Neural Computation, in press

Independent Factor Analysis

H. Attias

Sloan Center for Theoretical Neurobiology and
W.M. Keck Foundation Center for Integrative Neuroscience
University of California at San Francisco
513 Parnassus Avenue
San Francisco, CA 94143-0444

Abstract

We introduce the independent factor analysis (IFA) method for recovering independent hidden sources from their observed mixtures. IFA generalizes and unifies ordinary factor analysis (FA), principal component analysis (PCA), and independent component analysis (ICA), and can handle not only square noiseless mixing, but also the general case where the number of mixtures differs from the number of sources and the data are noisy. IFA is a two-step procedure. In the first step, the source densities, mixing matrix and noise covariance are estimated from the observed data by maximum likelihood. For this purpose we present an expectation-maximization (EM) algorithm, which performs unsupervised learning of an associated probabilistic model of the mixing situation. Each source in our model is described by a mixture of Gaussians, thus all the probabilistic calculations can be performed analytically. In the second step, the sources are reconstructed from the observed data by an optimal non-linear estimator. A variational approximation of this algorithm is derived for cases with a large number of sources, where the exact algorithm becomes intractable. Our IFA algorithm reduces to the one for ordinary FA when the sources become Gaussian, and to an EM algorithm for PCA in the zero-noise limit. We derive an additional EM algorithm specifically for noiseless IFA. This algorithm is shown to be superior to ICA since it can learn arbitrary source densities from the data. Beyond blind separation, IFA can be used for modeling multi-dimensional data by a highly constrained mixture of Gaussians, and as a tool for non-linear signal encoding.

1 Statistical Modeling and Blind Source Separation

In the blind source separation (BSS) problem one is presented with multi-variable data measured by L' sensors. It is known that these data arise from L source signals that are mixed together by some linear transformation corrupted by noise. It is further known that the sources are mutually statistically independent. The task is to obtain those source signals. However, the sources are not observable and nothing is known about their properties beyond their mutual statistical independence, nor about the properties of the mixing process and the noise. In the absence of this information, one has to proceed 'blindly' to recover the source signals from their observed noisy mixtures.

Despite its signal-processing appearance, BSS is a problem in statistical modeling of data. In this context, one wishes to describe the L' observed variables y_i, which are generally correlated, in terms of a smaller set of L unobserved variables x_j that are mutually independent. The simplest such description is given by a probabilistic linear model,

$$y_i = \sum_{j=1}^{L} H_{ij} x_j + u_i \ , \qquad i = 1, ..., L' \ , \tag{1}$$

where y_i depends on linear combinations of the x_j's with constant coefficients H_{ij}; the probabilistic nature of this dependence is modeled by the L' additive noise signals u_i. In general, the statistician's task is to estimate H_{ij} and x_j. The latter are regarded as the independent 'causes' of the data in some abstract sense; their relation to the actual physical causes is often highly non-trivial. In BSS, on the other hand, the actual causes of the sensor signals y_i are the source signals x_j and the model (1), with H_{ij} being the mixing matrix, is known to be the correct description.

One might expect that, since linear models have been analyzed and applied extensively for many years, the solution to the BSS problem can be found in some textbook or review article. However, this is not the case. Consider, e.g., the close relation of (1) to the well-known factor analysis (FA) model (see Everitt 1984). In the context of FA, the unobserved sources x_j are termed 'common factors' (usually just 'factors'), the noise u_i 'specific factors', and the mixing matrix elements H_{ij} 'factor loadings'. The factor loadings and noise variances can be estimated from the data by, e.g., maximum likelihood (there exists an efficient expectation-maximization algorithm for this purpose), leading to an optimal estimate of the factors. However, ordinary FA cannot perform BSS. Its inadequacy stems from using a Gaussian model for the probability density $p(x_j)$ of each factor. This seemingly technical point turns out to have important consequences, since it implies that FA exploits only second-order statistics of the observed data to perform those estimates and hence, in effect, does not require the factors to be mutually independent but merely uncorrelated. As a result, the factors (and factor loading matrix) are not defined uniquely but only to within an arbitrary rotation, since the likelihood function is rotation-invariant in factor space. Put in the context of BSS, the true sources and mixing matrix cannot be distinguished from any rotation thereof when only second-order statistics are used. More modern statistical analysis methods, such as projection pursuit (Friedman and Stuetzle 1981; Huber 1985) and generalized additive models (Hastie and Tibshirani 1990), do indeed use non-Gaussian densities (modeled by non-linear functions of Gaussian variables), but the resulting models are quite restricted and are not suitable for solving the BSS problem.

Most of the work in the field of BSS since its emergence in the mid '80s (see Jutten and Herault 1991; Comon, Jutten and Herault 1991) aimed at a highly idealized version of the problem where the mixing is square ($L' = L$), invertible, instantaneous and noiseless. This version is termed 'independent component analysis' (ICA) (Comon 1994). A satisfactory solution for ICA was found only in the last few years (Bell and Sejnowski 1995; Cardoso and Laheld 1996; Pham 1996; Pearlmutter and Parra 1997; Hyvärinen and Oja 1997). Contrary to FA, algorithms for ICA employ non-Gaussian models of the source densities $p(x_j)$. Consequently, the likelihood is no longer rotation-invariant and the maximum-likelihood estimate of the mixing matrix is unique; for appropriately chosen $p(x_j)$ (see below) it is also correct.

Mixing in realistic situations, however, generally includes noise and different numbers of sources and sensors. As the noise level increases, the performance of ICA algorithms deteriorates and the separation quality decreases, as manifested by cross-talk and noisy outputs. More importantly, many situations have a relatively small number of sensors but many sources, and one would like to lump the low-intensity sources together and regard them as effective noise, while the separation focuses on the high-intensity ones. There is no way to accomplish this using ICA methods.

Another important problem in ICA is determining the source density model. The ability to learn the densities $p(x_j)$ from the observed data is crucial. However, existing algorithms usually employ a source model that is either fixed or has only limited flexibility. When the actual source densities in the problem are known in advance, this model can be tailored accordingly; otherwise, an inaccurate model often leads to failed separation, since the global maximum of the likelihood shifts away from the one corresponding to the correct mixing matrix. In principle, one can use a flexible parametric density model whose parameters may also be estimated by maximum likelihood (Mackay 1996; Pearlmutter and Parra 1997). However, ICA algorithms use gradient-ascent maximization methods, which result in rather slow learning of the density parameters.

In this paper we present a novel unsupervised learning algorithm for blind separation of non-square, noisy mixtures. The key to our approach lies in the introduction of a new probabilistic generative model, termed the *independent factor* (IF) model, described schematically in Figure 1. This model is defined by (1), associated with arbitrary non-Gaussian adaptive densities $p(x_j)$ for the factors. We define *independent* *factor analysis* (IFA) as the reconstruction of the unobserved factors x_j from the observed data y_i. Hence, performing IFA amounts to solving the BSS problem.

IFA is performed in two steps. The first consists of learning the IF model, parametrized by the mixing matrix, noise covariance, and source density parameters, from the data. To make the model analytically tractable while maintaining the ability to describe arbitrary sources, each source density is modeled by a mixture of one-dimensional Gaussians. This enables us to derive an expectation-maximization (EM) algorithm, given by (29,30), which performs maximum-likelihood estimation of all the parameters, the source densities included.

