An Ensemble Learning Approach to Nonlinear Independent Component Analysis

Antti Honkela* and Juha Karhunen*

Abstract — **Blind extraction of independent sources from their nonlinear mixtures is generally a very difficult problem. This is because both the nonlinear mapping and the underlying sources are unknown, and must be learned in an unsupervised manner from the data. We use multilayer perceptrons as nonlinear generative models for the data, and apply Bayesian ensemble learning for optimizing the model. In this paper, we successfully apply this approach to real-world speech data.**

1 Introduction

It is fair to say that independent component analysis (ICA) and the closely related blind source separation (BSS) are now well understood problems when the observed data consists of linear instantaneous mixtures. Many well-performing algorithms have been introduced and analyzed for this case [3]. During the last years, several authors have tried to generalize linear ICA and BSS for nonlinear models; see [3, 4] for further information and references. A particular problem which makes nonlinear ICA much more difficult compared with the linear case is that the problem is highly non-unique without some suitable regularizing constraints [4, 3].

In this paper, the nonlinear mapping from the unknown sources to the observations is modeled with the familiar multi-layer perceptron (MLP) network [2]. However, the learning procedure is quite different from standard backpropagation, and is based on unsupervised ensemble learning. There the necessary regularization for nonlinear ICA is obtained by integrating over a full distribution for the model and sources instead of choosing only single values. This approach provides meaningful sources and nonlinear mapping, as shown by the experiments.

A similar generative model has been applied to the linear ICA/BSS problem in [1]. Our work uses a nonlinear data model, and applies a fully Bayesian treatment to the hyperparameters of the network or graphical model, too.

*Helsinki University of Technology, Neural Networks Research Centre, P.O. Box 5400, FIN-02015 HUT, Espoo, Finland. Email: Antti.Honkela@hut.fi, Juha.Karhunen@hut.fi, URL: http://www.cis.hut.fi.
This research has been funded by the European Commission project BLISS and the Finnish Center of Excellence Programme (2000 - 2005) under the project New Information Processing Principles.

2 Ensemble learning

A flexible model family, such as MLP networks, provides infinitely many possible explanations of different complexity for the observed data. In Bayesian learning all the possible explanations are taken into account and weighted according to their posterior probabilities. This approach optimally solves the tradeoff between under- and overfitting. All the relevant information needed in choosing an appropriate model is contained in the posterior probability density functions (pdfs) of different model structures.

In practice, exact treatment of the posterior pdfs of the models is impossible. Therefore, some suitable approximation method must be used. Ensemble learning [7, 6], which is a special case variational learning, is a recently developed method for parametric approximation of posterior pdfs where the search takes into account the probability *mass* of the models. Therefore, it does not suffer from overfitting. The basic idea in ensemble learning is to minimize the misfit between the posterior pdf and its parametric approximation.

Let us denote by $X = \{\mathbf{x}(t)|t\}$ the set of available data (mixture) vectors, by $S = \{\mathbf{s}(t)|t\}$ the respective source vectors, and by $\boldsymbol{\theta}$ all the unknown parameters of the data model. Furthermore, $P(S, \boldsymbol{\theta}|X)$ denotes the exact posterior pdf and $Q(S, \boldsymbol{\theta}|X)$ its parametric approximation. The misfit is measured with the Kullback-Leibler (KL) divergence C_{KL} between P and Q, defined by the cost function

$$C_{\mathrm{KL}} = \int Q(S, \boldsymbol{\theta}|X) \log \frac{Q(S, \boldsymbol{\theta}|X)}{P(S, \boldsymbol{\theta}|X)} d\boldsymbol{\theta} dS \qquad (1)$$

The Kullback-Leibler divergence measures the difference in the probability mass between the densities P and Q. Its minimum value 0 is achieved when the two densities are the same.

3 Model structure

We use MLP networks which have the universal approximation property for smooth continuous mappings [2]. They are well suited for modeling both strongly and mildly nonlinear mappings.

The data model used in this work is as follows. Let $\mathbf{x}(t)$ denote the observed data vector at time

t, and $\mathbf{s}(t)$ the vector of source signals (latent variables) at time t. The matrices $\mathbf{B}$ and $\mathbf{A}$ contain the weights of the output and the hidden layer of the network, respectively, and $\mathbf{a}$ is the bias vector of the hidden layer. The vector of nonlinear activation functions is denoted by $\mathbf{f}(\cdot)$, and $\mathbf{n}(t)$ is the Gaussian noise vector corrupting the observations. Using these notations, the data model is

$$\mathbf{x}(t) = \mathbf{B}\left[\mathbf{f}\left(\mathbf{A}\mathbf{s}(t) + \mathbf{a}\right)\right] + \mathbf{n}(t). \tag{2}$$

We have used as the activation function the sigmoidal tanh nonlinearity, which is a typical choice in MLP networks. Other continuous activation functions are possible, too. The sources (latent variables) are assumed to be independent and Gaussian. We have used a mixture of Gaussians model for the sources in [5]. Then it is possible to approximate sufficiently well any non-Gaussian source distribution. However, modeling the sources as mixtures of Gaussians provided a small improvement only in the results while complicating the learning process considerably, and hence the simpler Gaussian model was adopted in this paper.

