# Chapter 2

# Bayesian learning of latent variable models

**Juha Karhunen, Erkki Oja, Tapani Raiko, Alexander Ilin, Antti Honkela, Jaakko Luttinen, KyungHyun Cho**

## 2.1 Bayesian modeling and variational learning

Unsupervised learning methods are often based on a generative approach where the goal is to find a latent variable model which explains how the observations were generated. It is assumed that there exist certain latent variables (also called in different contexts source signals, factors, or hidden variables) which have generated the observed data through an unknown mapping. The goal of generative learning is to identify both the latent variables and the unknown generative mapping.

The success of a specific model depends on how well it captures the structure of the phenomena underlying the observations. Various linear models have been popular, because their mathematical treatment is fairly easy. However, in many realistic cases the observations have been generated by a nonlinear process. Unsupervised learning of a nonlinear model is a challenging task, because it is typically computationally much more demanding than for linear models, and flexible models require strong regularization for avoiding overfitting.

In Bayesian data analysis and estimation methods, all the uncertain quantities are modeled in terms of their joint probability distribution. The key principle is to construct the joint posterior distribution for all the unknown quantities in a model, given the data sample. This posterior distribution contains all the relevant information on the parameters to be estimated in parametric models, or the predictions in non-parametric prediction or classification tasks [1, 2].

Denote by $\mathcal{H}$ the particular model under consideration, and by $\boldsymbol{\theta}$ the set of model parameters that we wish to infer from a given data set $X$. The posterior probability density $p(\boldsymbol{\theta}|X, \mathcal{H})$ of the parameters given the data $X$ and the model $\mathcal{H}$ can be computed from the Bayes' rule

$$p(\boldsymbol{\theta}|X, \mathcal{H}) = \frac{p(X|\boldsymbol{\theta}, \mathcal{H})p(\boldsymbol{\theta}|\mathcal{H})}{p(X|\mathcal{H})} \tag{2.1}$$

Here $p(X|\boldsymbol{\theta}, \mathcal{H})$ is the likelihood of the parameters $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}|\mathcal{H})$ is the prior pdf of the parameters, and $p(X|\mathcal{H})$ is a normalizing constant. The term $\mathcal{H}$ denotes all the assumptions made in defining the model, such as the choice of a particular model class and structure, specific noise model, etc.

The parameters $\boldsymbol{\theta}$ of a particular model $\mathcal{H}_i$ are often estimated by seeking the peak value of a probability distribution. The non-Bayesian maximum likelihood (ML) method uses to this end the distribution $p(X|\boldsymbol{\theta}, \mathcal{H})$ of the data, and the Bayesian maximum a posteriori (MAP) method finds the parameter values that maximize the posterior probability density $p(\boldsymbol{\theta}|X, \mathcal{H})$. However, using point estimates provided by the ML or MAP methods is often problematic, because the model order estimation and overfitting (choosing too complicated a model for the given data) are severe problems [1, 2].

Instead of searching for some point estimates, the correct Bayesian procedure is to use all possible models to evaluate predictions and weight them by the respective posterior probabilities of the models. This means that the predictions will be sensitive to regions where the probability mass is large instead of being sensitive to high values of the probability density [3, 2]. This procedure optimally solves the issues related to the model complexity and choice of a specific model $\mathcal{H}_i$ among several candidates. In practice, however, the differences between the probabilities of candidate model structures are often very large, and hence it is sufficient to select the most probable model and use the estimates or predictions

given by it.

A problem with fully Bayesian estimation is that the posterior distribution (2.1) has a highly complicated form except for in the simplest problems. Therefore it is too difficult to handle exactly, and some approximative method must be used. Variational methods form a class of approximations where the exact posterior is approximated with a simpler distribution [4, 2]. In a method commonly known as *Variational Bayes (VB)* [1, 3, 2] the misfit of the approximation is measured by the Kullback-Leibler (KL) divergence between two probability distributions $q(v)$ and $p(v)$. The KL divergence is defined by

$$D(q \parallel p) = \int q(v) \ln \frac{q(v)}{p(v)} dv \tag{2.2}$$

which measures the difference in the probability mass between the densities $q(v)$ and $p(v)$.

A key idea in the VB method is to minimize the misfit between the actual posterior pdf and its parametric approximation using the KL divergence. The approximating density is often taken a diagonal multivariate Gaussian density, because the computations become then tractable. Even this crude approximation is adequate for finding the region where the mass of the actual posterior density is concentrated. The mean values of the Gaussian approximation provide reasonably good point estimates of the unknown parameters, and the respective variances measure the reliability of these estimates.

A main motivation of using VB is that it avoids overfitting which would be a difficult problem if ML or MAP estimates were used. VB method allows one to select a model having appropriate complexity, making often possible to infer the correct number of latent variables or sources. It has provided good estimation results in the very difficult unsupervised (blind) learning problems that we have considered.

