Post-nonlinear Independent Component Analysis by Variational Bayesian Learning

Alexander Ilin and Antti Honkela

Helsinki University of Technology, Neural Networks Research Centre
P.O. Box 5400, FI-02015 HUT, Espoo, Finland
{alexander.ilin,antti.honkela}@hut.fi
http://www.cis.hut.fi/projects/bayes/

Abstract. Post-nonlinear (PNL) independent component analysis (ICA) is a generalisation of ICA where the observations are assumed to have been generated from independent sources by linear mixing followed by component-wise scalar nonlinearities. Most previous PNL ICA algorithms require the post-nonlinearities to be invertible functions. In this paper, we present a variational Bayesian approach to PNL ICA that also works for non-invertible post-nonlinearities. The method is based on a generative model with multi-layer perceptron (MLP) networks to model the post-nonlinearities. Preliminary results with a difficult artificial example are encouraging.

1 Introduction

The problem of ICA have been studied by many authors in recent years. The general goal of ICA is to estimate some unknown signals (or sources) from a set of their mixtures by exploiting only the assumption that the mixed signals are statistically independent. The linear ICA model is well understood (see e.g. [1] for review) while the general nonlinear ICA and related nonlinear blind source separation (BSS) are more difficult problems from both theoretical and practical points of view [2, 1]. In fact, the general nonlinear ICA problem is ill-posed and most approaches to it are better classified as nonlinear BSS, where the goal is to estimate the specific sources that have generated the observed mixtures.

Post-nonlinear mixtures are a special case of the nonlinear mixing model studied first by Taleb and Jutten [3]. They are interesting for their separability properties and plausibility in many real world situations. In the PNL model, the nonlinear mixture has the following specific form:

$$x_i(t) = f_i\left[\sum_{j=1}^{M} a_{ij} s_j(t)\right] \qquad i = 1, \ldots, N \tag{1}$$

where $x_i(t)$ are the N observations, $s_j(t)$ are the M independent sources, a_{ij} denotes the elements of the unknown mixing matrix A and $f_i : \mathbb{R} \to \mathbb{R}$ are a set of scalar to scalar functions sometimes also called post-nonlinear distortions.

C.G. Puntonet and A. Prieto (Eds.): ICA 2004, LNCS 3195, pp. 766–773, 2004.

Most of the existing ICA methods for PNL mixtures assume that the source vectors $\mathbf{s}(t)$ and the observations $\mathbf{x}(t)$ are of the same dimensionality (i.e. $N = M$) and that all post-nonlinear distortions f_i are invertible. In this case, under certain conditions on the distributions of the sources (at most one Gaussian source) and the mixing structure ($\mathbf{A}$ has at least 2 nonzero entries on each row or column), PNL mixtures are separable with the same well-known indeterminacies as in the linear mixtures [4, 3].

However, as was shown in [5], overdetermined PNL mixtures (when there are more observations x_i than sources s_j, i.e. $N > M$) can be separable even when some of the distortions f_i are non-invertible functions. In [5], the general nonlinear factor analysis (NFA) model [6]

$$\mathbf{x}(t) = \mathbf{f}\left(\mathbf{s}(t), \boldsymbol{\theta}_f\right) + \mathbf{n}(t) \tag{2}$$

followed by the linear FastICA post-processing [1] was successfully applied to recover the independent sources from this kind of PNL mixtures.

In the present paper, we restrict the general NFA model of Eq. (2) to the special case of PNL mixtures of Eq. (1) and derive a learning algorithm based on variational Bayesian learning. In the resulting model, which we call post-nonlinear factor analysis (PNFA), the sources $s_j(t)$ are assumed to be Gaussian and therefore the nonlinear ICA problem can be solved by first learning the roughly Gaussian sources and then rotating them using any linear ICA algorithm to recover the independent components [6, 7].

The rest of the paper is structured as follows. First, the PNFA model is introduced in Sec. 2. The learning algorithm used to estimate the model is presented in Sec. 3 and the results of an experiment with a difficult artificial example in Sec. 4. The paper concludes with discussion in Sec. 5.

