Approximating Nonlinear Transformations of Probability Distributions for Nonlinear Independent Component Analysis

Antti Honkela
Neural Networks Research Centre, Helsinki University of Technology
P.O. Box 5400, FI-02015 HUT, Finland, E-mail: antti.honkela@hut.fi

***Abstract*— The nonlinear independent component analysis method introduced by Lappalainen and Honkela in 2000 uses a truncated Taylor series representation to approximate the nonlinear transformation from sources to observations. The approach uses information only at the single point of input mean and can produce poor results if the input variance is large. This feature has recently been identified to be the cause of instability of the algorithm with large source dimensionalities. In this paper, an improved approximation is presented. The derivatives used in the Taylor scheme are replaced with slopes evaluated by global Gauss-Hermite quadrature. The resulting approximation is more accurate under high input variance and the new learning algorithm more stable with high source dimensionalities.**

I. Introduction

Variational Bayesian learning has recently become very popular in the field of independent component analysis (ICA). Several authors have proposed methods based on applying a variational method called ensemble learning to a linear generative model with mixture-of-Gaussians source prior [1]–[4]. The same method can also be applied to nonlinear ICA or blind source separation (BSS) by replacing the linear generative model with a nonlinear one. In [5], the nonlinear mapping was modelled with a multilayer perceptron (MLP) network. The method has later also been extended to handle sources with temporal dependencies to create a powerful nonlinear state-space model [6], [7].

The variational Bayesian learning algorithm requires evaluation of certain statistics of the outputs of the model, given a distribution of parameter values. In case of a linear model, the statistics can be evaluated exactly, but with a nonlinear model they have to be approximated. In [5], the nonlinearity was handled by replacing it with a truncated Taylor series approximation. The method is simple and works well when the variance of the inputs is small enough. In cases of high input variance, however, the approximation loses accuracy. This has recently been identified to be the cause of instability of the algorithm with high source dimensionalites.

In this paper, a better approximation for the statistics of a nonlinear transform of a probability distribution is presented. The approximation is based on the idea of linearising the nonlinearity, but instead of the derivatives used in the Taylor scheme, different slopes evaluated with a global Gauss-Hermite quadrature method are used. The change is found to improve the accuracy of the approximation significantly and help avoid the stability problems.

The rest of the paper is organised as follows. The nonlinear ICA method is briefly introduced in Sec. II. Methods for approximating nonlinear transformations of probability distributions are introduced first in a general setting in Sec. III, and with specific application to MLP network in Sec. IV. The new approximation method is also presented in Sec. IV. In Sec. V, an experimental comparison of the methods is presented. The paper ends with conclusions in Sec. VI.

II. Nonlinear ICA by variational Bayesian learning

Let us denote the observed data by $\boldsymbol{X} = \{\mathbf{x}(t)|t\}$. Given the data, the goal is to estimate the sources $\boldsymbol{S} = \{\mathbf{s}(t)|t\}$ and other model parameters $\boldsymbol{\theta} = \{\theta_i|i\}$. The MLP network model for the observations can be written as

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t); \mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}) + \mathbf{n}(t) = \mathbf{B}\boldsymbol{\phi}(\mathbf{A}\mathbf{s}(t) + \mathbf{a}) + \mathbf{b} + \mathbf{n}(t), \quad (1)$$

where $\mathbf{n}(t)$ is Gaussian noise and $\boldsymbol{\phi}$ is the nonlinear activation function of the hidden neurons. The weights $\boldsymbol{W} = \{\mathbf{A}, \mathbf{B}, \mathbf{a}, \mathbf{b}\}$ are elements of $\boldsymbol{\theta}$ along with the parameters governing $\mathbf{n}(t)$ and hyperpriors of other parameters. The sources are assumed to be independent and have either a Gaussian mixture prior for nonlinear ICA or a Gaussian prior for simpler nonlinear factor analysis (NFA). In the latter case, the method can be extended to perform ICA by using a linear ICA algorithm as postprocessing for the extracted sources [7]. Because of this, the rest of the paper deals with the simpler NFA method.

As a variational Bayesian method, ensemble learning is based on approximating the posterior probability distribution of the sources and model parameters $p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})$ with a simpler tractable distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$. The approximation is fitted by minimising the Kullback-Leibler divergence between the approximation and the true posterior

$$D(q(\boldsymbol{S}, \boldsymbol{\theta})||p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})) = \left\langle \log \frac{q(\boldsymbol{S}, \boldsymbol{\theta})}{p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})} \right\rangle, \quad (2)$$

where $\langle\cdot\rangle$ denotes expectation over the distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$ [8], [9]. The approximation is set to be of fixed simple form, such as a multivariate Gaussian with a diagonal covariance used in NFA.