Due to the presence of noise, the sources can be recovered from the sensor signals only approximately. This is done in the second step of IFA using the posterior density of the sources given the data. Based on this posterior, we derive two different source estimators which provide optimal source reconstructions using the parameters learned in the first step. Both estimators, the first given by (36) and the second found iteratively using (38), are non-linear but each satisfies a different optimality criterion.

As the number of sources increases, the E-step of this algorithm becomes increasingly computationally expensive. For such cases we derive an approximate algorithm that is shown to be quite accurate. The approximation is based on the variational approach, first introduced in the context of feedforward probabilistic networks by Saul and Jordan (1995).

Our IFA algorithm reduces to ordinary FA when the model sources become Gaussian, and performs principal component analysis (PCA) when used in the zero-noise limit. An additional EM algorithm, derived specifically for noiseless IFA, is also presented (64–66). A particular version of this algorithm, termed 'Seesaw', is composed of two alternating phases, as shown schematically in Figure 8: the first phase learns the unmixing matrix while keeping the source densities fixed; the second phase freezes the unmixing matrix and learns the source densities using EM. Its ability to learn the source densities from the data in an efficient manner makes Seesaw a powerful extension of Bell and Sejnowski's (1995) ICA algorithm, since it can separate mixtures that ICA fails to separate.

IFA therefore generalizes and unifies ordinary FA, PCA and ICA and provides a new method for modeling multi-variable data in terms of a small set of independent hidden variables. Furthermore, IFA amounts to fitting those data to a mixture model of co-adaptive Gaussians (see Figure 3 (bottom right)), i.e., the Gaussians cannot adapt independently but are strongly constrained to move and expand together.

This paper deals only with instantaneous mixing. Real-world mixing situations are generally not instantaneous but include propagation delays and reverberations (described mathematically by convolutions in place of matrix multiplication in (1)). A significant step towards solving the convolutive BSS problem was taken by Attias and Schreiner (1998), who obtained a family of maximum likelihood-based learning algorithms for separating noiseless convolutive mixtures; Torkkola (1996) and Lee et al. (1997) derived one of those algorithms from information-maximization considerations. Algorithms for noisy convolutive mixing can be derived using an extension of the methods described here and will be presented elsewhere.

This paper is organized as follows. Section 2 introduces the IF model. The EM algorithm for learning the generative model parameters is presented in Section 3, and source reconstruction procedures are discussed in section 4. The performance of the IFA algorithm is demonstrated by its application to noisy mixtures of signals with arbitrary densities in Section 5. The factorized variational approximation of IFA is derived and tested in Section 6. The EM algorithm for noiseless IFA and its Seesaw version are presented and demonstrated in Section 7. Most derivations are relegated to Appendices A–C.

Notation

Throughout this paper, vectors are denoted by bold-faced lower-class letters and matrices by bold-faced upper-class letters. Vector and matrix elements are not bold-faced. The inverse of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{-1}$, and its transposition by $\mathbf{A}^T$ ($A^T_{ij} = A_{ji}$).

To denote ensemble averaging we use the operator E. Thus, if $\mathbf{x}^{(t)}$, $t = 1, ..., T$ are different observations of the random vector $\mathbf{x}$, then for any vector function $\mathbf{F}$ of $\mathbf{x}$,

$$E\mathbf{F}(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} \mathbf{F}(\mathbf{x}^{(t)}) \ . \tag{2}$$

The multi-variable Gaussian distribution for a random vector $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted by

$$\mathcal{G}(\mathbf{x}-\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mid \det(2\pi\boldsymbol{\Sigma}) \mid^{-1/2} \exp\left[-\left(\mathbf{x}-\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1} \left(\mathbf{x}-\boldsymbol{\mu}\right)/2\right] \ , \tag{3}$$

implying $E\mathbf{x} = \boldsymbol{\mu}$ and $E\mathbf{x}\mathbf{x}^T = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T$.

2 The Independent Factor (IF) Generative Model

Independent factor analysis is a two-step method. The first step is concerned with the unsupervised learning task of a generative model (Everitt 1984), named the *independent factor* (IF) model, which we introduce in the following. Let $\mathbf{y}$ be an $L' \times 1$ observed data vector. We wish to explain the correlated y_i in terms of L hidden variables x_j, referred to as 'factors', that are *mutually statistically independent*. Specifically, the data are modeled as dependent on linear combinations of the factors with constant coefficients H_{ij}, and an additive $L' \times 1$ random vector $\mathbf{u}$ makes this dependence non-deterministic:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{u} \ . \tag{4}$$

In the language of BSS, the independent factors $\mathbf{x}$ are the unobserved source signals and the data $\mathbf{y}$ are the observed sensor signals. The sources are mixed by the matrix $\mathbf{H}$. The resulting mixtures are corrupted by noise signals $\mathbf{u}$ originating in the sources, the mixing process (e.g., the propagation medium response), or the sensor responses.

In order to produce a generative model for the probability density of the sensor signals $p(\mathbf{y})$, we must first specify the density of the sources and of the noise. We model the sources x_i as L independent random variables with arbitrary distributions $p(x_i \mid \theta_i)$, where the individual i-th source density is parametrized by the parameter set θ_i.

The noise is assumed to be Gaussian with mean zero and covariance matrix $\boldsymbol{\Lambda}$, allowing correlations between sensors; note that even in situations where the sensor noise signals are independent, correlations may arise due to source noise or propagation noise. Hence

$$p(\mathbf{u}) = \mathcal{G}(\mathbf{u}, \boldsymbol{\Lambda}) \ . \tag{5}$$

Equations (4–5) define the IF generative model, which is parametrized by the source parameters $\boldsymbol{\theta}$, mixing matrix $\mathbf{H}$, and noise covariance $\boldsymbol{\Lambda}$. We denote the IF parameters collectively by

$$W = (\mathbf{H}, \boldsymbol{\Lambda}, \boldsymbol{\theta}) \ . \tag{6}$$

The resulting model sensor density is

$$\begin{aligned} p(\mathbf{y} \mid W) &= \int d\mathbf{x}\, p(\mathbf{y} \mid \mathbf{x})\, p(\mathbf{x}) \\ &= \int d\mathbf{x}\, \mathcal{G}(\mathbf{y} - \mathbf{H}\mathbf{x}, \boldsymbol{\Lambda}) \prod_{i=1}^{L} p(x_i, \theta_i) \ , \end{aligned} \tag{7}$$

where $d\mathbf{x} = \prod_i dx_i$. The parameters W should be adapted to minimize an error function which measures the distance between the model and observed sensor densities.

2.1 Source Model: Factorial Mixture of Gaussians

Although in principle $p(\mathbf{y})$ (7) is a perfectly viable starting point and can be evaluated by numerical integration given a suitably chosen $p(x_i)$, this could become quite computationally intensive in practice. A better strategy is to choose a parametric form for $p(x_i)$ which (i) is sufficiently general to model arbitrary source densities, and (ii) allows performing the integral in (7) analytically. A form that satisfies both these requirements is the mixture of Gaussians (MOG) model.