Variational Bayes is closely related to information theoretic approaches which minimize the description length of the data, because the description length is defined to be the negative logarithm of the probability. Minimal description length thus means maximal probability. In the probabilistic framework, we try to find the latent variables or sources and the nonlinear mapping which most probably correspond to the observed data. In the information theoretic framework, this corresponds to finding the latent variables or sources and the mapping that can generate the observed data and have the minimum total complexity. This information theoretic view also provides insights to many aspects of learning and helps to explain several common problems [5].

During the last two years, our research has extended to deep learning, which is not a Bayesian but a probabilistic latent variable analysis method. In deep learning one tries to find hierarchical representations of data, starting from observations towards more and more abstract representations. Deep learning can be cumbersome and difficult but on the other hand it can provide world record results in difficult classification problems. We have improved deep learning algorithms, making them more stable and robust against the choice of learning parameters. Deep learning is discussed in this chapter in its own subsection.

In the following subsections, we first discuss improvements in variational Bayesian learning, including a natural conjugate gradient algorithm which speeds up learning remarkably, as well as transformations of latent variables leading also to a faster convergence. After this we consider extensions of probabilistic principal component analysis (PCA) for treating missing values and achieving robustness in the presence of outliers. We then consider

time series modeling in bioinformatics to learn gene regulatory relationships from time series expression data. Our contributions to deep learning and Boltzmann machines are discussed in the next section. Finally, we have carried out some work on oscillatory neural networks, and applied our Bayesian methods to novelty detection in structural health monitoring and document classification utilizing relational information.

## 2.2 Algorithmic improvements for variational inference

### 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 [6], 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.

Recently, in [7], 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 [8]. The results for a number of datasets ranging from 200 to 500 samples of 21 dimensional speech spectrograms can be seen in Figure 2.1. 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 [8] 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 [7] 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 $\mathbf{s}(t)$:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\mu} + \mathbf{n}(t), \quad t = 1, \ldots, N. \tag{2.3}$$

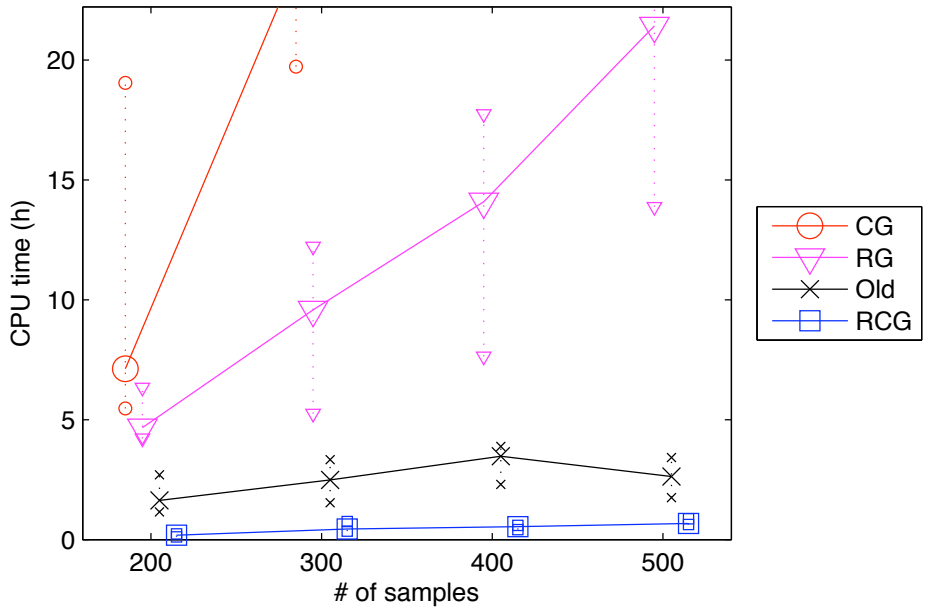



Figure 2.1: Convergence speed of the Riemannian conjugate gradient (RCG), the Riemannian gradient (RG) and the conjugate gradient (CG) methods as well as the heuristic algorithm (Old) with different data sizes. The lines show median times with 25 % and 75 % quantiles shown by the smaller marks. The times were limited to at most 24 hours, which was reached by a number of simulations.

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:

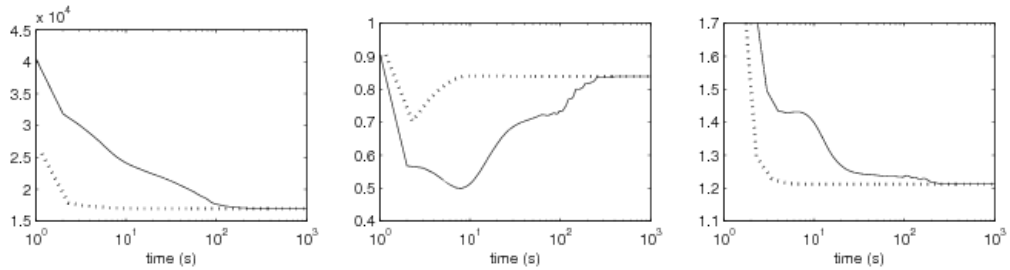
$$p\left(\boldsymbol{\mu}, \mathbf{A}, \mathbf{s}(1), ..., \mathbf{s}(N) \mid \{\mathbf{x}(t)\}\right) \approx q(\boldsymbol{\mu})q(\mathbf{A})q(\mathbf{s}(1)) \ldots q(\mathbf{s}(N)) \,.$$

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 [9]. 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 [10], 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 variables:

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




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

Figure 2.2: Convergence of VB PCA tested on artificial data. The dotted and solid curves represent the results with and without the proposed transformations, respectively.