2 Post-nonlinear Factor Analysis Model

Most PNL ICA methods [3, 8] separate sources by inverting the mixing model (1) and therefore by estimating the following separating structure

$$s_j(t) = \sum_{i=1}^{N} b_{ji} g_i(x_i(t), \boldsymbol{\theta}_i) \qquad j = 1, \ldots, M. \tag{3}$$

This approach implicitly assumes the existence of the inverse of the component-wise nonlinearities $g_i = f_i^{-1}$, and therefore fails in separable PNL mixtures with non-invertible distortions f_i [5].

To overcome this problem, we present the Bayesian PNFA algorithm which instead learns the generative model (1) in the following form (see Fig. 1):

$$x_i(t) = f_i\left[y_i(t), \boldsymbol{W}_i\right] + n_i(t) = f_i\left[\sum_{j=1}^{M} a_{ij} s_j(t), \boldsymbol{W}_i\right] + n_i(t) \tag{4}$$

where $y_i(t) = \sum_{j=1}^{M} a_{ij} s_j(t)$ and $n_i(t)$ is the observation noise. The post-nonlinear component-wise distortions f_i are modelled by multi-layer preceptron (MLP) networks with one hidden layer:

$$f_i(y, \boldsymbol{W}_i) = \mathbf{D}_i \boldsymbol{\phi}(\mathbf{C}_i y + \mathbf{c}_i) + d_i. \tag{5}$$

Here the parameters $\boldsymbol{W}_i$ of the MLPs include the column vectors $\mathbf{C}_i$, $\mathbf{c}_i$, row vector $\mathbf{D}_i$ and scalar d_i. A sigmoidal activation function $\boldsymbol{\phi}$ that operates component-wise on its inputs is used.

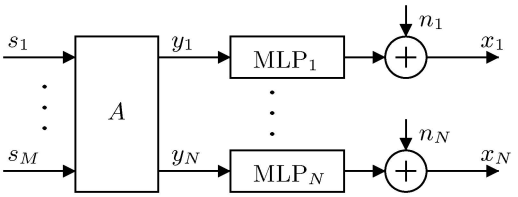

Fig. 1. The model structure of PNFA.

Implementing the Bayesian approach, we express all the model assumptions in the form of the joint distribution of the observations $\boldsymbol{X} = \{\mathbf{x}(t)|t\}$, the sources $\boldsymbol{S} = \{\mathbf{s}(t)|t\}$ and other model parameters $\boldsymbol{\theta} = \{\theta_i|i\}$.

Assuming independent Gaussian noise $n_i(t)$ yields the likelihood

$$p(\boldsymbol{X} \mid \boldsymbol{S}, \boldsymbol{\theta}) = \prod_{i,t} N\left(x_i(t);\ f_i\left[y_i(t), \boldsymbol{W}_i\right],\ e^{2v_{n,i}}\right) \tag{6}$$

where $N\left(x;\ \mu,\ \sigma^2\right)$ denotes a Gaussian density for variable x having mean μ and variance σ^2, and the variance parameter has lognormal hierarchical prior. The sources $s_j(t)$ are assumed to be Gaussian and have the prior

$$p(\boldsymbol{S} \mid \boldsymbol{\theta}) = \prod_{j,t} N\left(s_j(t);\ 0,\ e^{2v_{s,j}}\right). \tag{7}$$

The parameters of the prior distributions (such as the variance parameters $v_{n,i}$, $v_{s,j}$) as well as the other model parameters (such as the parameters $\boldsymbol{W}_i$ of the component-wise MLPs) are further assigned Gaussian priors making the prior $p(\boldsymbol{\theta})$ of the parameters hierarchical. For example, the noise parameters $v_{n,i}$ of different components of the data share a common prior:

$$p(v_{n,i} \mid \boldsymbol{\theta} \setminus v_{n,i}) = N\left(v_{n,i};\ m_{v_n},\ e^{2v_{v_n}}\right) \tag{8}$$

and the hyperparameters m_{v_n}, v_{v_n} have very flat Gaussian priors.

3 Learning

In this section, the variational Bayesian learning algorithm used to learn the model, is introduced.

