1

Nonlinear Independent Component Analysis

J. Karhunen, Helsinki University of Technology, Finland

Abstract

This chapter deals with independent component analysis (ICA) and blind source separation (BSS) for nonlinear data models. A fundamental difficulty especially in the nonlinear ICA problem is that it is highly non-unique without a suitable regularization. After considering this, two methods for solving the nonlinear ICA and BSS problems are presented in more detail. The first one is a maximum likelihood method based on a modified generative topographic mapping. The second approach applies Bayesian ensemble learning to a flexible multilayer perceptron model for finding the sources and nonlinear mixing mapping that have most probably given rise to the observed mixed data. Finally, other techniques introduced for the nonlinear ICA and BSS problems are briefly reviewed.

1.1 Introduction

Independent Component Analysis (ICA) (Lee (1998); Oja et al. (1997); Girolami (1999)) is a statistical technique which tries to represent the observed data in terms of statistically independent component variables. ICA is closely related to the blind source separation (BSS) problem (Cardoso (1998); Amari (1998); Lee (1998); Oja et al. (1997); Girolami (1999)), where the general goal is to separate mutually independent but otherwise unknown source signals from their observed mixtures without knowing the mixing process.

The basic linear data model used in ICA and BSS has the following form. Denote by $\mathbf{x}(t) = [x_1(t), \ldots, x_N(t)]^T$ the N-dimensional t:th observation or data vector. The vectors $\mathbf{x}(t)$ have a common unknown

zero-mean non-Gaussian statistical distribution. In basic ICA and BSS, one tries to fit to the data $\mathbf{x}(t)$ the linear expansion

$$\mathbf{x}(t) = A\mathbf{s}(t) = \sum_{j=1}^{M} s_j(t)\mathbf{e}_j \tag{1.1}$$

Here the source vector $\mathbf{s}(t) = [s_1(t), \ldots, s_M(t)]^T$ contains the M source signals or independent components $s_j(t)$ for the data vector $\mathbf{x}(t)$. $A = [\mathbf{e}_1, \ldots, \mathbf{e}_M]$ is a constant full-rank $N \times M$ matrix, called the mixing matrix. The vectors $\mathbf{e}_j$, $j = 1, \ldots, M$, are the basis vectors of ICA; see (Lee (1998); Oja et al. (1997)). In (1.1), the number of independent components or sources M is usually assumed to be at most equal to the number of mixtures N, and often $M = N$.

The independent components and source signals $s_j(t)$ are determined by finding an $M \times N$ inverse mapping (separating matrix) W so that the M-vector

$$\mathbf{a}(t) = W\mathbf{x}(t) \tag{1.2}$$

becomes an estimate $\mathbf{a}(t) = \hat{\mathbf{s}}(t)$ of the source vector $\mathbf{s}(t)$. To this end, several efficient algorithms have been proposed; see for example (Lee (1998); Oja et al. (1997); Girolami (1999); Amari (1998)) for further information and reviews. The estimate of the mixing matrix A can be computed as the pseudoinverse of W. Because of the blind nature of the linear ICA/BSS problem, the estimated sources or independent components can appear in any order in $\mathbf{a}(t)$, and their scaling and sign can be chosen arbitrarily. Usually the estimated sources are scaled so that their variance equals to unity.

The basic linear model (1.1) is often too simple for describing the observed data $\mathbf{x}(t)$ adequately. A natural extension is to consider the respective ICA or BSS problems for nonlinear data models. For instantaneous mixtures, the nonlinear data model has the general form

$$\mathbf{x}(t) = \mathbf{f}(\mathbf{s}(t)) \tag{1.3}$$

where $\mathbf{f}$ is an unknown real-valued N-component vector function. The ICA problem is now to estimate the realizations of the independent components $s_i(t)$ using only observations of the mixtures $x_j(t)$. In its full generality this nonlinear generalization of ICA is intractable, since the indeterminacies in the separating solutions are much more severe than in the linear case. This is discussed in more detail in the next section.

1.2 Nonlinear ICA and BSS

Assume now for simplicity that the number of independent components or source signals M equals to the number of mixtures N. The nonlinear ICA problem now consists of finding an inverse mapping $\mathbf{h} : R^N \to R^N$ which gives estimates of the independent components as

$$\mathbf{a}(t) = \mathbf{h}(\mathbf{x}(t)) \tag{1.4}$$

This important problem has drawn increasingly attention lately, see for example the papers (Hyvärinen (1999); Parra, Deco & Miesbach (1996); Pajunen & Karhunen (1997); Lin, Grier & Cowan (1997); Yang, Amari & Cichocki (1998); Taleb & Jutten (1999); Lee (1998); Cichocki et al. (1999)) and the references therein. However, nonlinear ICA is often difficult to apply for reasons discussed below.

First, solution of the nonlinear ICA problem is usually highly non-unique (Hyvärinen (1999)). Basically, this results from the fact that if x and y are two independent random variables, any their functions $f(x)$ and $g(y)$ are also independent. Thus if some ICA solution is found for the data model (1.3), it may be quite different from the solution of the respective nonlinear BSS problem, where one should find the original sources $\mathbf{s}(t)$.

It is worth emphasizing that while the solutions of ICA and BSS problems coincide for the basic linear model (1.1), they are clearly different for the nonlinear data model (1.3). Finding the true source signals $\mathbf{s}(t)$ that have generated the observed data is usually a clearly more meaningful and also more unique problem than the nonlinear ICA problem. However, solving it is not easy, and generally requires additional prior information or suitable regularizing constraints.

The question of existence and uniqueness of solutions for nonlinear independent component analysis has been addressed in (Hyvärinen (1999)). The authors show that there always exists an infinity of solutions if the space of the nonlinear mixing functions $\mathbf{f}$ is not limited. They also present a method for constructing parameterized families of nonlinear ICA solutions. A unique solution (up to a rotation) can be obtained in the two-dimensional special case if the mixing mapping $\mathbf{f}$ is constrained to be a conformal mapping together with some other assumptions; see (Hyvärinen (1999)) for details.

Several authors have recently studied separation of so-called *post-nonlinear* mixtures, see for example (Lee (1998); Yang, Amari & Cichocki (1998); Taleb & Jutten (1999)). They are a special case of the

general nonlinear mixing model (1.3) where each mixture has the form

$$x_i(t) = f_i\left(\sum_{j=1}^{N} a_{ij} s_j(t)\right), \qquad i = 1, \ldots, N \tag{1.5}$$

Here a_{ij} is the element on the i-th row and j-the column of the mixing matrix A. Thus the sources $s_j(t)$, $j = 1, \ldots, N$, are first mixed linearly according to the basic ICA/BSS model (1.1), but after that a nonlinear function f_i is applied to them to get the final observations $x_i(t)$. It can be shown (Taleb & Jutten (1999)) that for the post-nonlinear mixtures, the indeterminacies in the ICA and BSS problems are the same as for the basic linear instantaneous mixing model (1.1). The post-nonlinearity assumption is useful in signal processing because it can be thought of as a model for a nonlinear sensor distortion. In more general situations, it is a restrictive and somewhat arbitrary constraint.