The learning algorithm used in the nonlinear factor analysis method is in principle very simple. The cost function in Eq. (2) can mostly be evaluated exactly, up to an additive constant. The only difficulties arise from the likelihood term

$$\begin{aligned}\mathcal{C}_x &= \langle -\log p(\boldsymbol{X}|\boldsymbol{S},\boldsymbol{\theta})\rangle \\ &= \sum_t \langle -\log N(\mathbf{x}(t);\ \mathbf{f}(\mathbf{s}(t),\boldsymbol{W}),\boldsymbol{\Sigma}_x)\rangle\end{aligned} \tag{3}$$

that has to be approximated somehow. Here $N(\mathbf{x};\ \overline{\mathbf{x}},\boldsymbol{\Sigma}_x)$ denotes a Gaussian density for variable $\mathbf{x}$ having mean $\overline{\mathbf{x}}$ and covariance $\boldsymbol{\Sigma}_x$. Assuming such an approximation can be found, the whole learning can be performed by numerically minimising the cost using e.g. simple gradient descent.

Assuming a Gaussian noise model with diagonal noise covariance, the problem of approximating $\mathcal{C}_x$ reduces to finding good approximations for the mean

$$\overline{\mathbf{f}}(\mathbf{s},\boldsymbol{W}) = \left\langle \mathbf{f}(\mathbf{s},\boldsymbol{W})\right\rangle \tag{4}$$

and diagonal elements of the covariance

$$\widetilde{f}_i(\mathbf{s},\boldsymbol{W}) = \left\langle \left(f_i(\mathbf{s},\boldsymbol{W}) - \overline{f}_i(\mathbf{s},\boldsymbol{W})\right)^2\right\rangle \tag{5}$$

of the outputs of the MLP network.

III. Gaussian integration

The mean of a nonlinear function $\mathbf{g}$ of $\mathbf{y}\sim N(\overline{\mathbf{y}},\boldsymbol{\Sigma}_y)$ can be written as integral of a given function with a Gaussian weight

$$I(\mathbf{g}) = \overline{\mathbf{g}}(\mathbf{y}) = \int_{\mathbb{R}^n} \mathbf{g}(\mathbf{y}) N(\mathbf{y};\ \overline{\mathbf{y}},\boldsymbol{\Sigma}_y) d\mathbf{y}. \tag{6}$$

With this formalism, the covariance can be written as

$$\begin{aligned}I_{\text{cov}}(\mathbf{g}) &= \int_{\mathbb{R}^n} (\mathbf{g}(\mathbf{y}) - \overline{\mathbf{g}}(\mathbf{y}))(\mathbf{g}(\mathbf{y}) - \overline{\mathbf{g}}(\mathbf{y}))^T N(\mathbf{y};\ \overline{\mathbf{y}},\boldsymbol{\Sigma}_y) d\mathbf{y} \\ &= I\left((\mathbf{g}(\cdot) - \overline{\mathbf{g}}(\mathbf{y}))(\mathbf{g}(\cdot) - \overline{\mathbf{g}}(\mathbf{y}))^T\right)\end{aligned} \tag{7}$$

The problem of multivariate Gaussian integration has many applications and has thus been studied widely. It is needed in many problems in physics and mathematical finance, but the methods used in these applications seem to concentrate more on the accuracy of the approximation and use computationally intensive Monte Carlo and quasi-Monte Carlo methods [10]. Unfortunately, these methods are computationally too heavy for machine learning applications such as nonlinear Kalman filtering and nonlinear ICA, where the integrals are needed as part of an iterative algorithm. Therefore e.g. nonlinear Kalman filtering methods use either a simple Taylor approximation (extended Kalman filtering) or a simple quadrature with very few points (unscented Kalman filtering).

Finding good approximations for high-dimensional Gaussian integrals is in general very difficult. In [11] it is shown that when the required precision ϵ approaches zero, the worst-case complexity for evaluating that good approximation of the integral is of the order ϵ^{-d}, where d is the dimensionality of the input. In the examples of Sec. V, for instance, these dimensionalities are typically of the order of 1000. In case of fixed precision, however, dimensions with sufficiently small input variance can be safely ignored, thus limiting the growth of the complexity somewhat.