3.1 Variational Bayesian Learning

The PNFA model is learned using variational Bayesian method called ensemble learning [9–11]. It has recently become very popular in linear ICA [12–15] but it has been applied to nonlinear BSS [6, 16, 7] as well. Reasons for the popularity of ensemble learning include the ability to easily compare different models and its resistance to overfitting, which is especially important in applications with nonlinear models.

As a variational Bayesian method, ensemble learning is based on approximating the posterior distribution of the sources and model parameters $p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})$ with another, simpler distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$. The approximation is fitted by minimising the cost function

$$\mathcal{C} = \left\langle \log \frac{q(\boldsymbol{S}, \boldsymbol{\theta})}{p(\boldsymbol{S}, \boldsymbol{\theta}, \boldsymbol{X})} \right\rangle = D_{KL}(q(\boldsymbol{S}, \boldsymbol{\theta})||p(\boldsymbol{S}, \boldsymbol{\theta}|\boldsymbol{X})) - \log p(\boldsymbol{X}) \tag{9}$$

where $\langle \cdot \rangle$ denotes expectation over the distribution $q(\boldsymbol{S}, \boldsymbol{\theta})$ and $D_{KL}(q||p)$ is the Kullback-Leibler divergence between the distributions q and p. The approximation is restricted to be of fixed simple form, such as a multivariate Gaussian with a diagonal covariance used in PNFA.

3.2 Learning the Model

Most terms of the cost function in Eq. (9) are simple expectations over Gaussian variables that can be evaluated analytically. The only difficulties arise from the likelihood term

$$\mathcal{C}_x = \langle -\log p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{\theta}) \rangle \tag{10}$$

that has to be approximated somehow.

With the Gaussian noise model, the likelihood term can be written as

$$\begin{aligned} \mathcal{C}_x &= \sum_{t,i} \left\langle -\log N(x_i(t);\ f_{i,t},\ \sigma_n^2) \right\rangle \\ &= \sum_{t,i} \left[\frac{1}{2} \left\langle \log \sqrt{2\pi\sigma_n^2} \right\rangle + \left\langle \frac{1}{2\sigma_n^2} \right\rangle \left(\left[x_i(t) - \langle f_{i,t} \rangle \right]^2 + \mathrm{Var}[f_{i,t}] \right) \right] \end{aligned} \tag{11}$$

where $f_{i,t} = f_i\left[y_i(t), \boldsymbol{W}_i\right]$ and $\mathrm{Var}[\cdot]$ denotes variance under $q(\boldsymbol{S}, \boldsymbol{\theta})$. This can be thus evaluated if the mean and variance of the outputs of the MLP networks are known. Once the cost function can be computed, it can be minimised numerically. The minimisation is performed by a gradient based algorithm similar to one used in [6].

3.3 Evaluation of the Statistics of MLP Outputs

To simplify the notation, subindices i will be dropped in this section. The mean and variance of the inputs $y(t)$ of the MLP networks can be computed exactly. Assuming these are Gaussian, the mean and variance of the MLPs $f(y(t), \boldsymbol{W})$

can easily be evaluated using e.g. Gauss-Hermite quadrature, which in this scalar case for y using three points is equivalent to unscented transform.

The above discussion ignores the variance of the network weights $\boldsymbol{W}$. Their effect could be included by performing the unscented transform on full combined input of $y(t)$ and $\boldsymbol{W}$, but that would increase the computational burden too much. As the variances of the weights are usually small, their effects are represented sufficiently well by using first-order Taylor approximation of the network with respect to them [17]. Thus the mean of the output is approximated as

$$\langle f_t \rangle = \sum_j w_j f(\hat{y}_j(t), \overline{\boldsymbol{W}}) \tag{12}$$

where w_j are the weights and $\hat{y}_j(t) = \langle y(t) \rangle + t_j \operatorname{Var}[y(t)]^{1/2}$ are the basis points of the Gauss-Hermite quadrature corresponding to the abscissas t_j, and $\overline{\boldsymbol{W}}$ denotes the mean of the weights $\boldsymbol{W}$.