Lin (Lin (1999)) has recently derived some interesting theoretical results on ICA that are useful in describing the non-uniqueness of the general nonlinear ICA problem, too. Let the matrices H_s and H_x denote the Hessians of the logarithmic probability densities $\log p_s(\mathbf{s})$ and $\log p_x(\mathbf{x})$ of the source vector $\mathbf{s}$ and mixture (data) vector $\mathbf{x}$, respectively. Then for the the basic linear ICA model (1.1) it holds that (Lin (1999))

$$H_s = A^T H_x A \tag{1.6}$$

where A is the mixing matrix. If the components of $\mathbf{s}$ are truly independent, H_s should be a diagonal matrix. Due to the symmetry of the Hessian matrices H_s and H_x, (1.6) imposes $N(N-1)/2$ constraints for the elements of the $N \times N$ matrix A. Thus a constant mixing matrix A can be solved by estimating H_x at two different points, and assuming some values for the diagonal elements of H_s (Lin (1999)).

If the nonlinear mapping (1.3) is twice differentiable, we can approximate it locally at any point by the linear mixing model (1.1). There A is defined by the first order term $\partial \mathbf{f}(\mathbf{s})/\partial \mathbf{s}$ of the Taylor series expansion of $\mathbf{f}(\mathbf{s})$ at the desired point. But now A generally changes from point to point, so that the constraint conditions (1.6) still leave $N(N-1)/2$ degrees of freedom for determining the mixing matrix A (omitting the diagonal elements).

Another difficulty in the general nonlinear BSS (or ICA) methods proposed thus far is that they tend to be computationally rather demanding. Moreover, the computational load usually increases very rapidly with the

dimensionality of the problem, preventing in practice the application of nonlinear BSS methods to high-dimensional data sets.

Nonlinear ICA and BSS methods can be broadly divided into generative approaches and signal transformation approaches (Lappalainen et al. (2000)). In the generative approaches, the goal is to find a specific model which explains how the observations were generated. In our case, this amounts to estimating both the source signals $\mathbf{s}(t)$ and the unknown mixing mapping $\mathbf{f}(\cdot)$ that have generated the observed data $\mathbf{x}(t)$ through the general mapping (1.3). In the signals transformation methods, one tries to estimate the sources directly using the inverse transformation (1.4). In these methods, the number of estimated sources is the same as the number of observed mixtures (Lappalainen et al. (2000)).

In the following, two generative approaches are discussed in more detail. The first one uses a maximum likelihood method assuming a RBF (radial-basis function) type network structure (Haykin (1998); Bishop (1995)) for the generative mapping. The second approach uses a Bayesian ensemble learning method, modeling the generative mixing mapping by a multi-layer perceptron (MLP) network structure.

1.3 A maximum likelihood approach

1.3.1 Background

One of the earliest ideas (Pajunen, Hyvärinen & Karhunen (1996)) for achieving nonlinear ICA was to use the well-known self-organizing map (SOM) (Kohonen (1995)) to that end. The justification is that a SOM roughly defines a uniform density on a rectangular map with suitable choices. The marginal densities in the directions of the sides of the rectangle are then statistically independent, and hence provide a nonlinear ICA solution.

However, when this idea is applied to nonlinear blind source separation, the serious indeterminacies in the nonlinear case can cause problems. If the sources are uniformly distributed, then it can be heuristically justified that the regularization of the nonlinear separating mapping provided by the SOM approximately separates the sources. But when the true sources have non-uniform probability distributions, the separating mapping providing uniform densities inevitably causes distortions, which are in general the more serious the farther the true source densities are from the uniform ones.

A few years ago, Bishop, Svensen and Williams introduced so-called

generative topographic mapping (GTM) method as a principled alternative to SOM. Their method is presented in detail in (Bishop (1998)). In the basic GTM method, mutually similar plain delta functions that are uniformly distributed on a rectangular grid are used to model the discrete uniform density in the space of latent variables (Bishop (1998)), or the joint density of the sources in our case. The mapping from the sources to the observed data, corresponding in our nonlinear BSS problem the nonlinear mixing mapping (1.3), is then modeled using a mixture-of-Gaussians model. The parameters of the mixture-of-Gaussians model, defining the mixing mapping, are then found using a maximum likelihood method realized by the expectation maximization (EM) algorithm (Bishop (1995); Haykin (1998)). It is well-known that any continuous smooth enough mapping can be approximated with arbitrary accuracy using a mixture-of-Gaussians model with sufficiently many Gaussian basis functions (Haykin (1998); Bishop (1995)). This provides roughly stated the theoretical basis of the GTM method.

Using the basic GTM method instead of SOM for nonlinear blind source separation does not yet bring out any notable improvement, because the densities of the sources are still assumed to be uniform. However, it is straightforward to generalize the GTM method to *arbitrary known* source densities (Pajunen & Karhunen (1997)). The advantage of this approach is that we can directly regularize the inverse of the mixing mapping by using the known source densities. The modified GTM method is then used for finding a non-complex mixing mapping. The modified GTM method originally described in (Pajunen & Karhunen (1997)) differs from standard GTM (Bishop (1998)) in that the required joint density of the sources or latent variables is defined as a weighted sum of delta functions. The weighting coefficients are determined by discretizing the known source densities.

1.3.2 Modified GTM algorithm

Only the main points of the GTM algorithm are presented here, with emphasis on the modifications. For a complete description, see (Bishop (1998)). Generally, GTM is inspired by and closely related to the self-organizing map. Its key benefit is a firm theoretical foundation which helps to overcome some of the limitations of SOM. This provides also the basis of generalizing the GTM approach to arbitrary source densities.