A. First-order Taylor approximation

One of the simplest methods to evaluate the Gaussian integrals of Eqs. (6) and (7) is to substitute $\mathbf{g}(\mathbf{y})$ with a first-order Taylor approximation $\mathbf{g}(\overline{\mathbf{y}}) + D\mathbf{g}(\overline{\mathbf{y}})(\mathbf{y}-\overline{\mathbf{y}})$ about the mean. This approach is used for example in the extended Kalman filter. The resulting approximate mean is

$$\overline{\mathbf{g}}(\mathbf{y})_{\text{taylor}} = \mathbf{g}(\overline{\mathbf{y}}) \tag{8}$$

and covariance

$$\widetilde{\mathbf{g}}(\mathbf{y})_{\text{taylor}} = (D\mathbf{g}(\overline{\mathbf{y}}))\boldsymbol{\Sigma}_y(D\mathbf{g}(\overline{\mathbf{y}}))^T, \tag{9}$$

where $D\mathbf{g}(\overline{\mathbf{y}})$ is the Jacobian matrix of $\mathbf{g}$ evaluated at the point $\overline{\mathbf{y}}$.

In case of the mean, the approximation can relatively easily be extended to second-order by using only second-order information of the inputs as

$$\overline{g}_i(\mathbf{y})_{\text{taylor2}} = g_i(\overline{\mathbf{y}}) + \frac{1}{2}\operatorname{trace}(D^2 g_i(\overline{\mathbf{y}})\boldsymbol{\Sigma}_y). \tag{10}$$

In case of variance, the second-order approximation requires higher order statistics of the inputs. The second-order approximation is more accurate in case of low input variance, but it adds some new problems. In case of bounded $\mathbf{g}$, for instance, the mean estimates given by Eq. (10) are unbounded whereas those given by the first-order approximation in Eq. (8) are bounded. It is therefore not obvious that the second-order approximation should always be preferred over the first-order variant.

B. Gauss-Hermite quadrature

Gauss-Hermite quadrature is a method for evaluating numerically one-dimensional Gaussian integrals. The method can be iterated and thus applied in higher dimensions as well, but the number of function evaluations grows exponentially so it is not very practical in high dimensions.

The Gauss-Hermite quadrature approximation for one dimensional version of Eq. (6) with $y\sim N(\overline{y},\widetilde{y})$ is of the form

$$I_{\text{Gauss-Hermite}}(g) = \sum_{i=1}^{N} w_i g\left(\overline{y} + \sqrt{\widetilde{y}}t_i\right), \tag{11}$$

where the *abscissas* t_i and *weights* w_i are determined by requiring the approximation to be exact for polynomials up to a suitable degree. The number of points used can be determined by the level of accuracy needed.

Using these, the Gauss-Hermite approximation for mean and variance of a scalar function can be written as

$$\overline{g}(y)_{\text{GH}} = \sum_{i=1}^{N} w_i g(\overline{y} + \sqrt{\widetilde{y}}t_i) \tag{12}$$

and

$$\widetilde{g}(y)_{\text{GH}} = \sum_{i=1}^{N} w_i \left(g(\overline{y} + \sqrt{\widetilde{y}}t_i) - \overline{g}(y)\right)^2. \tag{13}$$

C. The unscented transform

The unscented transformation proposed by Julier and Uhlmann in 1996 [12] was designed to overcome the deficiencies of the Taylor approximation used in extended Kalman filter. The resulting unscented Kalman filter has since been developed further e.g. in [13].

In one dimension, the unscented transform is mostly equivalent to Gauss-Hermite quadrature. In higher dimensions, however, the number of points used in the approximation grows much more slowly with the dimensionality. In an n-dimensional case, the unscented transform is based on selecting a set $\mathcal{Y}$ of $2n$ weighted points together with the mean point that describe well the input distribution. In case of diagonal input covariance, the points will reside on the coordinate axes at a distance governed by corresponding standard deviation. These points are then transformed individually to get a new set of points $\mathcal{Z}_i = \mathbf{g}(\mathcal{Y}_i)$. The output mean and covariance are then computed as weighted mean and covariance of the transformed points $\mathcal{Z}$.