In this paper we shall describe the density of source i as a mixture of n_i Gaussians $q_i = 1, ..., n_i$ with means μ_{i,q_i}, variances ν_{i,q_i}, and mixing proportions w_{i,q_i}:

$$p(x_i \mid \theta_i) = \sum_{q_i=1}^{n_i} w_{i,q_i} \, \mathcal{G}(x_i - \mu_{i,q_i}, \nu_{i,q_i}) \ , \qquad \theta_i = \{w_{i,q_i}, \mu_{i,q_i}, \nu_{i,q_i}\} \ , \tag{8}$$

where q_i runs over the Gaussians of source i. For this mixture to be normalized, the mixing proportions for each source should sum up to unity: $\sum_{q_i} w_{i,q_i} = 1$.

The parametric form (8) provides a probabilistic generative description of the sources in which the different Gaussians play the role of hidden states. To generate the source signal x_i, we first pick a state q_i with probability $p(q_i) = w_{i,q_i}$, then draw a number x_i from the corresponding Gaussian density $p(x_i \mid q_i) = \mathcal{G}(x_i - \mu_{i,q_i}, \nu_{i,q_i})$.

Viewed in L-dimensional space, the joint source density $p(\mathbf{x})$ formed by the product of the one-dimensional MOG's (8) is itself a MOG. Its collective hidden states

$$\mathbf{q} = (q_1, ..., q_L) \tag{9}$$

consist of all possible combinations of the individual source states q_i. As Figure 3 (upper right) illustrates for $L = 2$, each state $\mathbf{q}$ corresponds to an L-dimensional Gaussian density whose mixing proportions $w_{\mathbf{q}}$, mean $\boldsymbol{\mu}_{\mathbf{q}}$ and diagonal covariance matrix $\mathbf{V}_{\mathbf{q}}$ are determined by those of the constituent source states,

$$w_{\mathbf{q}} = \prod_{i=1}^{L} w_{i,q_i} = w_{1,q_1} \cdots w_{L,q_L} \ , \qquad \boldsymbol{\mu}_{\mathbf{q}} = (\mu_{1,q_1}, ..., \mu_{L,q_L}) \ , \qquad \mathbf{V}_{\mathbf{q}} = \mathrm{diag}\,(\nu_{1,q_1}, ..., \nu_{L,q_L}) \ . \tag{10}$$

Hence we have

$$p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{i=1}^{L} p(x_i \mid \theta_i) = \sum_{\mathbf{q}} w_{\mathbf{q}} \, \mathcal{G}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{q}}, \mathbf{V}_{\mathbf{q}}) \ , \tag{11}$$

where the Gaussians factorize, $\mathcal{G}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{q}}, \mathbf{V}_{\mathbf{q}}) = \prod_i \mathcal{G}(x_i - \mu_{i,q_i}, \nu_{i,q_i})$, and the sum over collective states $\mathbf{q}$ (9) represents summing over all the individual source states, $\sum_{\mathbf{q}} = \sum_{q_1} \cdots \sum_{q_L}$.

Note that, contrary to ordinary MOG, the Gaussians in (11) are not free to adapt independently but are rather strongly constrained. Modifying the mean and variance of a single source state q_i would result in shifting a whole column of collective states $\mathbf{q}$. Our source model is therefore a mixture of *co-adaptive* Gaussians, termed 'factorial MOG'. We point out that Hinton and Zemel (1994) proposed and studied a related generative model, which differed from the present one in that all Gaussians had the same covariance; an EM algorithm for their model was derived by Ghahramani (1995). Different forms of co-adaptive MOG were used by Hinton et al. (1992) and by Bishop et al. (1998).

2.2 Sensor Model

The source model (11), combined by (4) with the noise model (5), leads to a two-step generative model of the observed sensor signals. This model can be viewed as a hierarchical feedforward network with a visible layer and two hidden layers, as shown in Figure 1. To generate sensor signals $\mathbf{y}$,

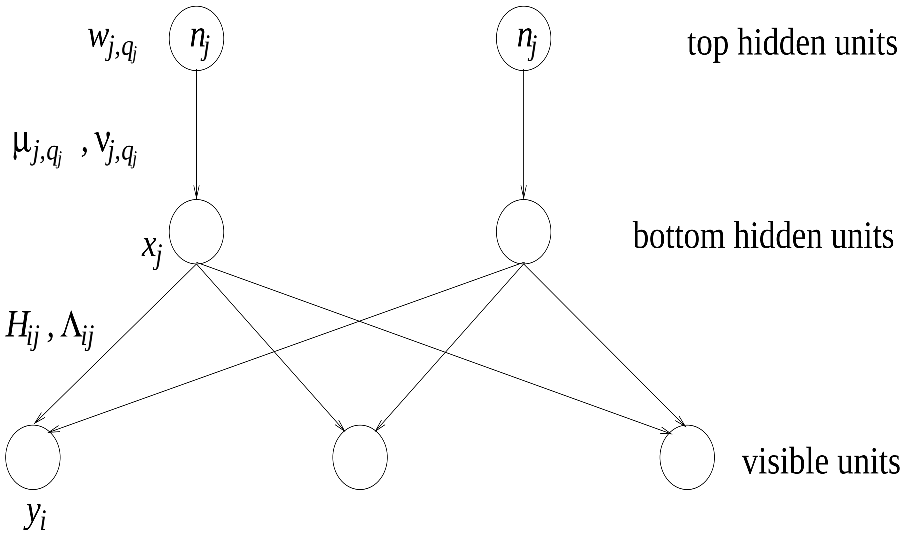

Figure 1: Feedforward network representation of the IF generative model. Each source signal x_j is generated by an independent n_j-state MOG model (8). The sensor signals y_i are generated from a Gaussian model (14) whose mean depends linearly on the sources.

(i) Pick a unit q_i for each source i with probability

$$p(\mathbf{q}) = w_{\mathbf{q}} \tag{12}$$

from the top hidden layer of source states. This unit has a top-down generative connection with weight μ_{j,q_j} to each of the units j in the bottom hidden layer. When activated, it causes unit j to produce a sample x_j from a Gaussian density centered at μ_{j,q_j} with variance ν_{j,q_j}; the probability of generating a particular source vector $\mathbf{x}$ in the bottom hidden layer is

$$p(\mathbf{x} \mid \mathbf{q}) = \mathcal{G}(\mathbf{x} - \boldsymbol{\mu}_{\mathbf{q}}, \mathbf{V}_{\mathbf{q}}) \ . \tag{13}$$

(ii) Each unit j in the bottom hidden layer has a top-down generative connection with weight H_{ij} to each unit i in the visible layer. Following the generation of $\mathbf{x}$, unit i produces a sample y_i from a Gaussian density centered at $\sum_j H_{ij} x_j$. In case of independent sensor noise, the variance of this density is Λ_{ii}; generally the noise is correlated across sensors and the probability for generating a particular sensor vector $\mathbf{y}$ in the visible layer is

$$p(\mathbf{y} \mid \mathbf{x}) = \mathcal{G}(\mathbf{y} - \mathbf{H}\mathbf{x}, \boldsymbol{\Lambda}) \ . \tag{14}$$