The mixture mapping is modeled as a linear combination of basis

functions:

$$\mathbf{x} = \mathbf{f}(\mathbf{s}; V) = VF(\mathbf{s}),\ F = [f_1, f_2, \ldots, f_M]^T \tag{1.7}$$

where f_j's are suitable fixed basis function (for example Gaussians). In GTM, one tries to maximize the likelihood of the data

$$L(V, \beta) = \sum\nolimits_n \log p(\mathbf{x}(n)|V, \beta) = \sum\nolimits_n \log \int p(\mathbf{x}(n)|\mathbf{s}, V, \beta) p(\mathbf{s}) d\mathbf{s} \tag{1.8}$$

where β^{-1} is the variance of $\mathbf{x}$ given $\mathbf{s}$ and V. It is assumed that the density of $\mathbf{s}$ is known, and can be approximated by the product of marginal densities $p(\mathbf{s}) = \prod_i p_i(s_i)$, where each marginal density is a discrete density with corresponding sample points as the vectors $\mathbf{s}_{ij}$ below. For easier notation, only two sources are considered here, yielding $p(\mathbf{s}) = \sum_{i,j} \alpha_{ij} \delta(\mathbf{s} - \mathbf{s}_{ij})$. The vectors $\mathbf{s}_{ij}$ are fixed in the source signal space, and the coefficients $\alpha_{ij} = p_1(i)p_2(j)$ where p_1 and p_2 are the discrete approximations of the marginal densities of $\mathbf{s}$. From (1.8), one gets

$$L(V, \beta) = \sum \log(\sum\nolimits_{i,j} p(\mathbf{x}(n)|\mathbf{s}_{ij}, V, \beta)) \tag{1.9}$$

Setting the derivatives of the likelihood to zero yields the equation allowing the update of the weight matrix V:

$$F^T G F V^T = F^T R X \tag{1.10}$$

where $X = [\mathbf{x}(1), \ldots, \mathbf{x}(K)]^T$, G is a diagonal matrix with elements $g_{ii} = \sum_n R_{ijn}(V, \beta)$, and the elements of R are

$$R_{ijn} = \frac{\alpha_{ij} p(\mathbf{x}(n)|\mathbf{s}_{ij}, V, \beta)}{\sum_{k,l} \alpha_{kl} p(\mathbf{x}(n)|\mathbf{s}_{kl}, V, \beta)} \tag{1.11}$$

Then the parameter β can be updated by

$$\frac{1}{\beta} = \frac{1}{KD} \sum\nolimits_{i,j} \sum\nolimits_n R_{ijn} \|\mathbf{f}(\mathbf{s}_{ij}; V) - \mathbf{x}(n)\|^2 \tag{1.12}$$

In GTM, the well-known EM algorithm is used for maximizing the likelihood. Here the E-step (1.11) consists of computing R_{ijn}, and the M-steps (1.10), (1.12) of updating V and β. The above derivation is quite similar as in the original GTM method (Bishop (1998)), only the prior density coefficients α_{ij} have been added to the model.

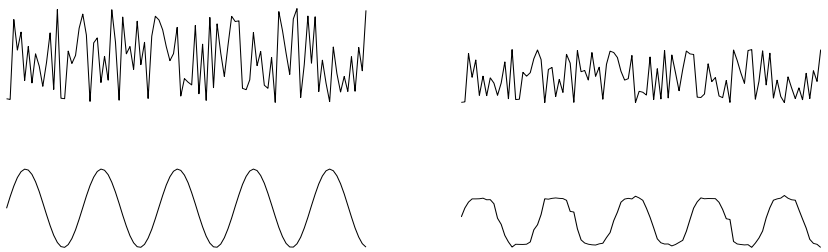

Fig. 1.1. ***A nonlinear source separation example.*** *Left: The original source signals. Right: Separated signals.*

1.3.3 An experiment

In the following, a simple experiment involving two sources (Fig. 1.1) and three noisy nonlinear mixtures is described. Linear mixtures of the sources were transformed using a multilayer perceptron network with a volume conserving architecture (see (Deco & Brauer (1995))). Such an architecture was chosen for ensuring that the total mixing mapping is bijective and therefore reversible. Also very complex distortions of the source densities are avoided. This choice also makes the total mixing mapping more complex than the post-nonlinear model used for example in (Taleb & Jutten (1999)) and (Yang, Amari & Cichocki (1998)). Finally, additive Gaussian noise was added.

The mixture can be written as

$$\mathbf{x} = A\mathbf{s} + \tanh(UA\mathbf{s}) + \mathbf{n} \tag{1.13}$$

where U is an upper-diagonal matrix with zero diagonal elements, and $\mathbf{n}$ denotes the noise vector. The nonzero elements of U were drawn from a standard Gaussian distribution. The matrix U ensures volume conservation of the nonlinearity applied to $A\mathbf{s}$. The two-dimensional joint densities of the mixtures are shown in Fig. 1.2. They clearly reveal the nonlinearity of the mixing mapping.

The algorithm presented above was used to learn a separating mapping. For reducing scaling effects, the mixtures were first whitened. After whitening the mixtures are uncorrelated and have an unit variance. Then the modified GTM algorithm was run for eight iterations using a 5×5 map. The separated sources are compared to the original sources in Fig. 1.1. The waveforms of the original sources are approximately recovered, even though there is some inevitable distortion due to the noise, discretization, and the difficulty of the problem. Also the

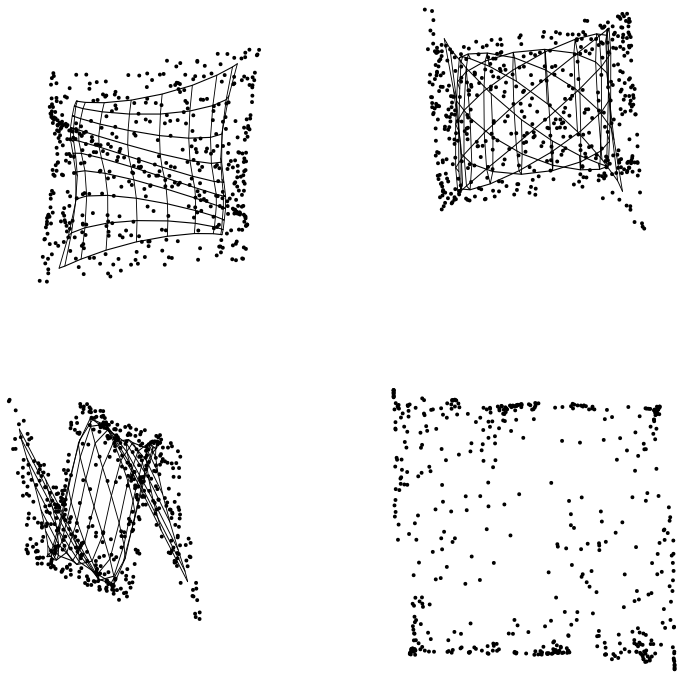

Fig. 1.2. ***Joint mixture densities with superimposed maps.*** *Top left:* $p(x_1, x_2)$. *Top right:* $p(x_1, x_3)$. *Bottom left:* $p(x_2, x_3)$. *Bottom right: output signal joint density.*

joint density of the separated sources shown in Fig. 1.2 indicates that a factorizable density has been approximately obtained. The superimposed maps were obtained by mapping a 10×10 grid of vectors $\mathbf{s}$ to the mixture space.

1.4 An ensemble learning approach to nonlinear BSS

1.4.1 Ensemble learning

In this section, we present a promising new generative approach for nonlinear blind source separation or independent component analysis. Here the nonlinear mapping (1.3) from the unknown sources $\mathbf{s}(t)$ to the known observations $\mathbf{x}(t)$ is modeled using the familiar multi-layer perceptron (MLP) network structure (Haykin (1998); Bishop (1995)). However, the learning procedure is based on unsupervised Bayesian ensemble learn-

ing. It is quite different from standard backpropagation learning which minimizes the mean-square representation error in MLP networks in a supervised manner (Haykin (1998); Bishop (1995).)