The unscented transform is intuitively appealing. With a suitable selection of points and their weights, it can achieve second-order accuracy with respect to Taylor approximation of the nonlinear transform. Additionally some information of higher order statistics of the input can be incorporated in the selection of the points. The non-local nature of the unscented transform also promises better accuracy for cases with high input variance in which the Taylor based approximations fail.

Despite its benefits, the unscented transform is not without drawbacks. As noted in the beginning of this section, maintaining the same level of accuracy of the approximation under increasing input dimensionality requires exponential increase in the number of function evaluations. The number of function evaluations used in unscented transform grows only linearly, so the accuracy of the approximation is bound to decrease as the dimensionality increases. Quantifying the decrease is, however, difficult, because the complexity result presented above is only valid on the limit of vanishing error.

The unscented transform is a good method for evaluating simple Gaussian integrals in low dimensions. In higher dimensions, choosing only two points for each dimension loses too much information and the results suffer.

IV. Application to the MLP

The original nonlinear factor analysis method [5] uses the first-order Taylor scheme of Eq. (9) for approximation of the variance and the second-order scheme of Eq. (10) for the mean. Looking at the results of earlier real experiments reported e.g. in [5]–[7] and the experimental analysis of approximation accuracy presented later in Sec. V-A, the method works very well when the input variance is low enough. Only when the number of estimated sources, and along with it the input variance, grows too large, the method will run into trouble.

The non-local nature of the unscented transform suggests that it should be better able to handle the cases of high input variance. This is confirmed by the results of experiments presented later in Sec. V-A. Unfortunately, the unscented transform seems to produce surprisingly poor results in low noise conditions and is thus as such unsuitable replacement for the old Taylor approximation. The poor results are probably due to the form of the function represented by the MLP including correlations caused by products of different input variables. These can be easily handled with minor extension to the Taylor approximation but are neglected by the unscented transform.

Overall, the linear parts of the MLP are easy to handle exactly. The only difficulties are caused by the nonlinearities, i.e. the activation functions of the hidden neurons. If those were replaced with linear functions, the whole network would be linear and even the simplest Taylor approximation would be exact. Because of this, it would seem reasonable to try to improve the Taylor approximation by using a more sophisticated approximation for those scalar functions without changing the whole scheme. The new method is thus basically the old Taylor approximation but with a linearisation of the activation functions based on the Gauss-Hermite quadrature instead of actual Taylor series expansion about the mean of the input.

A. First-order Taylor approximation

Let us examine the position of the activation function $\phi(y)$ of the hidden neurons of the MLP network in the Taylor approximation. According to the first-order Taylor approximation, the mean of the output is

$$\overline{\mathbf{f}}(\mathbf{s}, \boldsymbol{W}) = \overline{\mathbf{B}}\boldsymbol{\phi}(\overline{\mathbf{A}}\overline{\mathbf{s}} + \overline{\mathbf{a}}) + \overline{\mathbf{b}} \tag{14}$$

and the variance

$$\begin{aligned}\widetilde{f}_i(\mathbf{s}, \boldsymbol{W}) = {} & \nabla_{\mathbf{s}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}}) \boldsymbol{\Sigma}_{\mathbf{s}} \nabla_{\mathbf{s}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}})^T \\ & + \nabla_{\boldsymbol{W}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}}) \boldsymbol{\Sigma}_{\boldsymbol{W}} \nabla_{\boldsymbol{W}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}})^T,\end{aligned} \tag{15}$$

where the weights and sources are assumed to have distributions $\boldsymbol{W} \sim N(\overline{\boldsymbol{W}}, \boldsymbol{\Sigma}_{\boldsymbol{W}})$ and $\mathbf{s} \sim N(\overline{\mathbf{s}}, \boldsymbol{\Sigma}_{\mathbf{s}})$. The required derivative with respect to the inputs, for instance, is

$$\nabla_{\mathbf{s}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}}) = \overline{\mathbf{B}}_i \operatorname{diag}(\boldsymbol{\phi}'(\overline{\mathbf{A}}\overline{\mathbf{s}} + \overline{\mathbf{a}}))\overline{\mathbf{A}}, \tag{16}$$

where $\overline{\mathbf{B}}_i$ denotes the ith row of the mean matrix $\overline{\mathbf{B}}$ and $\operatorname{diag}(\mathbf{z})$ denotes a diagonal matrix with elements of vector $\mathbf{z}$ on its main diagonal. From this it is clear that both approximations can be broken into parts, i.e. evaluating the mean and variance of $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{a}$ first, then those of $\boldsymbol{\phi}(\mathbf{y})$ and finally those of $\mathbf{B}\boldsymbol{\phi}(\mathbf{y})$. The only approximations are done in the middle step, the first and the last are exact.