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 2.2 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.

## 2.3 Extensions of probabilistic PCA

### PCA of large-scale datasets with many missing values

Principal component analysis (PCA) is a classical data analysis technique. Some algorithms for PCA scale better than others to problems with high dimensionality. They also differ in the ability to handle missing values in the data. In our recent paper [11], a case is studied where the data are high-dimensional and a majority of the values are missing. In the case of very sparse data, overfitting becomes a severe problem even in simple linear models such as PCA. Regularization can be provided using the Bayesian approach by introducing prior for the model parameters. The PCA model can then be identified using, for example, maximum a posteriori estimates (MAPPCA) or variational Bayesian (VBPCA) learning.

In [11], we study different approaches to PCA for incomplete data. We show that faster convergence can be achieved using the following rule for the model parameters:

$$\theta_i \leftarrow \theta_i - \gamma \left( \frac{\partial^2 C}{\partial \theta_i^2} \right)^{-\alpha} \frac{\partial C}{\partial \theta_i} ,$$

where $\alpha$ is a control parameter that allows the learning algorithm to vary from the standard gradient descent ($\alpha = 0$) to the diagonal Newton's method ($\alpha = 1$). These learning rules can be used for standard PCA learning and extended to MAPPCA and VBPCA.

The algorithms were tested on the Netflix problem (`http://www.netflixprize.com/`), which is a task of predicting preferences (or producing personal recommendations) by using other people's preferences. The Netflix problem consists of movie ratings given by 480189 customers to 17770 movies. There are 100480507 ratings from 1 to 5 given, and the task is to predict 2817131 other ratings among the same group of customers and movies. 1408395 of the ratings are reserved for validation. Thus, 98.8% of the values are missing.

We used different variants of PCA in order to predict the test ratings in the Netflix data set. The obtained results are shown in Figure 2.3. The best accuracy was obtained using VB PCA with a simplified form of the posterior approximation (VBPCAd in Figure 2.3). That method was also able to provide reasonable estimates of the uncertainties of the predictions.

### Robust PCA for incomplete data

Standard PCA is known to be sensitive to outliers in the data because it is based on minimisation of a quadratic criterion such as the mean-square representation error. Thus, corrupted or atypical observations may cause the failure of PCA, especially for data sets with missing values. A standard way to cope with this problem is replacing the quadratic cost function of PCA a function which grows more slowly.

In [12], we present a new robust PCA model based on the Student-$t$ distribution and show how it can be identified for data sets with missing values. We make the assumption that the outliers can arise independently in each sensor (i.e. for each dimension of a data vector). This assumption is different to the previously introduced techniques [13] and it turns out to be important for modeling incomplete data sets. The proposed model can improve the quality of the principal subspace estimation and provide better reconstructions of missing

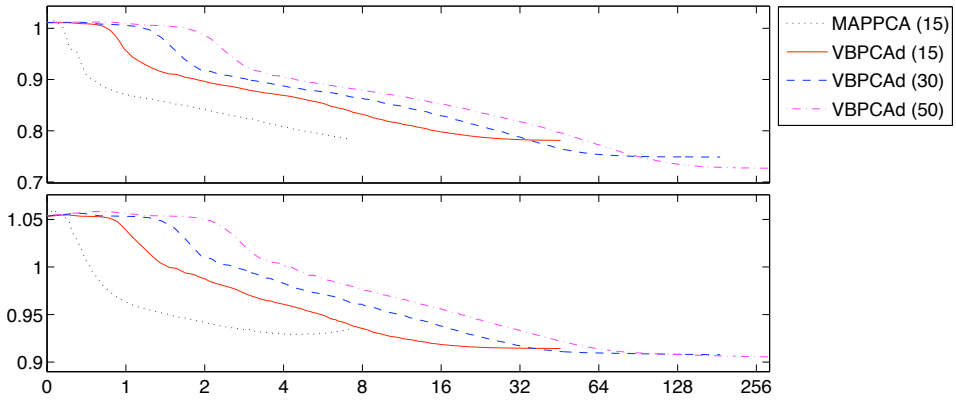



Figure 2.3: Root mean squared errors for the Netflix data (y-axis) plotted against the processor time in hours. The upper plot shows the training error while the lower plot shows the error for the probing data provided by Netflix. The time scale is linear from 0 to 1 and logarithmic above 1.

values. The model can also be used to remove outliers by estimating the true values of their corrupted components from the uncorrupted ones.