Correspondingly, the variance is approximated by a combined Gauss-Hermite and Taylor approximation

$$\begin{aligned} \operatorname{Var}[f_t] = \sum_j w_j \Big[& \Big(f(\hat{y}_j(t), \overline{\boldsymbol{W}}) - \langle f_t \rangle \Big)^2 \\ & + \nabla_{\boldsymbol{W}} f(\hat{y}_j(t), \overline{\boldsymbol{W}}) \operatorname{Cov}[\boldsymbol{W}] \nabla_{\boldsymbol{W}} f(\hat{y}_j(t), \overline{\boldsymbol{W}})^T \Big]. \end{aligned} \tag{13}$$

4 Experiments

The proposed PNFA algorithm was tested on a three-dimensional PNL mixture of two independent sources. The sources were a sine wave and uniformly distributed white noise. The PNL transformation used for generating the data contained two non-invertible post-nonlinear distortions:

$$\mathbf{y} = \begin{bmatrix} 1.2 & 0.2 \\ 1 & 0.7 \\ 0.2 & 0.8 \end{bmatrix} \mathbf{s} \qquad \mathbf{x} = \begin{bmatrix} (y_1 - 0.5)^2 \\ (y_2 + 0.4)^2 \\ \tanh(2y_3) \end{bmatrix}. \tag{14}$$

The observations were centered and normalised to unit variance and observation noise with variance 0.01 was added. The number of samples was 400.

The PNFA model was trained by trying different model structures, i.e. different numbers of hidden neurons in the PNL MLPs (5), and several random initialisations of the parameters to be optimised. The source initialisation was done by the principal component analysis of the observations. The best PNFA model[1] had 5 neurons in the hidden layers of all MLPs.

The PNL distortions learned by the best model after 10000 iterations is presented in Fig. 2: The post-nonlinearities f_i are estimated quite well except for

[1] The best model has the smallest value of the cost function (9) which corresponds to the maximum lower bound of the model evidence $p(\boldsymbol{X}|model)$.

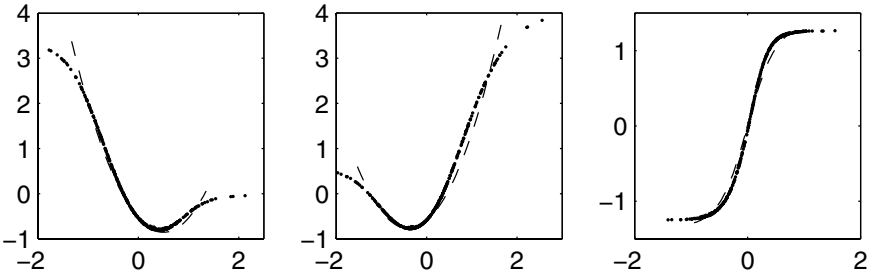

Fig. 2. The estimated post-nonlinear distortions f_i against the functions used for generating the data (the dashed line). Each point in the figure corresponds to a single observation.

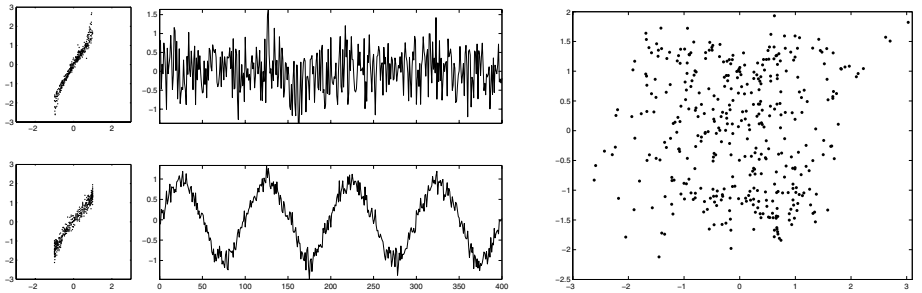

Fig. 3. The sources found by the PNFA and further rotated with the FastICA algorithm. (a) – the scatter plots; (b) – the estimated time series; (c) – the distribution of the sources. The signal-to-noise ratio is 12.95 dB.

some points at the edges. The difficulties mostly affect the two quadratic functions which are difficult to model with such small MLP networks and relatively few observations, especially at those edges.