A flexible model family, such as MLP networks, provides infinitely many possible explanations of different complexity to the observed data. Choosing too complex a model results in overfitting, where the model tries to make up meaningless explanations for the noise in addition to the true sources or independent components. Choosing too simple a model results in underfitting, leaving hidden some of the true sources that have generated the data.

An appropriate solution to this problem is that no single model should actually be chosen. Instead, all the possible explanations should be taken into account and weighted according to their posterior probabilities. This approach, known as Bayesian learning (Bishop (1995)), 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. The posterior pdfs of too simple models are low, because they leave a lot of the data unexplained. Also too complex models occupy little probability mass, even though they often show a high but very narrow peak in their posterior pdfs corresponding to the overfitted parameters.

In practice, exact treatment of the posterior pdfs of the models is impossible. Therefore, some suitable approximation method must be used. Ensemble learning (Barber & Bishop (1998); Lappalainen & Miskin (2000)), also known as 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 $X = \{\mathbf{x}(t)|t\}$ and $S = \{\mathbf{s}(t)|t\}$, and let $\boldsymbol{\theta}$ denote all the unknown parameters of the model. Furthermore, let $P(\boldsymbol{\theta}|X)$ denote the exact posterior pdf and $Q(\boldsymbol{\theta})$ its parametric approximation. The misfit is measured with the Kullback-Leibler (KL) divergence C_{KL} between P and Q, defined by the cost function

$$C_{\mathrm{KL}} = E_Q \left\{ \log \frac{Q}{P} \right\} = \int d\boldsymbol{\theta} Q(\boldsymbol{\theta}) \log \frac{Q(\boldsymbol{\theta})}{P(\boldsymbol{\theta}|X)} \tag{1.14}$$

Because the KL divergence involves an expectation over a distribution, it is sensitive to probability mass rather than to probability density.

1.4.2 Model structure

In this approach, MLP networks are used for modeling the nonlinear mixing mapping $\mathbf{f}(\cdot)$. They have the universal approximation property (Haykin (1998)) for smooth continuous mappings, and suit well to modeling both strongly and mildly nonlinear mappings.

The data model used in this approach is as follows (Lappalainen & Honkela (2000); Lappalainen (2000)). 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 B and A contain the weights of the output and the hidden layer of the network, respectively, and $\mathbf{b}$ is the bias vector of the hidden layer. The vector of nonlinear activation functions is denoted by $\mathbf{g}(\cdot)$, and $\mathbf{n}(t)$ is a Gaussian noise vector corrupting the observations. It can generally have a nonzero mean vector here. Using these notations, the data model is

$$\mathbf{x}(t) = B\left[\mathbf{g}\left(A\mathbf{s}(t) + \mathbf{b}\right)\right] + \mathbf{n}(t) \tag{1.15}$$

As the activation function, the hyperbolic tangent nonlinearity $g(y) = \tanh(y)$ is used, which is a typical choice in MLP networks. Other continuous activation functions could be used, too. The sources are assumed to be independent, and they are modeled by mixtures of Gaussians. The independence assumption is natural, because the goal of the model is to find the underlying independent components of the observations. By using mixtures of Gaussians, one can model sufficiently well any non-Gaussian distribution of the sources (Bishop (1995)). This type of representation has earlier been successfully applied to the standard linear ICA model in (Lappalainen (1999)).

The parameters of the network are: (1) the weight matrices A and B and the vector of biases $\mathbf{b}$; (2) the parameters of the distributions of the noise, source signals and column vectors of the weight matrices; (3) the hyperparameters used for defining the distributions of the biases and the parameters in the group (2). For a more detailed description, see (Lappalainen & Honkela (2000); Lappalainen (2000)). All the parameterized distributions are assumed to be Gaussian except for the sources, which are modeled as mixtures of Gaussians. This does not limit the generality of the approach severely, 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.

1.4.3 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 or some alternative method (Haykin (1998); Bishop (1995)). 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 only in this paper. A more detailed account of the method and discussion of potentially appearing problems can be found in (Lappalainen & Honkela (2000); Lappalainen (2000)).

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.

Furthermore, we first used a simpler nonlinear model where the sources had Gaussian distributions instead of mixtures of Gaussians. This is called *nonlinear factor analysis* model in the following. After this phase, the sources were rotated using an efficient linear ICA algorithm called FastICA (Hyvärinen et al. (1998)). The rotation of the sources was compensated by an inverse rotation of the weight matrix A of the hidden layer. The final representation of the data was then found by continuing learning using now the mixture-of-Gaussians model for the sources. In (Lappalainen & Honkela (2000)), this representation is called *nonlinear independent factor analysis*.

1.4.4 Computation of the cost function

In this subsection, the Kullback-Leibler cost function C_{KL} defined earlier in Eq. (1.14) in considered in more detail. For approximating and then minimizing it, we need two things: the exact formulation of the posterior density $P(\boldsymbol{\theta}|X)$ and its parametric approximation $Q(\boldsymbol{\theta})$.

According to the Bayes' rule, the posterior pdf of the unknown variables S and $\boldsymbol{\theta}$ is

$$P(S,\boldsymbol{\theta}|X) = \frac{P(X|S,\boldsymbol{\theta})P(S|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(X)} \tag{1.16}$$

The term $P(X|S,\boldsymbol{\theta})$ is obtained from the equation (1.15). Let us denote the mean vector of $\mathbf{n}(t)$ by $\boldsymbol{\mu}$ and the variance vector of the noise term $\mathbf{n}(t)$ by $\boldsymbol{\sigma^2}$. The distribution $P(x_i(t)|\mathbf{s}(t),\boldsymbol{\theta})$ is thus Gaussian with mean $\mathbf{b}_i^T\mathbf{g}(A\mathbf{s}+\mathbf{b}) + \mu_i$ 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 (1.16) are also products of simple Gaussian distributions, and they are obtained directly from the definition of the model structure (Lappalainen & Honkela (2000); Lappalainen (2000)). The term $P(X)$ does not depend on the model parameters and can be neglected.

The approximation $Q(S,\boldsymbol{\theta})$ must be simple for mathematical tractability and computational efficiency. We assume that it is a Gaussian density with a diagonal covariance matrix. This implies that the approximation is a product of independent distributions: $Q(\boldsymbol{\theta}) = \prod_i Q_i(\theta_i)$. The parameters of each Gaussian component density $Q_i(\theta_i)$ are its mean $\bar{\theta}_i$ and variance $\tilde{\theta}_i$.