We tested the robust PCA model on the Helsinki Testbed data set which at the moment of our studies contained many atypical measurements and missing values. The model was used to estimate four principal components of the temperature measurements from 79 stations in Southern Finland. Figure 2.4 presents the reconstruction of the data using our robust PCA model for four different stations. The reconstructions look very reasonable with most of the outliers being removed.

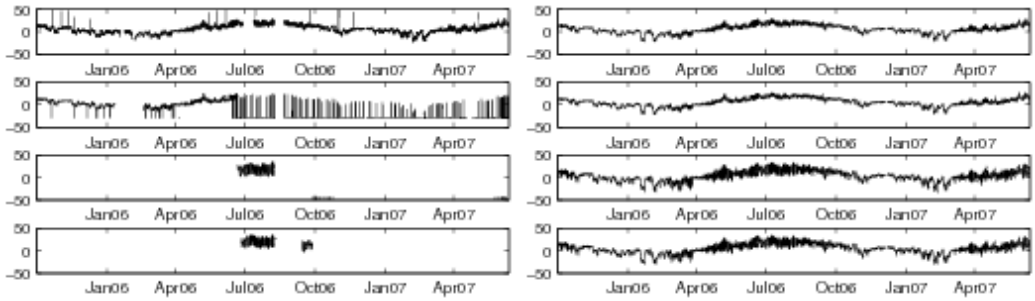



Figure 2.4: Four example signals from the Helsinki Testbed dataset and their reconstructions using the proposed robust PCA.

The results of this study are presented in more detail in the journal manuscript [14]. More traditional methods for robust PCA, also with missing values, have been studied in [15]. They are usually much easier to apply compared with Bayesian methods but less effective.

## 2.4 Gaussian process models of gene expression and gene regulation

Bayesian methods are well-suited for analysis of molecular biology data as the data sets practically always consist of very few samples with a high noise level. We have studied models of gene transcription regulation based on time series gene expression data in collaboration with Neil D. Lawrence and Magnus Rattray of the University of Sheffield. This is a very challenging modelling task as the time series are very short, typically at most a dozen time points.

Extending the model of [16] of single input motif systems, i.e. where a single transcription factor regulates a number of genes, we have developed a method of ranking putative targets of transcription factors based on expression data [17]. This is achieved by imposing a Gaussian process (GP) prior on the latent continuous time transcription factor gene expression profile, which drives a linear ODE model of transcription factor protein translation and target gene transcription. This linear ODE model leads to a joint GP model for all observable gene expression values and allows exact marginalisation of the latent functions. Candidate target genes can be ranked using model likelihood.

We have applied the model to genome-wide ranking of potential target genes of transcription factors. Fig. 2.5 shows results from experiments with key regulators of *Drosophila* mesoderm and muscle development. They show very high accuracy in terms of enrichment of detected transcription factor binding near the predicted target genes [17]. An implementation of the method is available in Bioconductor for R [18].

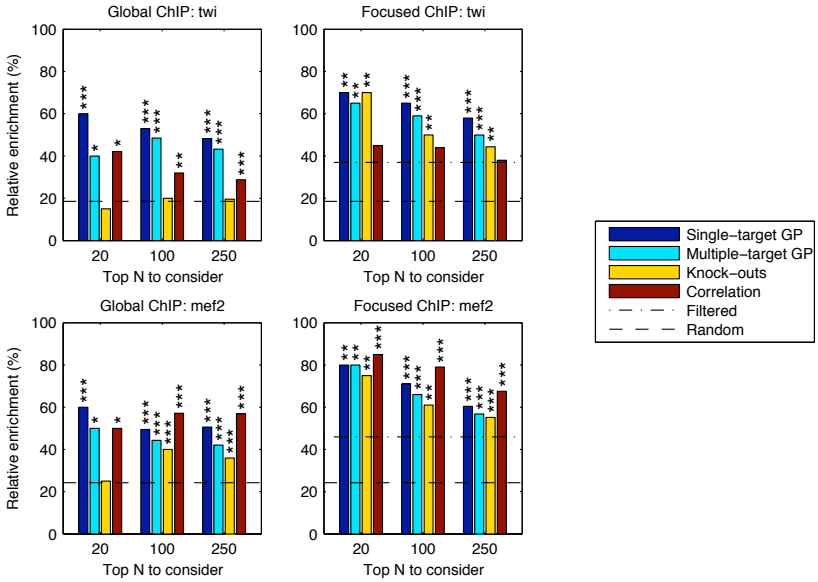



Figure 2.5: Evaluation results from [17] of two variants of the proposed GP-based ranking methods and two alternatives showing the relative frequency of positive predictions among $N$ top-ranking targets ("global" evaluations) and among $N$ top genes with annotated expression in mesoderm or muscle tissue ("focused" evaluations) for two studied transcription factors. The dashed line denotes the frequency in the full population and the dash-dot line within the population considered in focused evaluation. The bars show the frequency of targets with ChIP-chip binding within 2000 base pairs of the gene. $p$-values of results significantly different from random are denoted by '***': $p < 0.001$, '**': $p < 0.01$, '*': $p < 0.05$.