B. Gauss-Hermite approximation of hidden neurons

As noted before, the above method fails in case of high input variance because it relies on information of the activation function at a single point. To this end, an alternative approximation for the second step, evaluation of mean and variance of $\boldsymbol{\phi}(\mathbf{y})$, is proposed. Because of the nature of $\boldsymbol{\phi}$, the problem splits naturally to one-dimensional subproblems concerning each component separately. These can then be handled easily by

applying the Gauss-Hermite quadrature introduced in Sec. III-B. In order to keep the computational load reasonable, an approximation with three points was used. This also makes the procedure equivalent to applying the unscented transform to $\boldsymbol{\phi}(\mathbf{y})$.

Once the mean $\overline{\phi}(y_i)_{\mathrm{GH}}$ and variance $\widetilde{\phi}(y_i)_{\mathrm{GH}}$ as given in Eqs. (12) and (13) are known, it is easy to return back to the computations implied by the Taylor scheme by setting

$$\phi(y_i) := \overline{\phi}(y_i)_{\mathrm{GH}} \tag{17}$$

and

$$\phi'(y_i) := \sqrt{\frac{\widetilde{\phi}(y_i)_{\mathrm{GH}}}{\widetilde{y}_i}}, \tag{18}$$

where $\widetilde{y}_i$ is the variance of y_i. These formulae can be seen to define a global linearisation of the activation function in a sense that is optimal with respect to the assumed Gaussian input.

C. Computational considerations

Above, the new approximation has been derived for network inputs only. Corresponding approximations are needed for network weights as well, but they can be derived in a similar manner. The dependence of the outputs from the second layer weights $\mathbf{B}$ and $\mathbf{b}$ is linear so it can be handled trivially. The derivatives of the output with respect to first layer weights $\mathbf{A}$ and $\mathbf{a}$ are

$$\nabla_{\mathbf{A}_j} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}}) = \overline{B}_{ij}\phi'((\overline{\mathbf{A}}\overline{\mathbf{s}})_j + \overline{a}_j)\overline{s}, \tag{19}$$

where $\mathbf{A}_j$ is the jth row of matrix $\mathbf{A}$, and

$$\nabla_{\mathbf{a}} f_i(\overline{\mathbf{s}}, \overline{\boldsymbol{W}}) = \overline{\mathbf{B}}_i \operatorname{diag}(\boldsymbol{\phi}'(\overline{\mathbf{A}}\overline{\mathbf{s}} + \overline{\mathbf{a}})). \tag{20}$$

The Equations (16), (19) and (20) combined with the corresponding covariance matrices of the parameters each imply a different variance for the input $\mathbf{y}$ of the activation functions. Additionally, none of these is equal to the total variance of $\mathbf{y}$ which would seem the most natural choice for evaluating the approximation for the mean. Evaluating these four separate Gauss-Hermite approximations is computationally expensive, so combining them to evaluate several quantities with a single expansion is preferable.

Most of the variance of $\mathbf{y}$ is due to the variance of the sources, so using a common approximation for the source variance and mean introduces only very small errors. Unfortunately, using the same approximation also for the weights introduces significant errors, so another one must be used jointly for both $\mathbf{A}$ and $\mathbf{a}$.

V. Experimental results

In this section, experimental results on the accuracy and performance of the different approximations are presented. In the first experiment, the accuracy of the approximations is studied with random MLP networks and random inputs. In other experiments, the proposed method is compared to the original Taylor approximation in nonlinear factor analysis.[1]

[1]The Matlab code used in the NFA experiments is available at `http://www.cis.hut.fi/projects/bayes/software/`.

A. Approximation accuracy

In this experiment, the accuracy of different approximations was studied. The accuracy was evaluated by testing the approximations with 500 random input distributions, each with 100 random MLP networks. The means of the weights of the MLP networks were sampled randomly from a unit variance Gaussian distribution. The covariance of the weights was assumed to be diagonal with variances of the weights all equal to 10^{-3}. 100 input means were also randomly sampled from a unit variance Gaussian distribution. Five different values were tested for the variances at each of the input means, $10^{-3}, 10^{-2}, 10^{-1}, 1$ and 10. The results were then compared to assumed correct solutions evaluated with a Monte Carlo method. The dimensions of the MLP network were 5-30-10, i.e. 5 input neurons, 30 hidden neurons and 10 output neurons.

