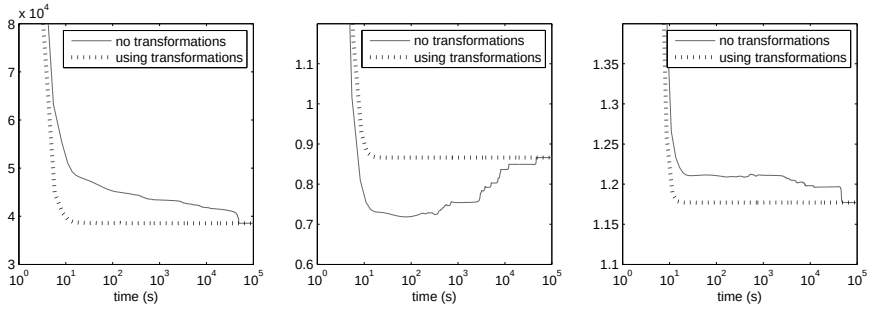
4 Experiments

In this section, we present a simple example which illustrates the significance of the transformations. We generate $N = 500$ data points with $M = 50$ dimensions having standard deviation of 5 along ten orthogonal directions and standard deviation of 1 along the remaining directions. The VB PCA model is fitted considering 20% of the data as missing, and using that part as a validation set. The model parameters are learned using $D = 30$ dimensions for the latent subspace. The Matlab toolbox for the experiments is available online at `http://www.cis.hut.fi/projects/bayes/`.

Fig. 1 shows the results for the models both with and without the presented transformations. The graphs show the VB cost (i.e., the negative of the log-likelihood lower bound) and root mean square error (RMSE) for the training and validation sets. Clearly, the transformations cause the iteration to converge much faster (notice the logarithmic scale). In both runs, the dimensionality of the latent space (5) was identified correctly but the transformations helped to determine the correct dimensionality much faster. This explains the overfitting effect in the run without transformation (see Fig. 1b,c).

5 Conclusions and discussion

In this paper, we showed how simple transformations of the latent space can speed up leaning of the variational Bayesian PCA model. The presented approach resembles the more general idea of using auxiliary parameters in VB



(a) VB cost (b) Training RMSE (c) Validation RMSE

Fig. 1: Experimental results obtained for the artificial data.

learning [6]. We gave theoretical justification and showed experimentally that the proposed transformations can significantly improve the rate of convergence. The transformations become extremely significant for large-scale datasets. More generally, we suggest that similar transformations can improve the algorithms for other variational Bayesian latent variable models. The exact formulae for other models can be derived using the presented methodology.

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

[2] C. M. Bishop. Variational principal components. In *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN'99)*, volume 1, pages 509–514, 1999.

[3] M. Collins, S. Dasgupta, and R. Schapire. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14 (NIPS)*, Cambridge, MA, 2002. MIT Press.

[4] H. Lappalainen and A. Honkela. Bayesian nonlinear independent component analysis by multi-layer perceptrons. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 93–121. Springer-Verlag, Berlin, 2000.

[5] A. Honkela and H. Valpola. Unsupervised variational Bayesian learning of nonlinear models. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 593–600. MIT Press, Cambridge, MA, USA, 2005.

[6] Y. Qi and T. S. Jaakkola. Parameter expanded variational Bayesian methods. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1097–1104. MIT Press, Cambridge, MA, 2007.

A Derivation of the rotation

We maximise the function (3) w.r.t. the rotation parameter $\boldsymbol{R}$ and therefore "const" in the following means a constant w.r.t. $\boldsymbol{R}$.