Both the posterior density $P(\boldsymbol{\theta}|X)$ and its approximation $Q(\boldsymbol{\theta})$ are

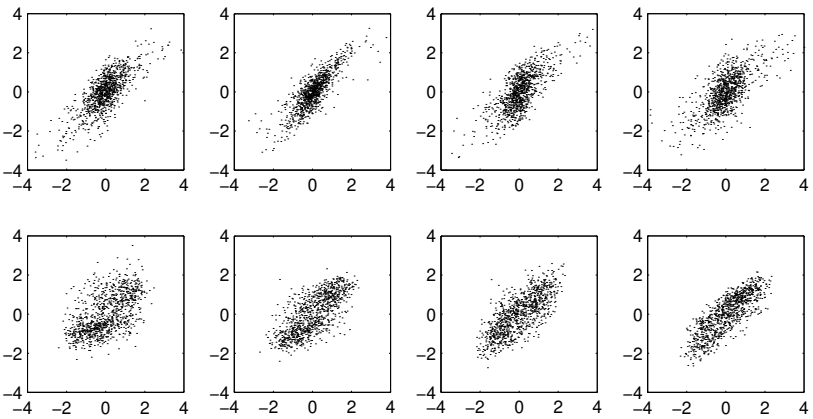

Fig. 1.3. ***Separation results given by linear ICA.*** *Original sources are on the x-axis of each scatter plot and the sources estimated by a linear ICA are on the y-axis. Signal to noise ratio is 0.7 dB.*

products of simple Gaussian terms, which simplifies the cost function (1.14) considerably: it splits into expectations of many simple terms. The terms of the form $E_Q\{\log Q_i(\theta_i)\}$ 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's series expansions of the nonlinear activation functions as explained in (Lappalainen & Honkela (2000); Lappalainen (2000)). The remaining terms are expectations of simple Gaussian terms which can be computed as in (Lappalainen (1999)).

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 (1.14) by $C_p = -E_Q\{\log P\}$ and $C_q = E_Q\{\log Q\}$. The variances $\tilde{\theta}_i$ are obtained by differentiating (1.14) with respect to $\tilde{\theta}_i$ (Lappalainen & Honkela (2000); Lappalainen (2000)):

$$\frac{\partial C}{\partial \tilde{\theta}} = \frac{\partial C_p}{\partial \tilde{\theta}} + \frac{\partial C_q}{\partial \tilde{\theta}} = \frac{\partial C_p}{\partial \tilde{\theta}} - \frac{1}{2\tilde{\theta}} \tag{1.17}$$

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

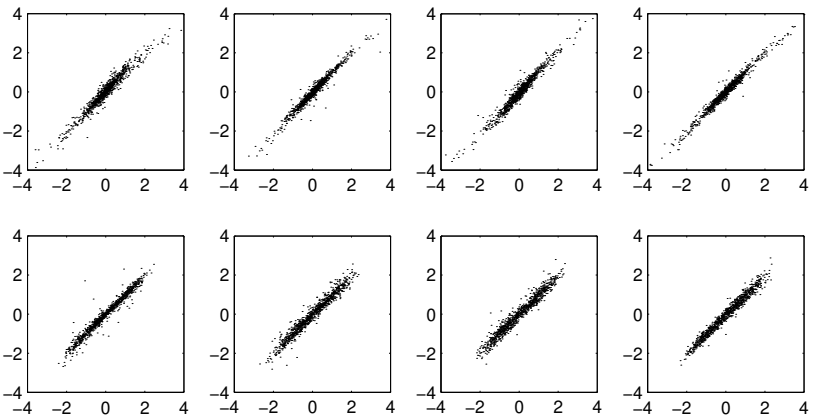

Fig. 1.4. ***Separation results given by nonlinear factor analysis.*** *Scatter plots of the sources after 2000 sweeps of nonlinear factor analysis followed by a rotation with a linear ICA. Signal-to-noise ratio is 13.2 dB.*

ances:

$$\tilde{\theta} = \left[2\frac{\partial C_p}{\partial \tilde{\theta}}\right]^{-1} \tag{1.18}$$

The means $\bar{\theta}_i$ can be estimated from the approximate Newton iteration (Lappalainen & Honkela (2000); Lappalainen (2000))

$$\bar{\theta} \leftarrow \bar{\theta} - \frac{\partial C_p}{\partial \bar{\theta}} \left[\frac{\partial^2 C}{\partial \bar{\theta}^2}\right]^{-1} \approx \bar{\theta} - \frac{\partial C_p}{\partial \bar{\theta}} \tilde{\theta} \tag{1.19}$$

1.4.5 Experimental results

In all the simulations, the total number of sweeps was 7500, where one sweep means going through all the observations $\mathbf{x}(t)$ once. As explained before, a nonlinear factor analysis (or nonlinear PCA subspace) representation using plain Gaussians as model distributions for the sources was estimated first. In the experiments, 2000 first sweeps were used for finding this intermediate representation. After a linear ICA rotation, the final mixture-of-Gaussians representation of the sources was then estimated during the remaining 5500 sweeps. In the following, experiments with artificially generated nonlinear data are first presented, followed by separation results on real-world process data.

In the first experiment, there were eight sources, four sub-Gaussian

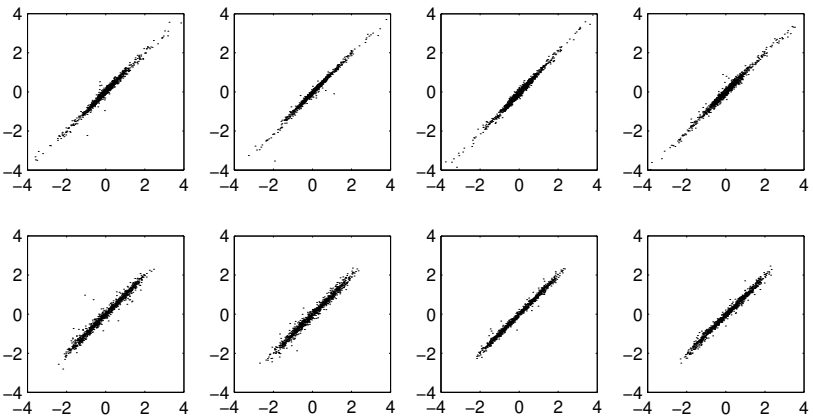

Fig. 1.5. ***Separation results given by nonlinear ICA.*** *The final results after using a mixture-of-Gaussians model for the sources for the last 5500 sweeps. Signal to noise ratio is 17.3 dB.*

and four super-Gaussian ones. The data were generated from these sources through a nonlinear mapping, which was obtained by using a randomly initialized MLP network having 30 hidden neurons and 20 output neurons. Gaussian noise having a standard deviation of 0.1 was added to the data. The nonlinearity used in the hidden neurons was chosen to be the inverse hyperbolic sine $\sinh^{-1}(t)$, which means that the nonlinear source separation algorithm using the MLP network with $\tanh(t)$ nonlinearities cannot use exactly the same weights.