The parameters of the network are: (1) the weight matrices $\mathbf{A}$ and $\mathbf{B}$ and the vector of biases $\mathbf{a}$; (2) the parameters of the distributions of the noise, source signals and column vectors of the weight matrices; (3) hyperparameters used for defining the distributions of the biases and the parameters in the group (2). All the parameterized distributions are assumed to be Gaussian. This does not limit severely the generality of the approach, but makes computational implementation simpler and much more efficient. The hierarchical description of the distributions of the parameters of the model used here is a standard procedure in probabilistic Bayesian modeling. Its strength lies in that knowledge about equivalent status of different parameters can be easily incorporated. For example all the variances of the noise components have a similar status in the model. This is reflected by the fact that their distributions are assumed to be governed by common hyperparameters.

4 Learning procedure

Usually MLP networks learn the nonlinear input-output mapping in a supervised manner using known input-output pairs, for which the mean-square mapping error is minimized using the back-propagation algorithm [2]. In our case, the inputs are the *unknown* source signals $\mathbf{s}(t)$, and only the outputs of the MLP network, namely the observed data vectors $\mathbf{x}(t)$, are known. Hence, unsupervised learning must be applied. Due to space limitations, we give an overall description of the proposed unsupervised learning procedure in this paper. A more detailed account of the method and discussion of potentially appearing problems can be found in [5].

The practical learning procedure used in all the experiments was the same. First, linear PCA (principal component analysis) is applied to find sensible initial values for the posterior means of the sources. Even though PCA is a linear method, it yields clearly better initial values than a random choice. The posterior variances of the sources are initialized to small values. Good initial values are important for the method because the network can effectively prune away unused parts. Initially the weights of the network have random values, and the network has quite a bad representation for the data. If the sources were adapted from random values, too, the network would consider many of the sources useless for the representation and prune them away. This would lead to a local minimum from which the network might not recover.

Therefore the sources were fixed at the values given by linear PCA for the first 50 sweeps through the entire data set. This allows the network to find a meaningful mapping from sources to the observations, thereby justifying using the sources for the representation. For the same reason, the parameters controlling the distributions of the sources, weights, noise and the hyperparameters are not adapted during the first 100 sweeps. They are adapted only after the network has found sensible values for the variables whose distributions these parameters control.

After this, the learning continued by using the nonlinear model where the sources have Gaussian distributions. This is called nonlinear factor analysis model in [5]. After this phase, the found sources were rotated using an efficient linear ICA algorithm called FastICA [3]. As mentioned earlier, the learning could have then continued using now the mixture-of-Gaussians model for the sources. In [5], that representation is called nonlinear independent factor analysis.

5 Computation of the cost function

In this section, we consider in more detail the Kullback-Leibler cost function C_{KL} defined earlier in Eq. (1). For approximating and then minimizing it, we need two things: the exact formulation of the posterior density $P(S, \boldsymbol{\theta}|X)$ and its parametric approximation $Q(S, \boldsymbol{\theta}|X)$.

According to the Bayes' rule, the posterior pdf of

the unknown variables S and $\boldsymbol{\theta}$ is

$$P(S, \boldsymbol{\theta}|X) = \frac{P(X|S, \boldsymbol{\theta})P(S|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(X)} \qquad (3)$$

The term $P(X|S, \boldsymbol{\theta})$ is obtained from the equation (2). Let us denote the mean of the ith noise component $n_i(t)$ by μ_i and the corresponding variance by by σ_i^2. The distribution $P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})$ is thus Gaussian with mean $\mathbf{b}_i^T\mathbf{f}(\mathbf{A}\mathbf{s} + \mathbf{a}) + \mu_i$ and variance σ_i^2. Here $\mathbf{b}_i^T$ denotes the ith row vector of $\mathbf{B}$. As usually, the noise components $n_i(t)$ are assumed to be independent, and therefore $P(X|S, \boldsymbol{\theta}) = \prod_{t,i} P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})$.