The results of the experiment are shown in Figs. 1, 2 and 3. Fig. 1 shows the mean squared error of different mean approximations. It confirms the suspicion that second-order Taylor approximation is better than first-order with low variance but worse with high variance. The unscented transform is surprisingly worse than even first-order Taylor approximation. The proposed method provides the best results on all levels of input variance.

The mean squared errors of different variance approximations on logarithmic scale are shown in Fig. 2. The most notable result is the rapid drop in accuracy of the Taylor approximation. The unscented transform and the proposed method provide more stable results with the proposed method being clearly better on all noise levels. Fig. 3 shows the maximum amount different methods underestimate the output variance. The plot shows the ratio of the true variance and estimated variance, so value 1 would be the optimal result. This result is shown separately because it is probably the most harmful type of error for our application. The results are rather similar to the mean squared errors of the variance estimate.

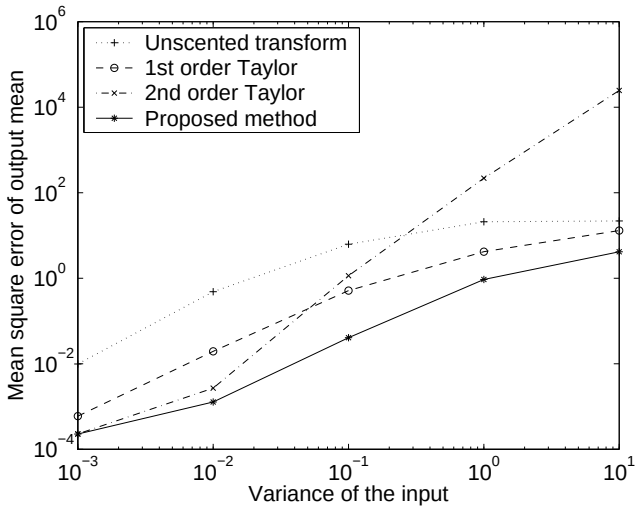

Fig. 1. Mean squared error of different mean approximations as a function of the input variance.

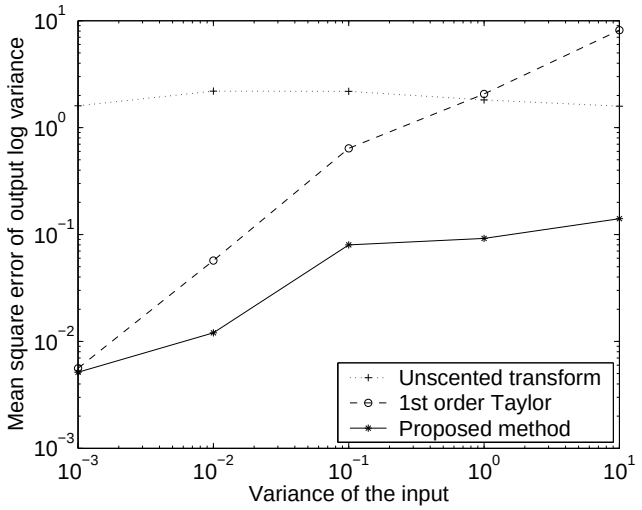

Fig. 2. Mean squared error of logarithm of different variance approximations as a function of the input variance.

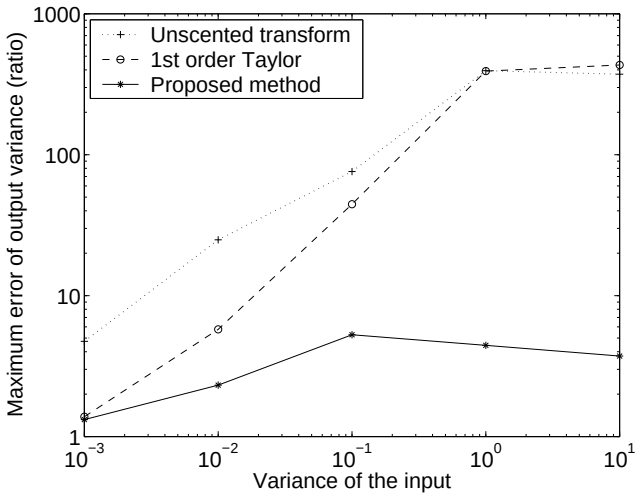

Fig. 3. The ratio of true variance and estimated variance of the greatest underestimation of output variance for different approximations as a function of the input variance.