Several different numbers of hidden neurons were tested in order to optimize the structure of the MLP network, but the number of sources was assumed to be known. This assumption is reasonable because it is possible to optimize the number of sources simply by minimizing the cost function as experiments with purely Gaussian sources have shown (Lappalainen & Honkela (2000); Lappalainen et al. (2000)). The network which minimized the cost function turned out to have 50 hidden neurons. The number of Gaussians in each of the mixtures modeling the distribution of each source was chosen to be three, and no attempt was made to optimize this number.

The results are depicted in Figs. 1.3, 1.4, and 1.5. Each figure shows eight scatter plots, corresponding to each of the eight original sources. The original source which was used for generating the data appears on the x-axis, and the respective estimated source in on the y-axis of each

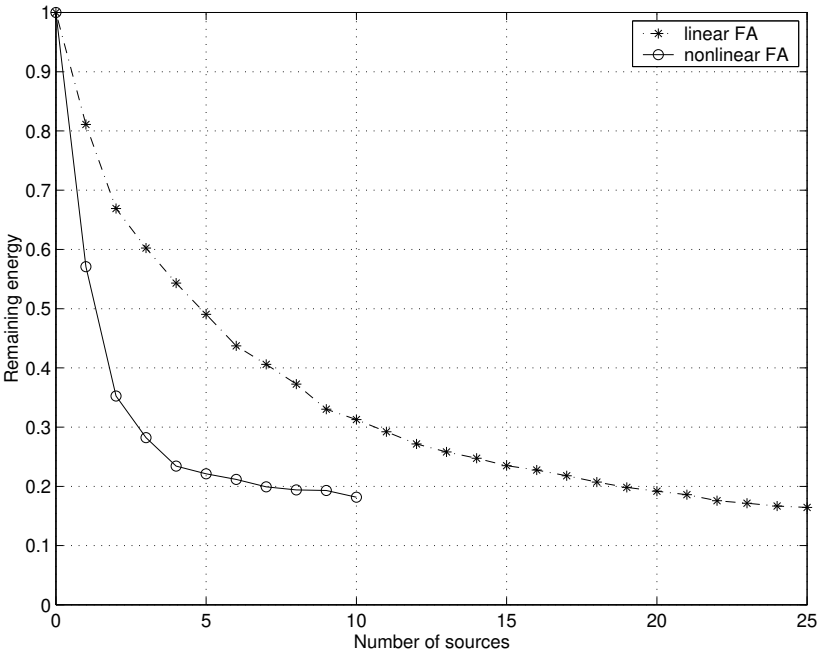

Fig. 1.6. ***A study of the intrinsic dimensionality of the data.*** *The figure shows the remaining energy of the process data as a function of the number of extracted components using linear and nonlinear factor analysis.*

plot. Each point corresponds to one data vector. The upper plots of each figure correspond to the super-Gaussian and the lower plots to the sub-Gaussian sources. Optimal result is a straight line implying that the estimated values of the sources coincide with the true values.

Figure 1.3 shows the result of a linear fixed-point (FastICA) algorithm (Hyvärinen et al. (1998)). The linear ICA is able to retrieve the original sources with only 0.7 dB SNR (signal-to-noise ratio). In practice a linear method could not deduce the number of sources, and the result would be even worse. The poor signal-to-noise ratio shows that the data really lies in a nonlinear subspace. Figure 1.4 depicts the results after 2000 sweeps with Gaussian sources (nonlinear factor analysis) followed by a rotation with linear FastICA. Now the SNR is considerably better, 13.2 dB, and the sources have clearly been retrieved. Figure 1.5 shows the final results after another 5500 sweeps when the mixture-of-Gaussians model have been used for the sources. The SNR has further improved to 17.3 dB.

Another data set consisted of 30 time series of length 2480 measured

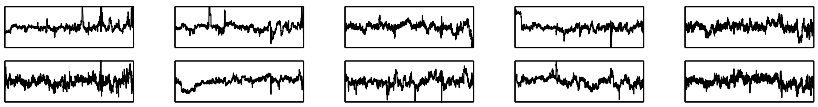

Fig. 1.7. ***Industrial pulp process:*** *The ten estimated sources. Time increases from left to right.*

using different sensors from an industrial pulp process. A human expert preprocessed the measured signals by roughly compensating for the time lags of the process originating from the finite speed of pulp flow through the process.

For studying the intrinsic dimensionality of the data, linear factor analysis was applied to the data. The result is shown in Fig. 1.6. The figure shows also the results with nonlinear factor analysis. It is obvious that the data are quite nonlinear, because nonlinear factor analysis is able to explain the data with 10 components equally well as linear factor analysis with 21 components.

Different numbers of hidden neurons and sources were tested with random initializations using Gaussian model for sources (nonlinear factor analysis). It turned out that the cost function was minimized for a network of 10 sources and 30 hidden neurons. The same network size was chosen for nonlinear blind source separation based on the mixture-of-Gaussians model for the sources. After 2000 sweeps using the nonlinear factor analysis model the sources were rotated with FastICA, and each source was modeled with a mixture of three Gaussian distributions. The resulting sources are shown in Fig. 1.7.

Figure 1.8 shows 30 subimages, each corresponding to a specific measurement made from the process. The original measurement appears as the upper time series in each subimage, and the lower time series is the reconstruction of this measurement given by the network. These reconstructions are the posterior means of the outputs of the network when the inputs were the estimated sources shown in Fig. 1.7. The original measurements show a great variability, but the reconstructions are strikingly accurate. In some cases it even seems that the reconstruction is less noisy than the original signal.

The experiments suggest that the estimated source signals can have meaningful physical interpretations. The results are encouraging, but further studies are needed to verify the interpretations of the signals.

The proposed ensemble learning method for nonlinear blind source

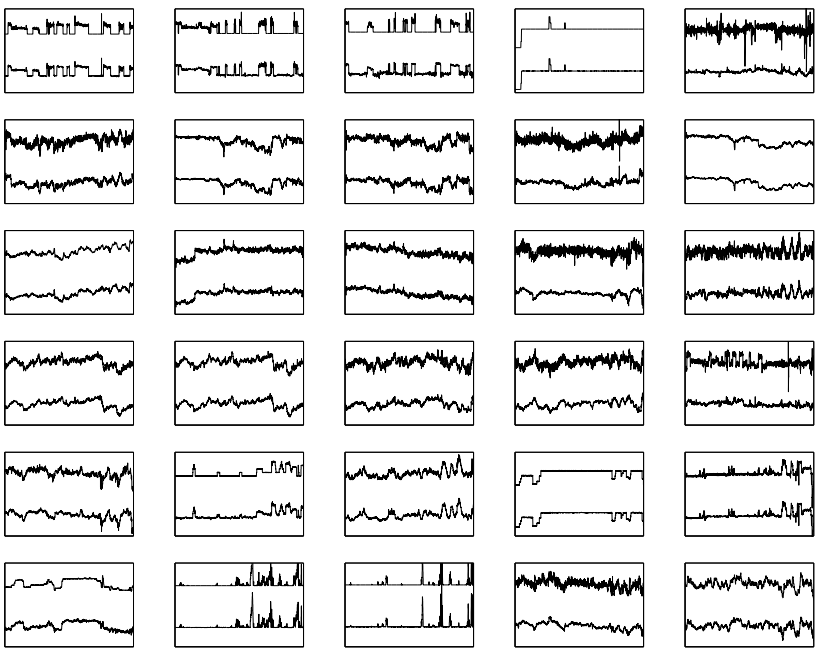

Fig. 1.8. ***Industrial pulp process:*** *The 30 original time series are shown on each plot on top of the reconstruction made from the sources shown in Fig. 1.7*