## 2.5 Deep learning and Boltzmann machines

Deep learning has gained its popularity recently as a way of learning complex and large probabilistic models [25]. Especially, deep neural networks such as a deep belief network and a deep Boltzmann machine have been applied to various machine learning tasks with impressive improvements over conventional approaches.

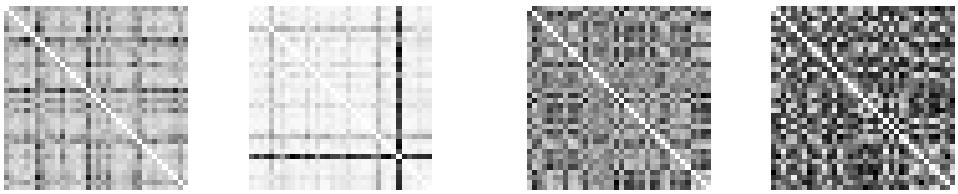
Deep neural networks are characterized by the large number of layers of neurons and by using layer-wise unsupervised pretraining to learn a probabilistic model for the data. A deep neural network is typically constructed by stacking multiple restricted Boltzmann machines (RBM) so that the hidden layer of one RBM becomes the visible layer of another RBM. Layer-wise pretraining of RBMs then facilitates finding a more accurate model for the data. Various papers (see, e.g., [26], [25] and references therein) empirically confirmed that such multi-stage learning works better than conventional learning methods.

Unfortunately, even training a simple RBM which consists of only two layers of visible and hidden neurons is known to be difficult [31, 32]. This problem is often evidenced by the decreasing likelihood during learning. These failures have discouraged using RBMs and its extensions such as deep Boltzmann machines for more sophisticated and variety of machine learning tasks.

In our recent conference papers [28], we have proposed to use parallel tempering, an ad-

vanced Markov-chain Monte-Carlo sampling, as a replacement of a simple Gibbs sampling in obtaining samples from a model distribution defined by an RBM. It was shown that a better model with higher log-likelihood could be found using the stochastic gradient method based on PT compared to a widely-used method of minimizing contrastive divergence.

Additionally to the problem of using a simple Gibbs sampling we have determined other possible problems that discourage using an RBM as a building block for building a deep neural network. In [29] we identified density of training samples and learning hyper-parameters, such as a learning rate and an initialization of parameters, as two sources of difficulty in training RBMs. Furthermore, we also discovered that the conventional form of an energy function of Gaussian-Bernoulli RBM (GRBM) is defected in some sense that learning becomes easily unstable, in [27].



Traditional Gradient Enhanced Gradient

Figure 2.6: The angles between the update directions for the weights of an RBM with 36 hidden neurons. White pixels correspond to small angles, while black pixels correspond to orthogonal directions. From left to right: traditional gradient after 26 updates, traditional gradient after 364 updates, enhanced gradient after 26 updates, and enhanced gradient after 364 updates.

We have derived a new update direction for training RBMs, called enhanced gradient, in [29]:

$$w_{ij} \leftarrow w_{ij} + \eta_w \nabla_e \, w_{ij} \tag{2.4}$$

$$b_i \leftarrow b_i + \eta_b \nabla_e \, b_i \tag{2.5}$$

$$c_j \leftarrow c_j + \eta_c \nabla_e \, c_j, \tag{2.6}$$

where $w_{ij}$, $b_i$ and $c_j$ are weight between a visible neuron $i$ and a hidden neuron $j$ and biases for a visible neurons $i$ and a hidden neuron $j$, respectively.

The enhanced gradient makes learning based on the stochastic gradient invariant to the density of training samples as well as the sparsity of hidden neurons. It turned out that the enhanced gradient is more robust to the choice of learning hyper-parameters and makes the gradient per hidden neuron more orthogonal to each other as can be see in Figure 2.6. It was shown to help avoid a common degenerate case where most hidden neurons learn a bias.

Also in [29], we proposed a new adaptation mechanism, call adaptive learning rate, for choosing a learning rate on-the-fly. The adaptive learning rate greedily adapts the learning rate while learning parameters by maximizing the locally estimated log-likelihood. Together with the enhanced gradient, it shows in Figure 2.7 that more stable and better models can be trained.

All three approaches– parallel tempering, the enhanced gradient, and the adpative learning rate– have been shown to work with extensions of RBMs. In [27], we showed that these

methods can be directly applied to a GRBM which replaces a binary visible neuron of an RBM with a Gaussian neuron. Furthermore, we showed that a hierarchical version of Boltzmann machines called deep Boltzmann machines (DBM) can readily use the proposed approaches in [30].

Additionally to studying Boltzmann machines for deep learning, a method of transforming a standard multi-layer perceptron by introducing linear shortcut connections and proposing transformations in non-linearities was proposed in [33]. It was shown in the paper that with the proposed transformations a faster convergence to a state-of-the-art performance can be achieved.