It is important to emphasize that our IF generative model is probabilistic: it describes the *statistics* of the unobserved source and observed sensor signals, i.e., the densities $p(\mathbf{x})$ and $p(\mathbf{y})$, rather than the actual signals $\mathbf{x}$ and $\mathbf{y}$. This model is fully described by the joint density of the visible layer and the two hidden layers,

$$p(\mathbf{q}, \mathbf{x}, \mathbf{y} \mid W) = p(\mathbf{q}) \, p(\mathbf{x} \mid \mathbf{q}) \, p(\mathbf{y} \mid \mathbf{x}) \ . \tag{15}$$

Notice from (15) that, since the sensor signals depend on the sources but not on the source states, i.e., $p(\mathbf{y} \mid \mathbf{x}, \mathbf{q}) = p(\mathbf{y} \mid \mathbf{x})$ (once $\mathbf{x}$ is produced, the identity of the state $\mathbf{q}$ that generated it becomes irrelevant), the IF network layers form a top-down first-order Markov chain.

The generative model attributes a probability $p(\mathbf{y})$ for each observed sensor data vector $\mathbf{y}$. We are now able to return to (7) and express $p(\mathbf{y})$ in a closed form. From (15) we have

$$p(\mathbf{y} \mid W) \ = \ \sum_{\mathbf{q}} \int d\mathbf{x} \, p(\mathbf{q}) \, p(\mathbf{x} \mid \mathbf{q}) \, p(\mathbf{y} \mid \mathbf{x}) = \sum_{\mathbf{q}} p(\mathbf{q}) \, p(\mathbf{y} \mid \mathbf{q}) \ , \tag{16}$$

where, thanks to the Gaussian forms (13,14), the integral over the sources in (16) can be performed analytically to yield

$$p(\mathbf{y} \mid \mathbf{q}) = \mathcal{G}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu}_{\mathbf{q}},\ \mathbf{H}\mathbf{V}_{\mathbf{q}}\mathbf{H}^T + \boldsymbol{\Lambda})\ . \tag{17}$$

Thus, like the source density, our sensor density model is a co-adaptive (although not factorial) MOG, as is illustrated in Figure 3 (bottom right). Changing one element of the mixing matrix would result in a rigid rotation and scaling of a whole line of states. Learning the IF model therefore amounts to fitting the sensor data by a mixture of co-adaptive Gaussians, then use them to deduce the model parameters.

3 Learning the IF Model

3.1 Error Function and Maximum Likelihood

To estimate the IF model parameters we first define an error function which measures the difference between our model sensor density $p(\mathbf{y} \mid W)$ (16) and the observed one $p^o(\mathbf{y})$. The parameters W are then adapted iteratively to minimize this error. We choose the Kullback-Leibler (KL) distance function (Cover and Thomas 1991), defined by

$$\mathcal{E}(W) = \int d\mathbf{y}\ p^o(\mathbf{y}) \log \frac{p^o(\mathbf{y})}{p(\mathbf{y} \mid W)} = -E\left[\log p(\mathbf{y} \mid W)\right] - H_{p^o} \tag{18}$$

where the operator E performs averaging over the observed $\mathbf{y}$. As is well known, the KL distance $\mathcal{E}$ is always non-negative and vanishes when $p(\mathbf{y}) = p^o(\mathbf{y})$.

The error (18) consists of two terms: the first is the negative log-likelihood of the observed sensor signals given the model parameters W. The second term is the sensor entropy, which is independent of W and will henceforth be dropped. Minimizing $\mathcal{E}$ is thus equivalent to maximizing the likelihood of the data with respect to the model.

The KL distance has an interesting relation also to the mean square point-by-point distance. To see it, we define the *relative error* of $p(\mathbf{y} \mid W)$ with respect to the true density $p^o(\mathbf{y})$ by

$$e(\mathbf{y}) = \frac{p(\mathbf{y}) - p^o(\mathbf{y})}{p^o(\mathbf{y})} \tag{19}$$

at each $\mathbf{y}$, omitting the dependence on W. When $p(\mathbf{y})$ in (18) is expressed in terms of $e(\mathbf{y})$, we obtain

$$\mathcal{E}(W) = -\int d\mathbf{y}\ p^o(\mathbf{y}) \log\left[1 + e(\mathbf{y})\right] \approx \frac{1}{2}\int d\mathbf{y}\ p^o(\mathbf{y})\ e^2(\mathbf{y})\ , \tag{20}$$

where the approximation $\log e \approx e - e^2/2$, valid in the limit of small $e(\mathbf{y})$, was used. Hence, in the parameter regime where the model $p(\mathbf{y} \mid W)$ is 'near' the observed density, minimizing $\mathcal{E}$ amounts to minimizing the mean square relative error of the model density. This property, however, has little computational significance.

A straightforward way to minimize the error (18) would be to use the gradient-descent method where, starting from random values, the parameters are incremented at each iteration by a small step in the direction of the gradient $\partial\mathcal{E}/\partial W$. However, this results in rather slow learning. Instead, we shall employ the expectation-maximization approach to develop an efficient algorithm for learning the IF model.

3.2 An Expectation-Maximization Algorithm

Expectation-maximization (EM) (Dempster et al. 1977; Neal and Hinton 1998) is an iterative method to maximize the log-likelihood of the observed data with respect to the parameters of the generative model describing those data. It is obtained by noting that, in addition to the likelihood $E[\log p(\mathbf{y} \mid W)]$ of the observed sensor data (see (18)), one may consider the likelihood $E[\log p(\mathbf{y}, \mathbf{x}, \mathbf{q} \mid W)]$ of the 'complete' data, composed of both the observed and the 'missing' data, i.e., the unobserved source signals and states. For each observed $\mathbf{y}$, this complete-data likelihood as a function of $\mathbf{x}, \mathbf{q}$ is a random variable. Each iteration then consists of two steps:

(E) Calculate the expected value of the complete-data likelihood, given the observed data and the current model. That is, calculate

$$\mathcal{F}(W', W) = -E\left[\log p(\mathbf{q}, \mathbf{x}, \mathbf{y} \mid W)\right] + \mathcal{F}_H(W') \ , \tag{21}$$

where, for each observed $\mathbf{y}$, the average in the first term on the r.h.s. is taken over the unobserved $\mathbf{x}, \mathbf{q}$ using the source posterior $p(\mathbf{x}, \mathbf{q} \mid \mathbf{y}, W')$; W' are the parameters obtained in the previous iteration, and $\mathcal{F}_H(W')$ is the entropy of the posterior (see (27)). The result is then averaged over all the observed $\mathbf{y}$. The second term on the r.h.s is W-independent and has no effect on the following.

