Algorithmic improvements for variational inference

Antti Honkela, Tapani Raiko, Alexander Ilin, Mikael Kuusela, Matti Tornio, Jaakko Luttinen, and Juha Karhunen

Riemannian conjugate gradient

Variational methods for approximate inference in machine learning often adapt a parametric probability distribution to optimize a given objective function. This view is especially useful when applying variational Bayes (VB) to models outside the conjugate-exponential family. For them, variational Bayesian expectation maximization (VB EM) algorithms are not easily available, and gradient-based methods are often used as alternatives.

In previous machine learning algorithms based on natural gradients [1], the aim has been to use maximum likelihood to directly update the model parameters $\boldsymbol{\theta}$ taking into account the geometry imposed by the predictive distribution for data $p(\boldsymbol{X}|\boldsymbol{\theta})$. The resulting geometry is often very complicated as the effects of different parameters cannot be separated and the Fisher information matrix is relatively dense.

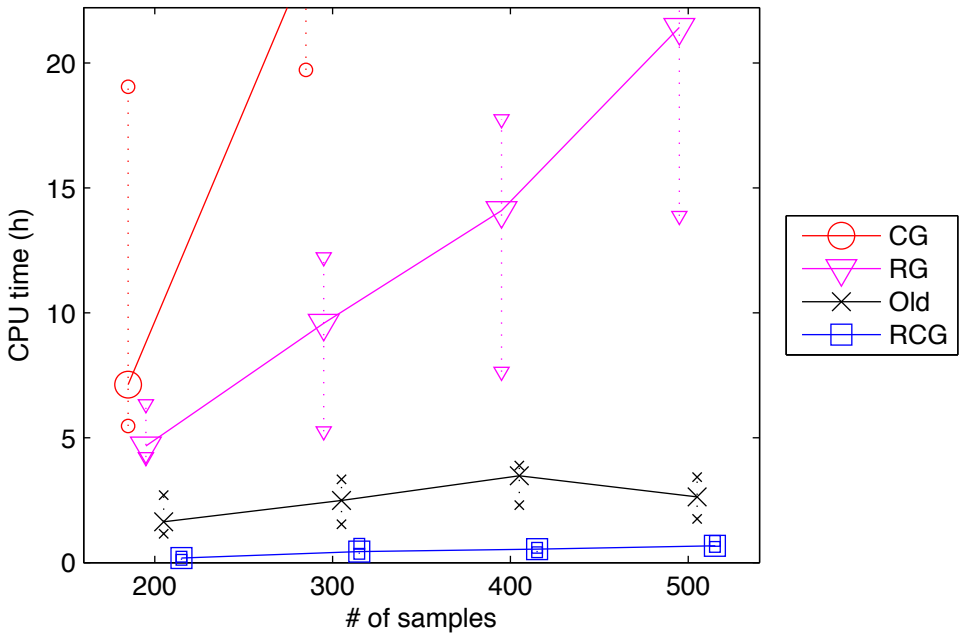


Figure 1: Convergence speed of the Riemannian conjugate gradient (RCG), the Riemannian gradient (RG) and the conjugate gradient (CG) methods as well as the heuristic algorithm (Old) with different data sizes. The lines show median times with 25 % and 75 % quantiles shown by the smaller marks. The times were limited to at most 24 hours, which was reached by a number of simulations.

Recently, in [2], we propose using natural gradients for free energy minimisation in variational Bayesian learning using the simpler geometry of the approximating distributions $q(\boldsymbol{\theta}|\boldsymbol{\xi})$. Because the approximations are often chosen to minimize dependencies between different parameters $\boldsymbol{\theta}$, the resulting Fisher information matrix with respect to the variational parameters $\boldsymbol{\xi}$ will be

mostly diagonal and hence easy to invert.

While taking into account the structure of the approximation, plain natural gradient in this case ignores the structure of the model and the global geometry of the parameters $\boldsymbol{\theta}$. This can be addressed by using conjugate gradients. Combining the natural gradient search direction with a conjugate gradient method yields our proposed *approximate Riemannian conjugate gradient (RCG)* method.

The RCG algorithm was compared against conjugate gradient (CG) and Riemannian gradient (RG) algorithms in learning a nonlinear state-space model [3]. The results for a number of datasets ranging from 200 to 500 samples of 21 dimensional speech spectrograms can be seen in Figure 1. The plain CG and RG methods were clearly slower than others and the maximum runtime of 24 hours was reached by most CG and some RG runs. RCG was clearly the fastest algorithm with the older heuristic method of [3] between these extremes. The results with a larger data set are very similar with RCG outperforming all alternatives by a factor of more than 10.

The experiments in [2] show that the natural conjugate gradient method outperforms both conjugate gradient and natural gradient methods by a large margin. Considering univariate Gaussian distributions, the regular gradient is too strong for model variables with small posterior variance and too weak for variables with large posterior variance. The posterior variance of latent variables is often much larger than the posterior variance of model parameters and the natural gradient takes this into account in a very natural manner.