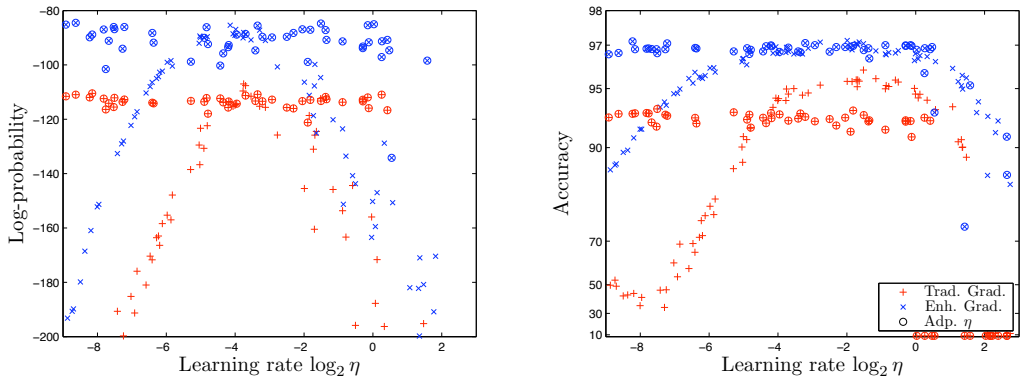



Figure 2.7: Log-probabilities and classificiation accuracies of test data for different initializations of the learning rate. The models were trained on MNIST using the stochastic gradient with parallel tempering.

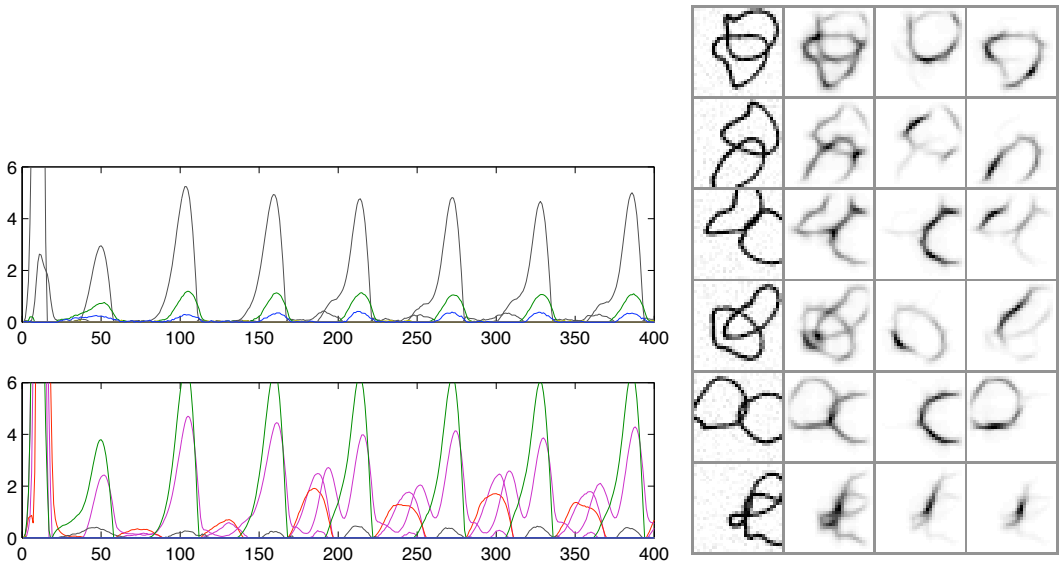



Figure 2.8: Left: The neural signals corresponding to the middle patch (top) and the patch below it (bottom) plotted as a function of time. The activities are given for the top-most data sample. Right: The segmentation result obtained from NMF analysis of the signals. First column from the left is the data, second column is the reconstruction from the feature activities, third and fourth columns are segmented objects.

## 2.6 Oscillatory neural networks

In [34] we studied the emergent properties of an artificial neural network which combines segmentation by oscillations and biased competition for perceptual processing. The aim was to progress in image segmentation by mimicking abstractly the way how the cerebral cortex works. In our model, the neurons associated with features belonging to an object start to oscillate synchronously, while competing objects oscillate with an opposing phase.

The overall structure of our network is such that there are so called areas that correspond to patches in the image. The areas get bottom-up input from the pixels. The areas should be connected to each other with local interactions only, that is, there is no hierarchy or global signals. The different areas should work in the same way, using the same algorithms. The emergent properties of the network are confirmed by experiments with artificial image data as seen in Figure 2.8.

## 2.7 Applications in climate science

We applied the Bayesian methodology for several problems in climate science.