(M) Minimize $\mathcal{F}(W', W)$ (i.e., maximize the corresponding averaged likelihood) with respect to W to obtain the new parameters:

$$W = \arg\min_{W''} \mathcal{F}(W', W'') \ . \tag{22}$$

In the following we develop the EM algorithm for our IF model. First, we show that $\mathcal{F}$ (21) is bounded from below by the error $\mathcal{E}$ (18), following Neal and Hinton (1998). Dropping the average over the observed $\mathbf{y}$, we have

$$\begin{aligned} \mathcal{E}(W) &= -\log p(\mathbf{y} \mid W) = -\log \sum_{\mathbf{q}} \int d\mathbf{x} \ p(\mathbf{q}, \mathbf{x}, \mathbf{y} \mid W) \\ &\leq -\sum_{\mathbf{q}} \int d\mathbf{x} \ p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}) \ \log \frac{p(\mathbf{q}, \mathbf{x}, \mathbf{y} \mid W)}{p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y})} \equiv \mathcal{F} \ , \end{aligned} \tag{23}$$

where the second line follows from Jensen's inequality (Cover and Thomas 1991) and holds for any conditional density p'. In EM, we choose p' to be the source posterior computed using the parameters from the previous iteration,

$$p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}) = p(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, W') \ , \tag{24}$$

which is obtained directly from (15) with $W = W'$.

Hence, after the previous iteration we have an approximate error function $\mathcal{F}(W', W)$ which, due to the Markov property (15) of the IF model, is obtained by adding up four terms,

$$\mathcal{E}(W) \leq \mathcal{F}(W', W) = \mathcal{F}_V + \mathcal{F}_B + \mathcal{F}_T + \mathcal{F}_H \ , \tag{25}$$

to be defined shortly. A closer inspection reveals that, while they all depend on the model parameters W, each of the first three terms involves only the parameters of a single layer (see Figure 1). Thus, $\mathcal{F}_V$ depends only on the parameters $\mathbf{H}, \mathbf{\Lambda}$ of the visible layer, whereas $\mathcal{F}_B$ and $\mathcal{F}_T$ depend on the parameters $\{\mu_{i,q_i}, \nu_{i,q_i}\}$ and $\{w_{i,q_i}\}$ of the bottom and top hidden layers, respectively; note that they also depend on all the previous parameters W'. From (15) and (23), the contribution of the different layers are given by

$$\begin{aligned} \mathcal{F}_V(W', \mathbf{H}, \mathbf{\Lambda}) &= -\int d\mathbf{x} \ p(\mathbf{x} \mid \mathbf{y}, W') \ \log p(\mathbf{y} \mid \mathbf{x}) \ , \\ \mathcal{F}_B(W', \{\mu_{i,q_i}, \nu_{i,q_i}\}) &= -\sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid \mathbf{y}, W') \int dx_i \ p(x_i \mid q_i, \mathbf{y}, W') \ \log p(x_i \mid q_i) \ , \\ \mathcal{F}_T(W', \{w_{i,q_i}\}) &= -\sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid \mathbf{y}, W') \ \log p(q_i) \ , \end{aligned} \tag{26}$$

and the last contribution is the negative entropy of the source posterior

$$\mathcal{F}_H(W') = \sum_{\mathbf{q}} \int d\mathbf{x}\, p(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, W') \, \log p(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, W') \ . \tag{27}$$

To get $\mathcal{F}_B$ (second line in (26)) we used $p(\mathbf{q} \mid \mathbf{x})p(\mathbf{x} \mid \mathbf{y}) = p(\mathbf{q} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{q}, \mathbf{y})$, which can be obtained using (15).

The EM procedure now follows by observing that (25) becomes an equality when $W = W'$, thanks to the choice (24). Hence, given the parameter values W' produced by the previous iteration, the E-step (21) results in the approximate error coinciding with the true error, $\mathcal{F}(W', W') = \mathcal{E}(W')$. Next, we consider $\mathcal{F}(W', W)$ and minimize it with respect to W. From (25), the new parameters obtained from the M-step (22) satisfy

$$\mathcal{E}(W) \le \mathcal{F}(W', W) \le \mathcal{F}(W', W') = \mathcal{E}(W') \ , \tag{28}$$

proving that the current EM step does not increase the error.

The EM algorithm for learning the IF model parameters is derived from (25–26) in Appendix A, where the new parameters W at each iteration are obtained in terms of the old ones W'. The learning rules for the mixing matrix and noise covariance are given by

$$\begin{aligned} \mathbf{H} &= E\mathbf{y}\langle \mathbf{x}^T \mid \mathbf{y}\rangle \left(E\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y}\rangle\right)^{-1} \ , \\ \mathbf{\Lambda} &= E\mathbf{y}\mathbf{y}^T - E\mathbf{y}\langle \mathbf{x}^T \mid \mathbf{y}\rangle \mathbf{H}^T \ , \end{aligned} \tag{29}$$

whereas the rules for the source MOG parameters are

$$\begin{aligned} \mu_{i,q_i} &= \frac{Ep(q_i \mid \mathbf{y})\langle x_i \mid q_i, \mathbf{y}\rangle}{Ep(q_i \mid \mathbf{y})} \ , \\ \nu_{i,q_i} &= \frac{Ep(q_i \mid \mathbf{y})\langle x_i^2 \mid q_i, \mathbf{y}\rangle}{Ep(q_i \mid \mathbf{y})} - \mu_{i,q_i}^2 \ , \\ w_{i,q_i} &= Ep(q_i \mid \mathbf{y}) \ . \end{aligned} \tag{30}$$

Notation. $\langle \mathbf{x} \mid \mathbf{y}\rangle$ is an $L \times 1$ vector denoting the conditional mean of the sources given the sensors; the $L \times L$ matrix $\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y}\rangle$ is the source covariance conditioned on the sensors. Similarly, $\langle x_i \mid q_i, \mathbf{y}\rangle$ denotes the mean of sensor i conditioned on both the hidden state q_i of this source and the observed sensors. $p(q_i \mid \mathbf{y})$ is the probability of the state q_i of source i conditioned on the sensors. The conditional averages are defined in (76,78). Both the conditional averages and the conditional probabilities depend on the observed sensor signals $\mathbf{y}$ and on the parameters W', and are computed during the E-step. Finally, the operator E performs averaging over the observed $\mathbf{y}$.

Scaling. In the BSS problem, the sources are defined only to within an order permutation and scaling. This ambiguity is implied by (4): the effect of an arbitrary permutation of the sources can be cancelled by a corresponding permutation of the columns of $\mathbf{H}$, leaving the observed $\mathbf{y}$ unchanged. Similarly, scaling source x_j by a factor σ_j would not affect $\mathbf{y}$ if the j-th column of $\mathbf{H}$ is scaled by $1/\sigma_j$ at the same time. Put another way, the error function cannot distinguish between the true $\mathbf{H}$ and a scaled and permuted version of it, and thus possesses multiple continuous manifolds of global minima. Whereas each point on those manifolds corresponds to a valid solution, their existence may delay convergence and cause numerical problems (e.g., H_{ij} may acquire arbitrarily large values). To minimize the effect of such excessive freedom, we maintain the variance of each source at unity by performing the following scaling transformation at each iteration:

$$\begin{aligned} \sigma_j^2 &= \sum_{q_j=1}^{n_j} w_{j,q_j}(\nu_{j,q_j} + \mu_{j,q_j}^2) - (\sum_{q_j=1}^{n_j} w_{j,q_j}\mu_{j,q_j})^2 \ , \\ \mu_{j,q_j} &\to \frac{\mu_{j,q_j}}{\sigma_j} \ , \quad \nu_{j,q_j} \to \frac{\nu_{j,q_j}}{\sigma_j^2} \ , \quad H_{ij} \to H_{ij}\sigma_j \ . \end{aligned} \tag{31}$$

This transformation amounts to scaling each source j by its standard deviation $\sigma_j = \sqrt{Ex_j^2 - (Ex_j)^2}$ and compensating the mixing matrix appropriately. It is easy to show that this scaling leaves the error function unchanged.