The first term is constant for the proposed rotation. Assuming flat prior for $\boldsymbol{\alpha}$ (i.e., $a_{\boldsymbol{\alpha}} \to 0$ and $b_{\boldsymbol{\alpha}} \to 0$), the fourth term is also constant, thus, the following terms remain:

$$\begin{aligned}
\langle \log p(\boldsymbol{X}_*) \rangle &= -\frac{1}{2} \operatorname{tr}(\boldsymbol{R}^{-1} \langle \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \rangle \boldsymbol{R}^{-\,\mathrm{T}}) + \text{const}\,, \\
-\langle \log q(\boldsymbol{X}_*) \rangle &= \frac{1}{2} \sum_n \log |\boldsymbol{R}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{x}_n} \boldsymbol{R}^{-\,\mathrm{T}}| + \text{const} = -N \log |\boldsymbol{R}| + \text{const}\,, \\
\langle \log p(\boldsymbol{W}_* | \boldsymbol{\alpha}_*) \rangle &= \frac{1}{2} M \sum_d \langle \log \alpha_{d*} \rangle - \frac{1}{2} \operatorname{tr}(\operatorname{diag} \langle \boldsymbol{\alpha}_* \rangle \boldsymbol{R}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{R}) + \text{const}\,, \\
-\langle \log q(\boldsymbol{W}_*) \rangle &= \frac{1}{2} \sum_m \log |\boldsymbol{R}^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{w}_m} \boldsymbol{R}| + \text{const} = M \log |\boldsymbol{R}| + \text{const}\,.
\end{aligned}$$

The flat prior yields $\langle \alpha_{d*} \rangle \approx M / (\boldsymbol{r}_d^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{r}_d)$ and therefore the second term in $\langle \log p(\boldsymbol{W}_* | \boldsymbol{\alpha}_*) \rangle$ is a constant:

$$-\frac{1}{2} \operatorname{tr}(\operatorname{diag} \langle \boldsymbol{\alpha}_* \rangle \boldsymbol{R}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{R}) \propto \sum_d \langle \alpha_{d*} \rangle \boldsymbol{r}_d^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{r}_d = \sum_d M = \text{const}\,.$$

We represent $\boldsymbol{R}$ using its singular value decomposition as $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{\Lambda}$ is diagonal. Matrix $\boldsymbol{V}$ affects only the term

$$\frac{M}{2} \sum_d \langle \log \alpha_{d*} \rangle \approx -\frac{M}{2} \log \prod_d \frac{1}{2} \boldsymbol{r}_d^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{r}_d + \text{const}\,.$$

Maximisation of this term w.r.t. $\boldsymbol{V}$ is equivalent to minimisation of the product of the diagonal elements of $\boldsymbol{V}^{\mathrm{T}} \boldsymbol{\Lambda} \boldsymbol{U}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}$. For a positive definite and symmetric matrix, the product of its diagonal elements is bounded below by the determinant[1] and it equals the determinant if the matrix is diagonal. Since an orthogonal rotation does not change the determinant, the optimal $\boldsymbol{V}$ is obtained when $\boldsymbol{V}^{\mathrm{T}} \boldsymbol{\Lambda} \boldsymbol{U}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{V}$ is diagonal, that is, $\boldsymbol{V}$ has the eigenvectors of $\boldsymbol{\Lambda} \boldsymbol{U}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{U} \boldsymbol{\Lambda}$ in its columns. This yields

$$\frac{M}{2} \sum_d \langle \log \alpha_{d*} \rangle \approx -\frac{M}{2} \log |\boldsymbol{R}^{\mathrm{T}} \langle \boldsymbol{W}^{\mathrm{T}} \boldsymbol{W} \rangle \boldsymbol{R}| = -M \log |\boldsymbol{R}| + \text{const}$$

and therefore the overall lower bound simplifies to

$$-\frac{1}{2} \operatorname{tr}\left((\boldsymbol{U}\boldsymbol{\Lambda})^{-1} \langle \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} \rangle (\boldsymbol{U}\boldsymbol{\Lambda})^{-\,\mathrm{T}} \right) - N \log |\boldsymbol{U}\boldsymbol{\Lambda}| + \text{const}\,.$$

The result (4) is obtained by equating the derivative w.r.t. $\boldsymbol{U}\boldsymbol{\Lambda}$ to zero.

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