Transformation of latent variables

Variational methods have been used for learning linear latent variable models in which observed data vectors $\mathbf{x}(t)$ are modeled as linear combination of latent variables $\mathbf{s}(t)$:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\mu} + \mathbf{n}(t), \quad t = 1, \ldots, N. \tag{1}$$

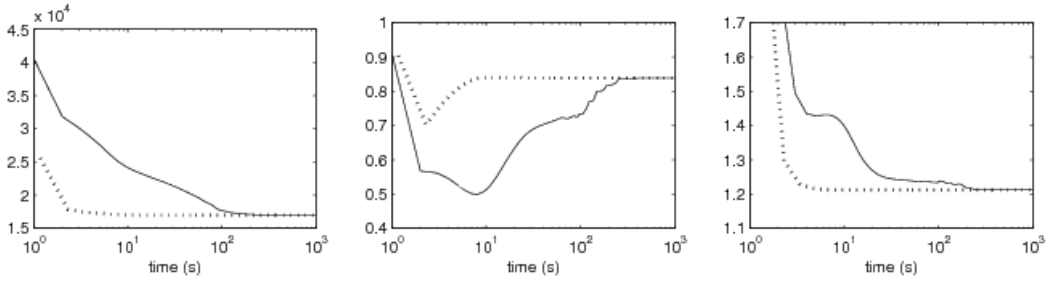
The latent variables are assigned some prior distributions, such as zero-mean Gaussian priors with uncorrelated components in the basic factor analysis model. When VB learning is used, the true posterior probability density function (pdf) of the unknown variables is approximated using a tractable pdf factorized as follows:

$$p\left(\boldsymbol{\mu}, \mathbf{A}, \mathbf{s}(1), ..., \mathbf{s}(N) \mid \{\mathbf{x}(t)\}\right) \approx q(\boldsymbol{\mu})q(\mathbf{A})q(\mathbf{s}(1)) \ldots q(\mathbf{s}(N)) \,.$$

This form of the posterior approximation q ignores the strong correlations present between the variables, which often causes slow convergence of VB learning.

Parameter-expanded VB (PX-VB) methods were recently proposed to address the slow convergence problem [4]. 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 and faster convergence is facilitated. One of the main challenges for applying the PX-VB methodology is to use proper reparameterization of the original model.

In our journal paper [5], we present a similar idea in the context of VB learning of factor analysis models. There we use auxiliary parameters $\mathbf{b}$ and $\mathbf{R}$ which translate and rotate the latent



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

Figure 2: Convergence of VB PCA tested on artificial data. The dotted and solid curves represent the results with and without the proposed transformations, respectively.

variables:

$$\begin{aligned} \mathbf{s}(t) &\leftarrow \mathbf{s}(t) - \mathbf{b} & \boldsymbol{\mu} &\leftarrow \boldsymbol{\mu} + \mathbf{A}\mathbf{b} \\ \mathbf{s}(t) &\leftarrow \mathbf{R}\mathbf{s}(t) & \mathbf{A} &\leftarrow \mathbf{A}\mathbf{R}^{-1}\,. \end{aligned}$$

The optimal parameters $\mathbf{b}$ and $\mathbf{R}$ which minimize the misfit between the posterior pdf and its approximation can then be computed analytically. This corresponds to joint optimization of factors $q(\mathbf{s}(t))$. In our paper, we show that the proposed transformations essentially perform centering and whitening of the hidden factors taking into account their posterior uncertainties.

We tested the effect of the proposed transformations by applying the VB PCA model to an artificial dataset consisting of $N = 200$ samples of normally distributed 50-dimensional vectors $\mathbf{x}(t)$. Figure 2 shows the minimized VB cost and the root mean squared error (RMSE) computed on the training and test sets during learning. The curves indicate that the method first overfits providing a solution with an unreasonably small RMSE. Later, learning proceeds toward a better solution yielding smaller test RMSE. Note that using the proposed transformations reduced the overfitting effect at the beginning of learning, which led to faster convergence to the optimal solution.

References

[1] S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[2] A. Honkela, T. Raiko, M. Kuusela, M. Tornio, and J. Karhunen. Approximate Riemannian Conjugate Gradient Learning for Fixed-Form Variational Bayes. In Journal of Machine Learning Research (JMLR), volume 11, pages 3235–3268, November 2010.

[3] H. Valpola and J. Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural Computation*, 14(11):2647–2692, 2002.

[4] Y. Qi, T. S. Jaakkola. Parameter expanded variational Bayesian methods. In *Advances in Neural Information Processing Systems 19*, pp. 1097–1104, Cambridge, MA, 2007.

[5] J. Luttinen and A. Ilin. Transformations in variational Bayesian factor analysis to speed up learning. *Neurocomputing*, 73(7-9):1093–1102, 2010.