3.3 Hierarchical Interpretation

The above EM algorithm can be given a natural interpretation in the context of our hierarchical generative model (see Figure 1). From this point of view, it bears some resemblance to the mixture of experts algorithm of Jordan and Jacobs (1994). Focusing first on the learning rules (30) for the top hidden layer parameters, one notes their similarity to the usual EM rules for fitting a MOG model. To make the connection explicit we rewrite the rules (30) on the left column below,

$$\begin{aligned}
\mu_{i,q_i} &= \frac{E\int dx_i\, p(x_i \mid \mathbf{y})\ [p(q_i \mid x_i, \mathbf{y})\, x_i]}{E\int dx_i\, p(x_i \mid \mathbf{y})\ [p(q_i \mid x_i, \mathbf{y})]} && \longleftrightarrow \quad \frac{E\, p(q_i \mid x_i)\, x_i}{E\, p(q_i \mid x_i)}\,, \\
\nu_{i,q_i} &= \frac{E\int dx_i\, p(x_i \mid \mathbf{y})\ [p(q_i \mid x_i, \mathbf{y})\, x_i^2]}{E\int dx_i\, p(x_i \mid \mathbf{y})\ [p(q_i \mid x_i, \mathbf{y})]} - \mu_{i,q_i}^2 && \longleftrightarrow \quad \frac{E\, p(q_i \mid x_i)\, x_i^2}{E\, p(q_i \mid x_i)} - \mu_{i,q_i}^2\,, \\
w_{i,q_i} &= E\int dx_i\, p(x_i \mid \mathbf{y})\ [p(q_i \mid x_i, \mathbf{y})] && \longleftrightarrow \quad E\, p(q_i \mid x_i)\,,
\end{aligned} \tag{32}$$

where to go from (30) to the left column of (32) we used $p(q_i \mid \mathbf{y}) = \int dx_i\, p(x_i, q_i \mid \mathbf{y})$ and $p(q_i \mid \mathbf{y})\langle m(x_i) \mid q_i, \mathbf{y}\rangle = \int dx_i\, m(x_i)\, p(x_i, q_i \mid \mathbf{y})$ (see (78)). Note that each p in (32) should be read as p'.

Shown on the right column of (32) are the standard EM rules for learning a one-dimensional MOG model parametrized by μ_{i,q_i}, ν_{i,q_i} and w_{i,q_i} for each source x_i, assuming the source signals were directly observable. A comparison with the square-bracketed expressions on the left column shows that the EM rules (30) for the IF source parameters are precisely the rules for learning a separate MOG model for each source i, with the actual x_i replaced by all values x_i that are possible given the observed sensor signals $\mathbf{y}$, weighted by their posterior $p(x_i \mid \mathbf{y})$.

The EM algorithm for learning the IF model can therefore be viewed hierarchically: the visible layer learns a noisy linear model for the sensor data, parametrized by $\mathbf{H}$ and $\mathbf{\Lambda}$. The hidden layers learn a MOG model for each source. Since the actual sources are not available, all possible source signals are used, weighted by their posterior given the observed data; this couples the visible and hidden layers since all the IF parameters participate in computing that posterior.

3.4 Relation to Ordinary Factor Analysis

Ordinary factor analysis (FA) uses a generative model of independent Gaussian sources with zero mean and unit variance, $p(x_i) = \mathcal{G}(x_i, 1)$, mixed (see (4)) by a linear transformation with added Gaussian noise whose covariance matrix $\mathbf{\Lambda}$ is diagonal. This is a special case of our IF model obtained when each source has a single state ($n_i = 1$ in (8)). From (16,17), the resulting sensor density is

$$p(\mathbf{y} \mid W) = \mathcal{G}(\mathbf{y}, \mathbf{H}\mathbf{H}^T + \mathbf{\Lambda})\,, \tag{33}$$

since we now have only one collective source state $\mathbf{q} = (1, 1, ..., 1)$ with $w_{\mathbf{q}} = 1$, $\boldsymbol{\mu}_{\mathbf{q}} = \mathbf{0}$, and $\mathbf{V}_{\mathbf{q}} = \mathbf{I}$ (see (9-10)).

The invariance of FA under factor rotation mentioned in Section 1 is manifested in the FA model density (33). For any $L \times L'$ matrix $\mathbf{P}$ whose rows are orthonormal (i.e., $\mathbf{P}\mathbf{P}^T = \mathbf{I}$ – a rotation matrix), we can define a new mixing matrix $\mathbf{H}' = \mathbf{H}\mathbf{P}$. However, the density (33) does not discriminate between $\mathbf{H}'$ and the true $\mathbf{H}$ since $\mathbf{H}'\mathbf{H}'^{\,T} = \mathbf{H}\mathbf{H}^T$, rendering FA unable to identify the true mixing matrix. Notice from (4) that the factors corresponding to $\mathbf{H}'$ are obtained from the true sources by that rotation: $\mathbf{x}' = \mathbf{P}^T\mathbf{x}$. In contrast, our IF model density (16,17) is, in general, not invariant under the transformation $\mathbf{H} \to \mathbf{H}'$; the rotational symmetry is broken by the MOG source model. Hence the true $\mathbf{H}$ can, in principle, be identified.

We point out that for square mixing ($L' = L$) the symmetry of the FA density (33) is even larger: for an arbitrary diagonal noise covariance $\mathbf{\Lambda}'$, the transformation $\mathbf{\Lambda} \to \mathbf{\Lambda}'$, $\mathbf{H} \to \mathbf{H}' = (\mathbf{H}\mathbf{H}^T + \mathbf{\Lambda} - \mathbf{\Lambda}')^{1/2}\mathbf{P}$ leaves (33) invariant. Hence not only the mixing but also the noise cannot be identified in this case.

The well-known EM algorithm for FA (Rubin and Thayer 1982) is obtained as a special case of our IFA algorithm (29,30), by freezing the source parameters at their values under (33) and using only the learning rules (29). Given the observed sensors $\mathbf{y}$, the source posterior now becomes simply a Gaussian, $p(\mathbf{x} \mid \mathbf{y}) = \mathcal{G}(\mathbf{x} - \boldsymbol{\rho}, \mathbf{\Sigma})$, whose covariance and data-dependent mean are given by

$$\mathbf{\Sigma} = \left(\mathbf{H}^T\mathbf{\Lambda}^{-1}\mathbf{H} + \mathbf{I}\right)^{-1} \ , \qquad \boldsymbol{\rho}(\mathbf{y}) = \mathbf{\Sigma}\ \mathbf{H}^T\mathbf{\Lambda}^{-1}\mathbf{y} \ , \tag{34}$$

rather than the MOG implied by (80-82). Consequently, the conditional source mean and covariance (84) used in (29) become $\langle \mathbf{x} \mid \mathbf{y} \rangle = \boldsymbol{\rho}(\mathbf{y})$ and $\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y} \rangle = \mathbf{\Sigma} + \boldsymbol{\rho}(\mathbf{y})\boldsymbol{\rho}(\mathbf{y})^T$.