B. Nonlinear factor analysis with artificial data

The nonlinear factor analysis method using the new approximation was tested using the same artificial data set that was used in [5]. The data set consisted of 20-dimensional vectors that were formed by mapping 4 sub-Gaussian and 4 super-Gaussian sources nonlinearly with a random MLP network. The number of samples was 1000. The results were evaluated by the signal-to-noise ratio of the sources recovered by optimal linear reconstruction from the estimated sources to the true sources. The additional linear reconstruction was used because the NFA method cannot find the correct rotation for the sources. The rotation could be recovered blindly using a linear ICA algorithm, but this was not used as it would have increased the computational burden and added another possible source of error.

The results of the experiment are shown in Fig. 4. With 10 sources, the algorithm using the Taylor approximation produces better results than the proposed approximation. The results attained by the Taylor algorithm do, however, yield significantly lower cost function value than the results of the proposed algorithm also with the new approximation, so the problem is due to suboptimal optimisation algorithm. When the number of sources is increased to 15, the Taylor algorithm can no longer produce any reasonable results. Even the best of the 12 simulations starts to diverge right at the start. The results of the proposed algorithm are affected by the increase in the number of sources only slightly.

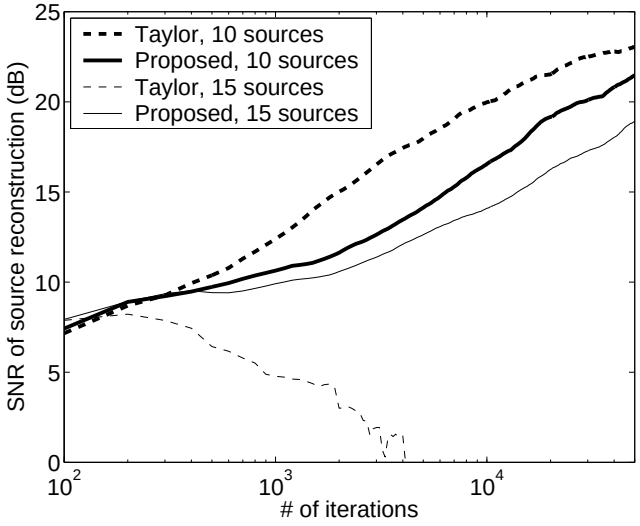

Fig. 4. Comparison of signal to noise ratios attained with different approximations in the NFA algorithm using either 10 or 15 sources. The results show the best of 12 simulations with different random initialisations at each point.

C. Nonlinear factor analysis with natural data

The new method was tested on natural data with the speech data compression experiment also presented in [7]. The data set used in the experiment consisted of spectrograms of 24 individual words of Finnish speech, spoken by 20 different speakers. The spectra were modified to mimic the reception abilities of the human ear. This is a standard preprocessing procedure for speech recognition. The preprocessed data consisted of 2547 30-dimensional spectrogram vectors.

Linear factor analysis as well as nonlinear factor analysis with both old Taylor based approximation and proposed new approximation were applied to the data to extract different number of sources or factors. The nonlinear factor analysis methods were run for 10000 iterations. Fig. 5 shows the residual energy left unexplained by the given number of sources. Nonlinear factor analysis is able to explain the data equally well with fewer factors than the linear method. The differences between different approximations in nonlinear

factor analysis are mostly small. The proposed approximation is able to reliably estimate even 20 components while the Taylor approximation method consistently fails when trying to estimate more than 13. It is possible that using many different random initialisations might help the Taylor method estimate a few more components as shown in the results reported in [7], but the difference in the stability of the methods is very clear.

Additionally, in the cases where the Taylor approximation produces better results than the proposed method, the cost function value of those simulations is also lower when evaluated using the proposed approximation than the one attained in the actual simulation using the proposed method. The worse results are therefore due to the optimisation method used with the new approximation and should be remediable by improving the optimisation algorithm.