The terms $P(S|\boldsymbol{\theta})$ and $P(\boldsymbol{\theta})$ in (3) are also products of simple Gaussian distributions, and they are obtained directly from the definition of the model structure [5]. The term $P(X)$ does not depend on the model parameters and can be neglected.

The approximation $Q(S, \boldsymbol{\theta}|X)$ must be simple for mathematical tractability and computational efficiency. First, we assume that the source signals S are independent of the other parameters $\boldsymbol{\theta}$, so that $Q(S, \boldsymbol{\theta}|X)$ decouples into $Q(S, \boldsymbol{\theta}|X) = Q(S|X)Q(\boldsymbol{\theta}|X)$. For the parameters $\boldsymbol{\theta}$, a Gaussian density with a diagonal covariance matrix is used. This implies that the approximation is a product of independent distributions: $Q(\boldsymbol{\theta}|X) = \prod_i Q_i(\theta_i|X)$. The parameters of each Gaussian component density $Q_i(\theta_i|X)$ are its mean $\bar{\theta}_i$ and variance $\tilde{\theta}_i$. The pdf $Q(S|X)$ is similar.

Both the posterior density $P(S, \boldsymbol{\theta}|X)$ and its approximation $Q(S, \boldsymbol{\theta}|X)$ are products of simple Gaussian terms, which simplifies the cost function (1) considerably: it splits into expectations of many simple terms. The terms of the form $E_Q\{\log Q_i(\theta_i|X)\}$ are negative entropies of Gaussians, having the exact values $-(1 + \log 2\pi\tilde{\theta}_i)/2$. The most difficult terms are of the form $-E_Q\{\log P(x_i(t)|\mathbf{s}(t), \boldsymbol{\theta})\}$. They are approximated by applying second order Taylor series expansions of the nonlinear activation functions as explained in [5]. The remaining terms are expectations of simple Gaussian terms which can be computed as in [6].

The cost function C_{KL} is a function of the posterior means $\bar{\theta}_i$ and variances $\tilde{\theta}_i$ of the source signals and the parameters of the network. This is because instead of finding a point estimate, the joint posterior pdf of the sources and parameters is estimated in ensemble learning. The variances give information about the reliability of the estimates.

Let us denote the two parts of the cost function (1) arising from the denominator and numerator of the logarithm respectively by $C_p = -E_Q\{\log P\}$ and $C_q = E_Q\{\log Q\}$. The variances $\tilde{\theta}_i$ are obtained by differentiating (1) with respect to $\tilde{\theta}_i$ [5]:

$$\frac{\partial C_{\mathrm{KL}}}{\partial \tilde{\theta}_i} = \frac{\partial C_p}{\partial \tilde{\theta}_i} + \frac{\partial C_q}{\partial \tilde{\theta}_i} = \frac{\partial C_p}{\partial \tilde{\theta}_i} - \frac{1}{2\tilde{\theta}_i} \qquad (4)$$

Equating this to zero yields a fixed-point iteration for updating the variances:

$$\tilde{\theta}_i = \left[2\frac{\partial C_p}{\partial \tilde{\theta}_i}\right]^{-1} \qquad (5)$$

The means $\bar{\theta}_i$ can be estimated from the approximate Newton iteration [5]

$$\bar{\theta}_i \leftarrow \bar{\theta}_i - \frac{\partial C_p}{\partial \bar{\theta}_i}\left[\frac{\partial^2 C_p}{\partial \bar{\theta}_i^{\,2}}\right]^{-1} \approx \bar{\theta}_i - \frac{\partial C_p}{\partial \bar{\theta}_i}\tilde{\theta}_i \qquad (6)$$

6 Experimental results

In all our experiments, the total number of sweeps was 7500, where one sweep means going through all the observations once. A nonlinear factor analysis representation using plain Gaussians as model distributions for the sources was estimated first, and the final results were then obtained by applying linear ICA to the results.

The data set consisted of spectrograms of 24 individual words of Finnish speech, spoken by 20 different speakers. The spectrum was 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.

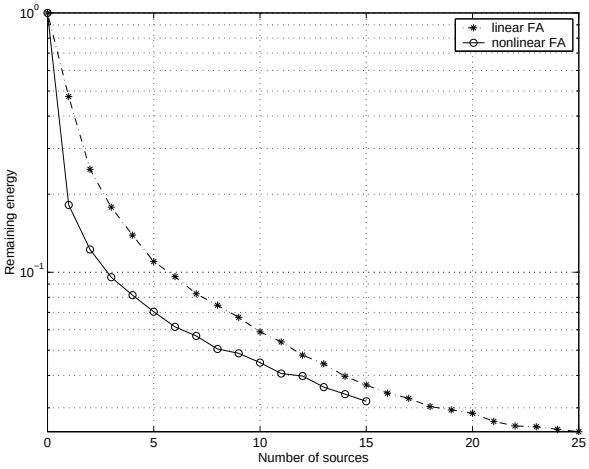

Figure 1: The remaining energy of the speech data as a function of the number of extracted components using linear and nonlinear factor analysis.