In our papers [19, 21], we consider the problem of historical reconstruction of climate fields, which is a problem of infilling missing values in the observational data. We take the statistical approach and propose a probabilistic model called Gaussian-process factor analysis (GPFA). The model is based on standard matrix factorization

$$\mathbf{Y} = \mathbf{W}\mathbf{X} + \text{noise} = \sum_{d=1}^{D} \mathbf{w}_{:d}\mathbf{x}_{d:}^{\mathrm{T}} + \text{noise},$$

where $\mathbf{Y}$ is a data matrix in which each row contains measurements in one spatial location and each column corresponds to one time instance. The goal is to learn the model parameters $\mathbf{W}$, $\mathbf{X}$ from available observations in order to reconstruct the missing values in $\mathbf{Y}$. Each $\mathbf{x}_{d:}$ is a vector representing the time series of one of the $D$ factors whereas $\mathbf{w}_{:d}$ is a vector of loadings which are spatially distributed. We assume that both factors $\mathbf{x}_{d:}$ and corresponding loadings $\mathbf{w}_{:d}$ have prominent structures that we model using the Gaussian process methodology [20]. The model is identified in the framework of variational Bayesian learning and high computational cost of GP modeling is reduced by using sparse approximations derived in the variational methodology.

Another problem studied in our group is parametric tuning of climate models. Climate models contain closure parameters which can act as effective "tuning handles" of the simulated climate. These appear in physical parameterization schemes where unresolved variables are expressed by predefined parameters rather than being explicitly modeled. In the current climate model tuning process, best expert knowledge is used to define the optimal closure parameter values, based on observations, process studies, large eddy simulations, etc.

Our research group participates in the Academy of Finland project called "Novel advanced mathematical and statistical methods for understanding climate" (NOVAC, 2010-2013), whose goal is to develop algorithmic ways for closure parameter estimation. We focus on the atmospheric model ECHAM5 but the methodology is generic and applicable in any multi-scale problem with similar closure parameters [22].

The uncertainties of the closure parameters are estimated using Markov chain Monte Carlo (MCMC) simulations [23]. The MCMC approach is, however, computationally very expensive and only maximally optimized MCMC techniques make the approach realistic in practice. We develop new tools based on adaptive algorithms, multiple computational grids, parallel chains as well as methods based on early rejection.

The central problem in closure parameter estimation is how to formulate the likelihood function. This task is not trivial because of the chaotic nature of climate models. Climate model simulations quickly diverge from observations, which makes classical parameter estimation based on direct comparison of model simulations and observations inefficient. Our initial approach to circumvent the chaoticity problem was to formulate the likelihood function in terms of summary statistics. In [23], the likelihood is evaluated by comparing some temporal and spatial averages of observed and simulated data. Several summary statistics potentially useful for climate model tuning have been studied in [24].

## 2.8 Other Applications

We applied nonlinear factor analysis to novelty detection for structural health monitoring in [35]. In vibration-based structural health monitoring damage in structure is tried to detect from damage-sensitive features. Because neither prior information nor data about expected damage are normally available, damage detection problem must be solved by using a novelty detection approach. Features, which are sensitive to damage, are often sensitive to environmental and operational variations. Therefore elimination of these variations is essential for reliable damage detection. At present many of the damage detection methods are linear, though it has been shown that many of the vibration changes in structures are bilinear or nonlinear. We proposed to use nonlinear factor analysis to detect damage via elimination of external effectsfrom damage features. The effectiveness of the proposed method was demonstrated by analyzing the experimental Z24 Bridge data with a comparison to a linear method [35]. It was shown that elimination of adverse effects and damage detection are feasible.

In [36], we studied document classification utilising relational information. Two major types of relational information can be utilized in automatic document classification as background information: relations between terms, such as ontologies, and relations between documents, such as web links or citations in articles. We introduced a model where a traditional bag-of-words type classifier is gradually extended to utilize both of these information types. The experiments with data from the Finnish National Archive show that classification accuracy improves from 70% to 74% when the General Finnish Ontology YSO is used as background information, without using relations between documents.

## References

[1] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[2] C. Bishop, *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[3] H. Lappalainen and J. Miskin. Ensemble learning. In M. Girolami, editor, *Advances in Independent Component Analysis*, Springer, 2000, pages 75–92.

[4] M. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. An introduction to variational methods for graphical models. In M. Jordan, editor, *Learning in Graphical Models*, MIT Press, 1999, pages 105–161.

[5] A. Honkela and H. Valpola. Variational learning and bits-back coding: an information-theoretic view to Bayesian learning. *IEEE Transactions on Neural Networks*, 15(4):267–282, 2004.

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

[7] 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.

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

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

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

[11] A. Ilin and T. Raiko. Practical approaches to principal component analysis in the presence of missing values. In Journal of Machine Learning Research (JMLR), volume 11, pages 1957–2000, July, 2010.

[12] J. Luttinen, A. Ilin, and Juha Karhunen. Bayesian robust PCA for incomplete data. In *Proc. of the 8th International Conference on Independent Component Analysis and Blind Signal Separation (ICA 2009)*, pp. 66–73, Paraty, Brazil, March 2009.

[13] C. Archambeau, N. Delannay, M. Verleysen. Robust probabilistic projections. In *Proc. of the 23rd International Conference on Machine Learning (ICML 2006)*, pp. 33-40, New York, NY, USA, 2006.