4 Recovering the Sources

Once the IF generative model parameters have been estimated, the sources can be reconstructed from the sensor signals. A complete reconstruction is possible only when noise is absent and the mixing is invertible, i.e., if $\mathbf{\Lambda} = \mathbf{0}$ and rank $\mathbf{H} \geq L$; in this case, the sources are given by the pseudo-inverse of $\mathbf{H}$ via the linear relation $\mathbf{x} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$.

In general, however, an estimate $\hat{\mathbf{x}}(\mathbf{y})$ of the sources must be found. There are many ways to obtain a parametric estimator of an unobserved signal from data. In the following we discuss two of them, the least mean squares (LMS) and maximum a-posteriori probability (MAP) source estimators. Both are non-linear functions of the data, but each satisfies a different optimality criterion. It is easy to show that for Gaussian sources, they both reduce to the same linear estimator of ordinary FA, given by $\hat{\mathbf{x}}(\mathbf{y}) = \boldsymbol{\rho}(\mathbf{y})$ in (34). For non-Gaussian sources, however, the LMS and MAP estimators differ and neither has an a-priori advantage over the other. For either choice, obtaining the source estimate $\{\hat{\mathbf{x}}(\mathbf{y})\}$ for a given sensor data set $\{\mathbf{y}\}$ completes the IFA of these data.

4.1 LMS Estimator

As is well known, the optimal estimate in the least-square sense, i.e., that minimizes $E(\hat{\mathbf{x}} - \mathbf{x})^2$, is given by the conditional mean of the sources given the observed sensors,

$$\hat{\mathbf{x}}^{LMS}(\mathbf{y}) = \langle \mathbf{x} \mid \mathbf{y} \rangle = \int d\mathbf{x}\ \mathbf{x}\ p(\mathbf{x} \mid \mathbf{y}, W) \ , \tag{35}$$

where $p(\mathbf{x} \mid \mathbf{y}, W) = \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{y})p(\mathbf{x} \mid \mathbf{q}, \mathbf{y})$ (see (80–82)) is the source posterior and depends on the generative parameters. This conditional mean has already been calculated for the E-step of our EM algorithm; as shown in Appendix A, it is given by a weighted sum of terms that are linear in the data $\mathbf{y}$,

$$\hat{\mathbf{x}}^{LMS}(\mathbf{y}) \quad = \quad \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{y})\ \ (\mathbf{A}_{\mathbf{q}}\,\mathbf{y} + \mathbf{b}_{\mathbf{q}}) \ , \tag{36}$$

where $\mathbf{A}_{\mathbf{q}} = \mathbf{\Sigma}_{\mathbf{q}}\mathbf{H}^T\mathbf{\Lambda}^{-1}$, $\mathbf{b}_{\mathbf{q}} = \mathbf{\Sigma}_{\mathbf{q}}\mathbf{V}_{\mathbf{q}}^{-1}\boldsymbol{\mu}_{\mathbf{q}}$, and $\mathbf{\Sigma}_{\mathbf{q}}$ is given in terms of the generative parameters in (81). Notice that the weighting coefficients themselves depend non-linearly on the data via $p(\mathbf{q} \mid \mathbf{y}) = p(\mathbf{y} \mid \mathbf{q})p(\mathbf{q}) / \sum_{\mathbf{q}'} p(\mathbf{y} \mid \mathbf{q}')p(\mathbf{q}')$ and (12,17).

4.2 MAP Estimator

The MAP optimal estimator maximizes the source posterior $p(\mathbf{x} \mid \mathbf{y})$. For a given y, maximizing the posterior is equivalent to maximizing the joint $p(\mathbf{x}, \mathbf{y})$ or its logarithm, hence

$$\hat{\mathbf{x}}^{MAP}(\mathbf{y}) = \arg\max_{\mathbf{x}} \left[\log p(\mathbf{y} \mid \mathbf{x}) + \sum_{i=1}^{L} \log p(x_i)\right] \ . \tag{37}$$

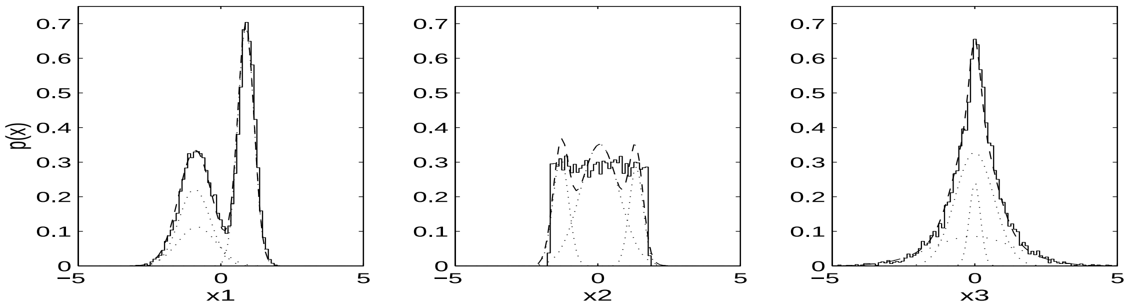

Figure 2: Source density histograms (solid lines) and their MOG models learned by IFA (dashed lines). Each model is a sum of three weighted Gaussian densities (dotted lines). Shown are bimodal (left) and uniform (middle) synthetic signals and a real speech signal (right).

A simple way to compute this estimator is to maximize the quantity on the r.h.s. of (37) iteratively using the method of gradient ascent, for each data vector $\mathbf{y}$. After initialization, $\hat{\mathbf{x}}(\mathbf{y})$ is incremented at each iteration by

$$\delta\hat{\mathbf{x}} = \eta\mathbf{H}^T\mathbf{\Lambda}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) - \eta\phi(\hat{\mathbf{x}}) \ , \tag{38}$$

where η is the learning rate and $\phi(\mathbf{x})$ is an $L \times 1$ vector given by the logarithmic derivative of the source density (8),

$$\phi(x_i) = -\frac{\partial \log p(x_i)}{\partial x_i} = -\sum_{q_i=1}^{n_i} p(q_i \mid x_i)\frac{x_i - \mu_{i,q_i}}{\nu_{i,q_i}} \ . \tag{39}$$

A good initialization is provided by the pseudo-inverse relation $\hat{\mathbf{x}}(\mathbf{y}) = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$. However, since the posterior may have multiple maxima, several initial values should be used in order to identify the highest maximum.