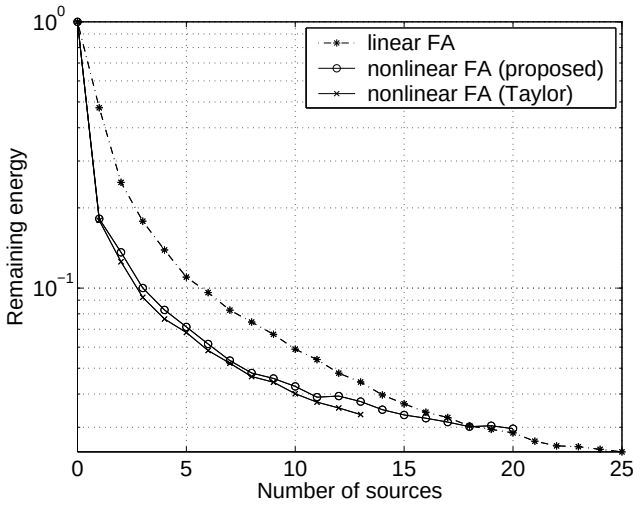

Fig. 5. The remaining (residual) energy of the speech data as a function of the number of extracted components using linear factor analysis and nonlinear factor analysis with proposed approximation and Taylor approximation. There are no results for Taylor approximation with more than 13 components because all the simulations diverged.

VI. Conclusions

In this paper, a new method for estimating the mean and the variance of a nonlinear transform of a probability distribution was proposed. The method is especially designed for use with nonlinear transforms modelled by MLP networks. It is based on standard first-order Taylor method of linearising the mapping about the input mean, except that the derivatives of the nonlinear activation function are replaced by slopes evaluated globally by Gauss-Hermite quadrature. The global nature of the approximation increases its accuracy with large input variances significantly while guaranteeing second-order accuracy for cases of small input variance.

The new approximation was used to derive a new learning algorithm for the nonlinear factor analysis (NFA) model originally proposed in [5]. The new algorithm was able to avoid the stability problems from which the old algorithm suffered, but unfortunately the results suffered slightly. The better optima found by the old algorithm are also clearly better with respect to the cost function evaluated with the new method, so the problem is most likely due to the highly tuned optimisation algorithm used in the NFA method. The optimisation algorithm was designed for the simpler approximation and may thus not work in the desired manner with the more complicated new method. In future, the complicated hand-tuned optimisation algorithm should be replaced with something more suitable for the new approximation.

The computation time required by the nonlinear factor analysis algorithm using the new approximation is larger than with the old Taylor approximation. The increase is, however, typically less than 50 %.

Acknowledgements

The author wishes to thank Juha Karhunen, Erkki Oja and Tapani Raiko for useful comments and discussions. This work was supported by the Finnish Centre of Excellence Programme (2000–2005) under the project New Information Processing Principles.

References

[1] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.

[2] H. Lappalainen, "Ensemble learning for independent component analysis," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, Aussois, France, 1999, pp. 7–12.

[3] J. Miskin and D. J. C. MacKay, "Ensemble learning for blind source separation," in *Independent Component Analysis: Principles and Practice*, S. Roberts and R. Everson, Eds. Cambridge University Press, 2001, pp. 209–233.

[4] W. Penny, R. Everson, and S. Roberts, "ICA: model order selection and dynamic source models," in *Independent Component Analysis: Principles and Practice*, S. Roberts and R. Everson, Eds. Cambridge University Press, 2001, pp. 299–314.

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

[6] H. Valpola and J. Karhunen, "An unsupervised ensemble learning method for nonlinear dynamic state-space models," *Neural Computation*, vol. 14, no. 11, pp. 2647–2692, 2002.

[7] H. Valpola, E. Oja, A. Ilin, A. Honkela, and J. Karhunen, "Nonlinear blind source separation by variational Bayesian learning," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 3, pp. 532–541, 2003.

[8] G. E. Hinton and D. van Camp, "Keeping neural networks simple by minimizing the description length of the weights," in *Proc. of the 6th Ann. ACM Conf. on Computational Learning Theory*, Santa Cruz, CA, USA, 1993, pp. 5–13.

[9] D. J. C. MacKay, "Developments in probabilistic modelling with neural networks – ensemble learning," in *Neural Networks: Artificial Intelligence and Industrial Applications. Proc. of the 3rd Annual Symposium on Neural Networks*, 1995, pp. 191–198.

[10] J. F. Traub and A. G. Weschultz, *Complexity and Information*. Cambribge University Press, 1998.

[11] F. Curbera, "Delayed curse of dimension for Gaussian integration," *Journal of Complexity*, vol. 16, no. 2, pp. 474–506, 2000.

[12] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Robotics Research Group, Department of Engineering Science, University of Oxford, Tech. Rep., 1996.

[13] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York: Wiley, 2001, pp. 221–280.