[14] J. Luttinen, A. Ilin, and Juha Karhunen. Bayesian robust PCA for incomplete data. Revised version submitted to *Neural Processing Letters*, 2012.

[15] J. Karhunen. Robust PCA methods for complete and missing data. *Neural Network World*, 21(5):357–392, 2011.

[16] P. Gao, A. Honkela, M. Rattray, and N. D. Lawrence. Gaussian process modelling of latent chemical species: applications to inferring transcription factor activities. *Bioinformatics* 24(16):i70–i75, 2008.

[17] A. Honkela, C. Girardot, E. H. Gustafson, Y.-H. Liu, E. E. M. Furlong, N. D. Lawrence, and M. Rattray. Model-based method for transcription factor target identification with limited data. *Proc Natl Acad Sci U S A* 107(17):7793–7798, 2010.

[18] A. Honkela, P. Gao, J. Ropponen, M. Rattray, and N. D. Lawrence. tigre: Transcription factor Inference through Gaussian process Reconstruction of Expression for Bioconductor. *Bioinformatics* 27(7):1026–1027, 2011.

[19] J. Luttinen and A. Ilin. Variational Gaussian-process factor analysis for modeling spatio-temporal data. In *Advances in Neural Information Processing Systems 22*, 2009.

[20] C. E. Rasmussen, C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[21] A. Ilin and J. Luttinen. Variational Gaussian-process factor analysis for modeling spatio-temporal data. In *Proc. of the 11th Meeting on Statistical Climatology*, Edinburgh, Scotland, 2010.

[22] H. Haario, E. Oja, A. Ilin, H. Järvinen and J. Tamminen Novel advanced mathematical and statistical methods for understanding climate (NOVAC). In *Proc. of the 11th Meeting on Statistical Climatology*, Edinburgh, Scotland, 2010.

[23] H. Järvinen, P. Räisänen, M. Laine, J. Tamminen, A. Ilin, E. Oja, A. Solonen, and H. Haario. Estimation of ECHAM5 climate model closure parameters with adaptive MCMC. *Atmospheric Chemistry and Physics Discussion*, Vol. 10, No 5, pp. 11951–11973, 2010.

[24] J. Saarimäki. Performance Metrics for the Atmospheric Model ECHAM5. Master's thesis, Aalto University, 2010

[25] Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2:1–127.

[26] Salakhutdinov, R. (2009). *Learning Deep Generative Models*. PhD thesis, University of Toronto.

[27] Cho, K., Ilin, A., and Raiko, T. (2011a). Improved Learning of Gaussian-Bernoulli Restricted Boltzmann Machines. In *Proceedings of the Twentith International Conference on Artificial Neural Networks*, ICANN 2011.

[28] Cho, K., Raiko, T., and Ilin, A. (2010). Parallel Tempering is Efficient for Learning Restricted Boltzmann Machines. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2010)*, pages 3246 – 3253, Barcelona, Spain.

[29] Cho, K., Raiko, T., and Ilin, A. (2011b). Enhanced Gradient and Adaptive Learning Rate for Training Restricted Boltzmann Machines. In *Proceedings of the Twenty-seventh International Conference on Machine Learning*, ICML 2011.

[30] Cho, K., Raiko, T., and Ilin, A. (2011c). Gaussian-bernoulli deep boltzmann machine. In *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, Sierra Nevada, Spain.

[31] Fischer, A. and Igel, C. (2010). Empirical analysis of the divergence of Gibbs sampling based learning algorithms for restricted Boltzmann machines. In *Proceedings of the 20th international conference on Artificial neural networks: Part III*, ICANN'10, pages 208–217, Berlin, Heidelberg. Springer-Verlag.

[32] Schulz, H., Müller, A., and Behnke, S. (2010). Investigating Convergence of Restricted Boltzmann Machine Learning. In *NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning*.

[33] Raiko, T., Valpola, H., and LeCun, Y. (2011). Deep Learning Made Easier by Linear Transformations in Perceptrons. In *NIPS 2011 Workshop on Deep Learning and Unsupervised Feature Learning*, Sierra Nevada, Spain.

[34] T. Raiko and H. Valpola. Chapter 7: Oscillatory Neural Network for Image Segmentation with Biased Competition for Attention. In From Brains to Systems: Brain-Inspired Cognitive Systems 2010, Advances in Experimental Medicine and Biology, volume 718, pages 75–86, Springer New York, 2011.

[35] V. Lämsä and T. Raiko. Novelty Detection by Nonlinear Factor Analysis for Structural Health Monitoring. In IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2010), pages 468–473, Kittilä, Finland, August, 2010.

[36] K. Nyberg, T. Raiko, T. Tiinanen, and E. Hyvönen. Document Classification Utilising Ontologies and Relations between Documents, In Proceedings of the Eighth Workshop on Mining and Learning with Graphs (MLG 2010), Washington DC, USA, July, 2010.