separation can be extended in several ways. An obvious extension is inclusion of time delays into the data model, making use of the temporal information often present in the sources. This would probably help in describing the process data even better. Using the Bayesian framework, it is also easy to treat missing observations or only partly known inputs.

1.5 Other approaches

Burel (Burel (1992)) was probably the first to introduce an algorithm for nonlinear blind source separation. His method is based on back-propagation type neural learning for parametric nonlinearities, and suffers from computational complexity and problems with local minima. In a series of papers (Deco & Brauer (1995); Parra, Deco & Miesbach (1995); Parra, Deco & Miesbach (1996)), Deco and Parra with their co-authors developed methods based on volume conserving symplectic transformations for nonlinear independent component analysis. The

constraint of volume conservation is, however, somewhat arbitrary, and so these methods are usually not able to recover the original sources. Also their computational load tends to be high.

The idea of using self-organizing map (SOM) (Kohonen (1995)) for the general nonlinear BSS problem was introduced in (Pajunen, Hyvärinen & Karhunen (1996)). However, this approach is limited mainly to separation of sources having distributions not too far from the uniform distribution. The nonlinear mixing process is considered to be regular: it is required that the inverse of the mixing mapping should be the least complex mapping which yields independent components (Pajunen, Hyvärinen & Karhunen (1996)). Also Lin, Grier, and Cowan (Lin, Grier & Cowan (1997)) independently proposed using SOM for nonlinear blind source separation in a different manner.

The ensemble learning approach to nonlinear BSS, discussed in more detail earlier in this chapter, is based on using multi-layer perceptron (MLP) networks as a flexible model for the nonlinear mixing mapping (1.3). Several authors have used autoassociative MLP networks for learning similar type mappings. Both the generative model and its inversion are learned simultaneously, but separately without utilizing the fact that the models are connected. Autoassociative MLPs have shown some success in nonlinear data representation (Haykin (1998)), but generally they suffer from slow learning prone to local minima.

Most works on autoassociative MLPs use point estimates for weights and sources obtained from minimization of the mean-square representation error for the data. It is then impossible to reliably choose the structure of the model, and problems with over- or underfitting can be severe. Hecht-Nielsen (Hecht-Nielsen (1995); Hecht-Nielsen (1996)) proposed using so-called replicator networks for universal optimal nonlinear coding of input data. These networks are autoassociate MLP networks, where the data vectors are mapped onto a unit hypercube having an uniform distribution. The coordinates of the data on the axes of the hypercube, called natural coordinates, form then in fact a nonlinear ICA solution, even though this was not noted in the original papers (Hecht-Nielsen (1995); Hecht-Nielsen (1996)).

Hochreiter and Schmidhuber (Hochreiter & Schmidhuber (1999)) have used in context with MLP networks a minimum description length based method which does estimate the distribution of the weights, but has no model for the sources. It is then impossible to measure the description length of the sources. Anyway, their method shows interesting connections with ICA; sometimes it provides a nonlinear ICA solution,

sometimes not (Hochreiter & Schmidhuber (1999)). Another general approach proposed recently for solving the nonlinear BSS or ICA problems is based on state-space modeling (Cichocki et al. (1999)).

Various separation methods for the considerably simpler case of the post-nonlinear mixture model (1.5) have been introduced at least in (Taleb & Jutten (1999); Lee (1998); Yang, Amari & Cichocki (1998)). In particular, a MLP model is used for separating sources from their post-nonlinear mixtures in (Yang, Amari & Cichocki (1998)). This method is an extension of the well-known natural gradient method for linear BSS (Lee (1998)).

In the ensemble learning method discussed before, the necessary regularization for nonlinear ICA is achieved by choosing the model and sources that have most probably generated the observed data. A similar generative model as there is applied to the linear ICA/BSS problem in (Attias (1999)). The ensemble learning based method described in this chapter differs from the method introduced in (Attias (1999)) in that it uses a more general nonlinear data model, and applies a fully Bayesian treatment to the hyperparameters of the network or graphical model, too. Connections of the ensemble learning method with other Bayesian approaches are discussed in more detail in (Lappalainen & Honkela (2000); Lappalainen et al. (2000)).

Nonlinear blind source separation and independent component analysis are generally difficult problems both computationally and conceptually. Therefore, local linear ICA/BSS methods have received some attention recently as a practical compromise between linear ICA and completely nonlinear ICA or BSS. These methods are more general than standard linear ICA in that several different linear ICA models are used to the describe the observed data. The local linear ICA models can be either overlapping as in the mixture-of-ICA methods introduced in (Lee, Lewicki & Sejnowski (1999a); Lee, Lewicki & Sejnowski (1999b)) or non-overlapping as in the clustering based methods proposed in (Karhunen & Malaroiu (1999a); Karhunen & Malaroiu (1999b)).

1.6 Conclusions and remarks

In this chapter, generalizations of standard linear ICA and BSS problems to nonlinear data models have been considered. Contrary to the linear case, solutions of the respective ICA and BSS problems usually differ from each other when the data model is nonlinear. Especially the nonlinear ICA problem is ill-posed without some suitable extra con-

straints or regularization, having in general infinitely many qualitatively different solutions.

Solving the nonlinear ICA or BSS problems appropriately using solely the independence assumption of the source signals is usually not possible except for in simple special cases, for example in the case of post-nonlinear mixtures. Otherwise, suitable additional information on the problem is required. This extra information is often provided in the form of regularizing contraints. Various methods proposed for regularizing the nonlinear ICA or BSS problems have been briefly reviewed in the previous section. Another possibility is to have more information about the sources or mixtures themselves. Examples of this type of approaches are methods developed for separating post-nonlinear mixtures, and the maximum likelihood approach which requires knowledge of the probability distributions of the sources.

A large part of this chapter has been devoted to a recently introduced fully Bayesian approach based on ensemble learning for solving the nonlinear BSS (or ICA) problem. This method applies the well-known MLP network, which suits well to modeling both mildly and strongly nonlinear mappings. The proposed unsupervised ensemble learning method tries to find the sources and the mapping that have most probably generated the observed data. It is believed that this principle provides an appropriate regularization for the nonlinear source separation problem. The results are encouraging for both artificial and real-world data. The method allows nonlinear source separation for larger scale problems than some previously proposed computationally quite demanding methods, and it can be easily extended into various directions.