The sources found by PNFA were further rotated by the FastICA algorithm to obtain independent signals (see Fig. 3). The scatter plots in Fig. 3a show how well the original sources were reconstructed. Each point corresponds to one source $s_i(t)$. The abscissa of a point is the original source which was used for generating the data and the ordinate is the estimated source. The optimal result would be a straight line which would mean that the estimated values of the sources coincide with the true values. Again, the sources were estimated quite well except for some points at the edges.

This result is somewhat natural due to the great difficulty of the test problem: There are only two bounded sub-Gaussian sources in the mixture and their linear combinations are quite far from Gaussianity assumed by PNFA. Another difficulty is the complex PNL mapping with a small number of observations and several non-invertible post-nonlinear distortions. Removing any of the observa-

tions from the mixture would make the mixing process non-injective and the separation problem unsolvable.

5 Discussion

In this paper, we presented a new Bayesian algorithm for learning the post-nonlinear mixing structure. The algorithm which we call post-nonlinear factor analysis is based on modelling the component-wise post-nonlinear distortions by MLP networks and using variational Bayesian learning.

An important feature of the proposed technique is that it learns the generative model of the observations while most existing PNL methods estimate the complementary separating structure. This makes the algorithm applicable to some post-nonlinear ICA problems unsolvable for the alternative methods.

We tested PNFA on a very challenging ICA problem and the obtained experimental results are very promising. The PNFA algorithm complemented by a linear ICA method was able to recover original sources from a globally invertible PNL mixture with non-invertible post-nonlinear distortions. This cannot be achieved by existing alternative methods [5].

The presented results are still preliminary and further investigations of the algorithm are needed. For example, the problem with local minima appears more severe for PNL mixtures with non-invertible distortions. Another interesting question is whether PNFA can improve the source restoration quality compared to the general NFA method applied to PNL problems.

An important issue is how the proposed PNL ICA technique works in higher-dimensional problems: Due to the Gaussianity assumption for the sources, the performance of the algorithm may be better for a greater number of mixed sources. Also, we are planning to implement a mixture-of-Gaussians model for the sources like in [12, 6] in order to improve the source estimation quality.

Acknowledgements

This work was partially done in the Lab. des Images et des Signaux at Institut National Polytechnique de Grenoble (INPG) in France. The authors would like to thank Sophie Achard, Christian Jutten and Harri Valpola for the fruitful discussions and help. This research has been partially funded by the European Commission project BLISS.

References

1. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. J. Wiley, 2001.
2. A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, 1999.
3. A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Trans. on Signal Processing*, vol. 47, no. 10, pp. 2807–2820, 1999.

4. C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pp. 245–256, 2003. Invited paper in the special session on nonlinear ICA and BSS.
5. A. Ilin, S. Achard, and C. Jutten, "Bayesian versus constrained structure approaches for source separation in post-nonlinear mixtures," in *Proc. International Joint Conference on Neural Networks (IJCNN 2004)*, 2004. To appear.
6. H. Lappalainen and A. Honkela, "Bayesian nonlinear independent component analysis by multi-layer perceptrons," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 93–121, Berlin: Springer-Verlag, 2000.
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. A. Taleb and C. Jutten, "Batch algorithm for source separation in postnonlinear mixtures," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, (Aussois, France), pp. 155–160, 1999.
9. 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), pp. 5–13, 1993.
10. 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*, pp. 191–198, 1995.
11. H. Lappalainen and J. Miskin, "Ensemble learning," in *Advances in Independent Component Analysis* (M. Girolami, ed.), pp. 75–92, Berlin: Springer-Verlag, 2000.
12. H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.
13. H. Lappalainen, "Ensemble learning for independent component analysis," in *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, (Aussois, France), pp. 7–12, 1999.
14. 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.), pp. 209–233, Cambridge University Press, 2001.
15. 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.), pp. 299–314, Cambridge University Press, 2001.
16. 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.
17. A. Honkela, "Approximating nonlinear transformations of probability distributions for nonlinear independent component analysis," in *Proc. International Joint Conference on Neural Networks (IJCNN 2004)*, 2004. To appear.