For studying the dimensionality of the data, linear factor analysis was applied to the data. The results are shown in Fig. 1. The figure shows also

the results with nonlinear factor analysis. All the results were obtained by using an MLP network with 30 hidden neurons. The data are clearly nonlinear, because nonlinear factor analysis is able to explain it equally well with fewer components than linear factor analysis. The difference is especially clear when the number of components is relatively small.

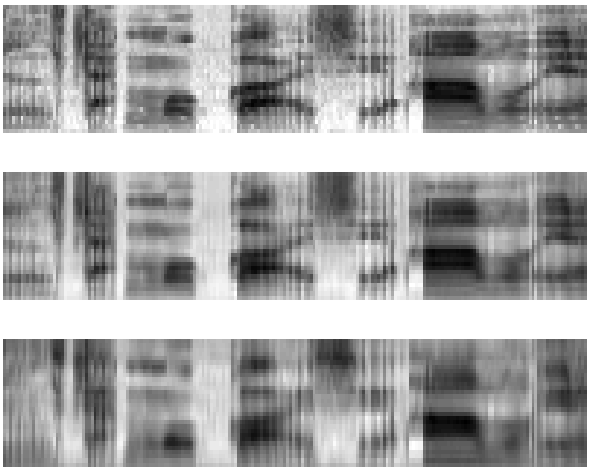

Figure 2: A short fragment of the data used in the speech modeling experiment. The first subfigure shows the original data, the second shows the reconstruction from 8 nonlinear components and the last shows the reconstruction from 8 linear components.

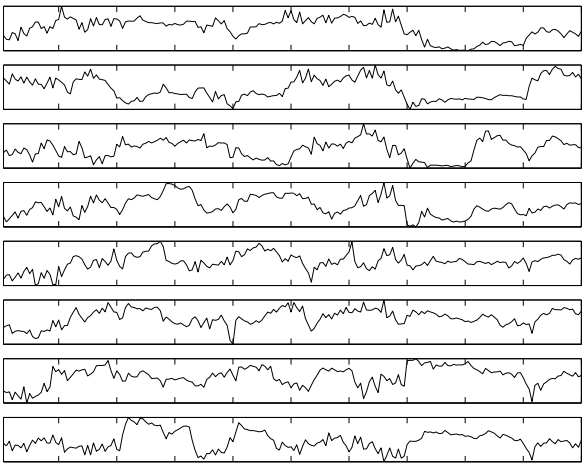

Figure 3: Extracted sources corresponding to the data fragment in Fig. 2.

A small segment of the original data and its reconstructions with eight nonlinear and linear components are shown in Fig. 2. The reconstructed spectrograms are somewhat smoother than the original one. Still, all the discriminative features of the original spectrum are well preserved in the nonlinear reconstruction. The linear reconstruction is not as good, especially at the beginning. The extracted nonlinear sources are shown in Fig. 3.

The sources found by the algorithm could be used as features for a speech recognition system. Since the essential contents of the data can be represented with fewer components than with linear methods, nonlinear ICA should provide better performance for feature extraction. The proposed method ignores all temporal information in the data, but it could easily be extended to take that into account. This would probably lead to better results with the speech data.

7 Conclusions

We have presented a fully Bayesian approach based on ensemble learning for solving the difficult nonlinear blind source separation problem. The MLP network used suits well for modeling both mildly and strongly nonlinear mappings. The presented unsupervised ensemble learning method tries to find the sources and the mapping that have most probably generated the observed data. We believe that this provides an appropriate regularization for the nonlinear source separation problem. The results with real-world speech data are encouraging. The proposed approach allows nonlinear source separation for larger-scale problems than previously proposed nonlinear ICA or BSS approaches [3] which typically suffer from a high computational load, and it can be easily extended in various directions.

References

[1] H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

[2] S. Haykin. *Neural Networks – A Comprehensive Foundation, 2nd ed.* Prentice-Hall, 1998.

[3] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* J. Wiley, 2001.

[4] A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

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

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

[7] D. MacKay. Developments in probabilistic modelling with neural networks—ensemble learning. In *Proc. of the 3rd Annual Symposium on Neural Networks*, pages 191–198, Berlin, 1995. Springer.