A lot of work remains to be done in developing suitable methods for the nonlinear ICA and BSS problems, and understanding better which constraints are most suitable in each situation. A number of different approaches have been proposed, but no comparisons are yet available for assessing their strentghts and weaknesses.

Acknowledgments

The author wishes to thank H. Lappalainen, P. Pajunen, A. Hyvärinen, X. Giannakopoulos, and A. Honkela for their co-operation, helpful comments, and useful discussions on the topic the chapter.

References

Amari, S.-I. & Cichocki, A. 1998 Adaptive blind signal processing - neural network approaches. *Proc. of the IEEE* **86**, 2026–2048.

Attias, H. 1999 Independent Factor Analysis. *Neural Computation* **11**, 803–851.

Barber, D. & Bishop, C. 1998 Ensemble learning in Bayesian neural networks. In *Neural Networks and Machine Learning* (ed. M. Jordan, M. Kearns & S. Solla). Pp. 215–237. Springer.

Bishop, C. 1995 *Neural Networks for Pattern Recognition.* Clarendon Press.

Bishop, C., Svensen, M. & Williams, C. 1998 GTM: The Generative Topographic Mapping. *Neural Computation* **10** 215–234.

Burel, G. 1992 Blind separation of sources: A nonlinear neural algorithm. *Neural Networks* **5**, 937–947.

Cardoso, J.-F. 1998 Blind signal separation: statistical principles. *Proc. of the IEEE* **86**, 2009–2025.

Cichocki, A., Zhang, L., Choi, S. & Amari, S.-I. 1999 Nonlinear dynamic independent component analysis using state-space and neural network models. In *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99).* Aussois, France, January 1999, pp. 99–104.

Deco, G. & Brauer, W. (1995) Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures. *Neural Networks* **8**, 525–535.

Girolami, M. 1999 *Self-Organising Neural Networks - Independent Component Analysis and Blind Source Separation.* Springer.

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

Hecht-Nielsen, R. 1995 Replicator neural networks for universal optimal source coding. *Science* **269**, 1860–1863.

Hecht-Nielsen, R. 1996 Data manifolds, natural coordinates, replicator neural networks, and optimal source coding. In *Proc. 1996 Int. Conf. on Neural Information Processing (ICONIP'96).* Hong Kong, September 1996, pp. 41–45.

Hochreiter, S. & Schmidhuber, J. 1999 Feature Extraction Through LOCOCODE. *Neural Computation* **11**, 679–714.

Hyvärinen, A. et al. 1998 *The FastICA MATLAB toolbox.* Available at http://www.cis.hut.fi/projects/ica/fastica/. Helsinki Univ. of Technology, Finland, 1998.

Hyvärinen, A. & Pajunen, P. 1999 Nonlinear independent component analysis: existence and uniqueness results. *Neural Networks* **12**, 209–219.

Karhunen, J. & Malaroiu, S. 1999 Local independent component analysis using clustering. In *Proc. of the First Int. Workshop on Independent Component Analysis and Signal Separation.* Aussois, France, January 1999, pp. 43–48.

Karhunen, J. & Malaroiu, S. 1999 Locally linear independent component analysis. In *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN'99).* Washington, DC, July 1999, pp. 882–887.

Kohonen, T. 1995 *Self-Organizing Maps.* Springer.

Lappalainen, H. 1999 Ensemble learning for independent component analysis. In *Proc. of the First Int. Workshop on Independent Component Analysis and Signal Separation.* Aussois, France, January 1999, pp. 7–12.

Lappalainen, H. 2000 Nonlinear independent component analysis using ensemble learning: theory. To appear in *Proc. of the 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation.* Espoo, Finland, June 2000.

Lappalainen, H. Giannakopoulos, X., Honkela, A. & Karhunen, J. 2000 Nonlinear independent component analysis using ensemble learning: experiments and discussion. To appear in *Proc. of the 2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation.* Espoo, Finland, June 2000.

Lappalainen, H. & Honkela, A. 2000 Bayesian nonlinear independent component analysis by multi-layer perceptrons. In *Advances in Independent Component Analysis* (ed. M. Girolami). In press. Springer.

Lappalainen, H. & Miskin, J. 2000 Ensemble learning. In *Advances in Independent Component Analysis* (ed. M. Girolami). In press. Springer.

Lee, T.-W. 1998 *Independent Component Analysis - Theory and Applications.* Kluwer.

Lee, T.-W., Lewicki, M. & Sejnowski, T. 1999 ICA mixture models for unsupervised classification and automatic context switching. In *Proc. of the First Int. Workshop on Independent Component Analysis and Signal Separation.* Aussois, France, January 1999, pp. 209–214.

Lee, T.-W., Lewicki, M. & Sejnowski, T. 1999 Unsupervised classification with non-Gaussian mixture models using ICA. In *Advances in Neural Information Processing Systems 11.* MIT Press, pp. 508–514.

Lin, J. 1999 Factorizing probability density functions: generalizing ICA. In *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99).* Aussois, France, January 1999, pp. 313–318.

Lin, J., Grier, D. & Cowan, J. 1997 Faithful representation of separable distributions. *Neural Computation* **9**, pp. 1305–1320.

Oja, E., Karhunen, J., Hyvärinen, A., Vigario, R. & Hurri, J. 1997 Neural independent component analysis - approaches and applications. In *Brain-Like Computing and Intelligent Information Systems* (ed. S.-I. Amari & N. Kasabov). Pp. 167–188. Springer.

Pajunen, P., Hyvärinen, A. & Karhunen, J. 1996 Nonlinear blind source separation by self-organizing maps. In *Proc. 1996 Int. Conf. on Neural information Processing (ICONIP'96).* Hong Kong, September 1996, pp. 1207–1210.

Pajunen, P. & Karhunen, J. 1997 A maximum likelihood approach to nonlinear blind separation. In *Proc. 1997 Int. Conf. on Artificial Neural Networks.* Lausanne, Switzerland, October 1997, pp. 541–546.

Parra, L., Deco, G. & Miesbach, S. 1995 Redundancy reduction with information-preserving nonlinear maps. *Network* **6**, 61–72.

Parra, L., Deco, G. & Miesbach, S. 1996 Statistical independence and novelty detection with information-preserving nonlinear maps. *Neural Computation* **8**, 260–269.

Taleb, A. & Jutten, C. 1999 Source separation in post-nonlinear mixtures. *IEEE Trans. on Signal Processing* **47**, 2807–2820.

Yang, H., Amari, S.-I. & Cichocki, A. 1998 Information-theoretic approach to blind separation of sources in non-linear mixture. *Signal Processing* **64**, 291–300.