Notice that $\hat{\mathbf{x}}^{MAP}$ is a fixed point of the equation $\delta\hat{\mathbf{x}} = 0$. This equation is non-linear, reflecting the non-Gaussian nature of the source densities. A simple analysis shows that this fixed point is stable when $\mid \det \mathbf{H}^T\mathbf{\Lambda}^{-1}\mathbf{H} \mid > \mid \prod_i \phi'(\hat{x}_i^{MAP}) \mid$, and the equation can then be solved by iterating over $\hat{\mathbf{x}}$ rather than using the slower gradient ascent. For Gaussian sources with unit covariance, $\phi(\mathbf{x}) = \mathbf{x}$ and the MAP estimator reduces to the ordinary FA one $\boldsymbol{\rho}(\mathbf{y})$ (34).

5 IFA: Simulation Results

Here we demonstrate the performance of our EM algorithm algorithm for IFA on mixtures of sources corrupted by Gaussian noise at different intensities. We used 5sec-long speech and music signals obtained from commercial CD's at the original sampling rate of 44.1kHz, that were down-sampled to $f_s = 8.82$kHz, resulting in $T = 44100$ sample points. These signals are characterized by peaky unimodal densities, as shown in Figure 2 (right). We also used synthetic signals obtained by a random number generator. These signals had arbitrary densities, two examples of which are shown in Figure 2 (left, middle).

All signals were scaled to have unit variance and mixed by a random $L' \times L$ mixing matrix $\mathbf{H}^0$ with varying number of sensors L'. L' white Gaussian signals with covariance matrix $\mathbf{\Lambda}^0$ were added to these mixtures. Different noise levels were used (see below). The learning rules (29,30) were iterated in batch mode, starting from random parameter values.

In all our experiments, we modeled each source density by a $n_i = 3$-state MOG, which provided a sufficiently accurate description of the signals we used, as Figure 2 (dashed and dotted lines) shows. In principle, prior knowledge of the source densities can be exploited by freezing the source parameters at the values corresponding to a MOG fit to their densities, and learning only the mixing matrix and noise

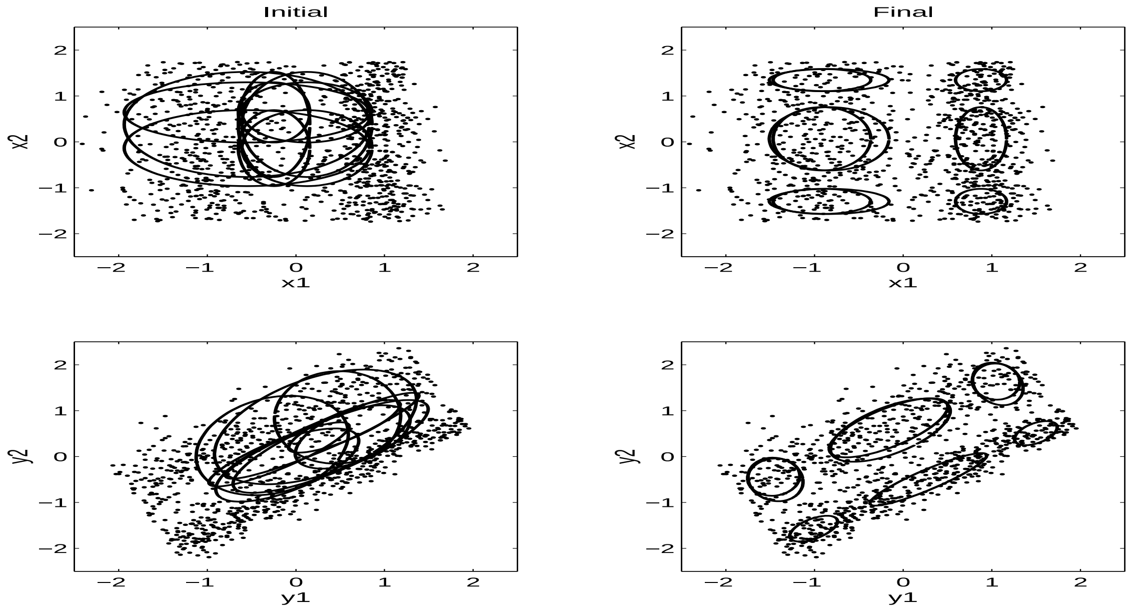

Figure 3: IFA learns a co-adaptive MOG model of the data. Top: the joint density of sources x_1, x_2 (dots) whose individual densities are shown in Figure 2. Bottom: the observed sensor density (dots) resulting from a linear 2×2 mixing of the sources contaminated by low noise. The MOG source model (11) is represented by ellipsoids centered at the means $\boldsymbol{\mu}_{\mathbf{q}}$ of the source states; same for the corresponding MOG sensor model (16,17). Note that the mixing affects a rigid rotation and scaling of the states. Starting from random source parameters (left), as well as random mixing matrix noise covariance, IFA learns their actual values (right).

covariance, which would result in faster convergence. However, we allowed the source parameters to adapt as well, starting from random values. Learning the source densities is illustrated in Figure 3.

Figure 4 (top, solid lines) shows the convergence of the estimated mixing matrix $\mathbf{H}$ towards the true one $\mathbf{H}^0$, for $L' = 3, 8$ mixtures of the $L = 3$ sources whose densities are histogrammed in Figure 2. Plotted are the matrix elements of the product

$$\mathbf{J} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{H}^0 \; . \tag{40}$$

Notice that for the correct estimate $\mathbf{H} = \mathbf{H}^0$, $\mathbf{J}$ becomes the unit matrix $\mathbf{I}$. Recall that the effect of source scaling is eliminated by (31); to prevent possible source permutations from affecting this measure, we permuted the columns of $\mathbf{H}$ such that the largest element (in absolute value) in column i of $\mathbf{J}$ would be J_{ii}. Indeed, this product is shown to converge to $\mathbf{I}$ in both cases.

To observe the convergence of the estimated noise covariance matrix $\boldsymbol{\Lambda}$ towards the true one $\boldsymbol{\Lambda}^0$, we measured the KL distance between the corresponding noise densities. Since both densities are Gaussian (see (5)), it is easy to calculate this distance analytically:

$$K_n = \int d\mathbf{u}\, \mathcal{G}(\mathbf{u}, \boldsymbol{\Lambda}^0)\, \log \frac{\mathcal{G}(\mathbf{u}, \boldsymbol{\Lambda}^0)}{\mathcal{G}(\mathbf{u}, \boldsymbol{\Lambda})} = \frac{1}{2}\mathrm{Tr}\, \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}^0 - \frac{L'}{2} - \frac{1}{2}\log|\det \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}^0\,| \; . \tag{41}$$

We recall that the KL distance is always non-negative; notice from (41) that $K_n = 0$ when $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^0$. Differentiating with respect to $\boldsymbol{\Lambda}$ shows that this is the only minimum point. As shown in Figure 4 (bottom, dashed line), K_n approaches zero in both cases.

The convergence of the estimated source densities $p(x_i)$ (8) was quantified by measuring their KL distance from the true densities $p^0(x_i)$. For this purpose, we first fitted a MOG model, $p^0(x_i) = \sum_{q_i} w^0_{i,q_i}\mathcal{G}(x_i - \mu^0_{i,q_i}, \nu^0_{i,q_i})$, to each source i and obtained the parameters w^0_{i,q_i}, μ^0_{i,q_i}, ν^0_{i,q_i} for $q_i = 1, 2, 3$. The KL distance

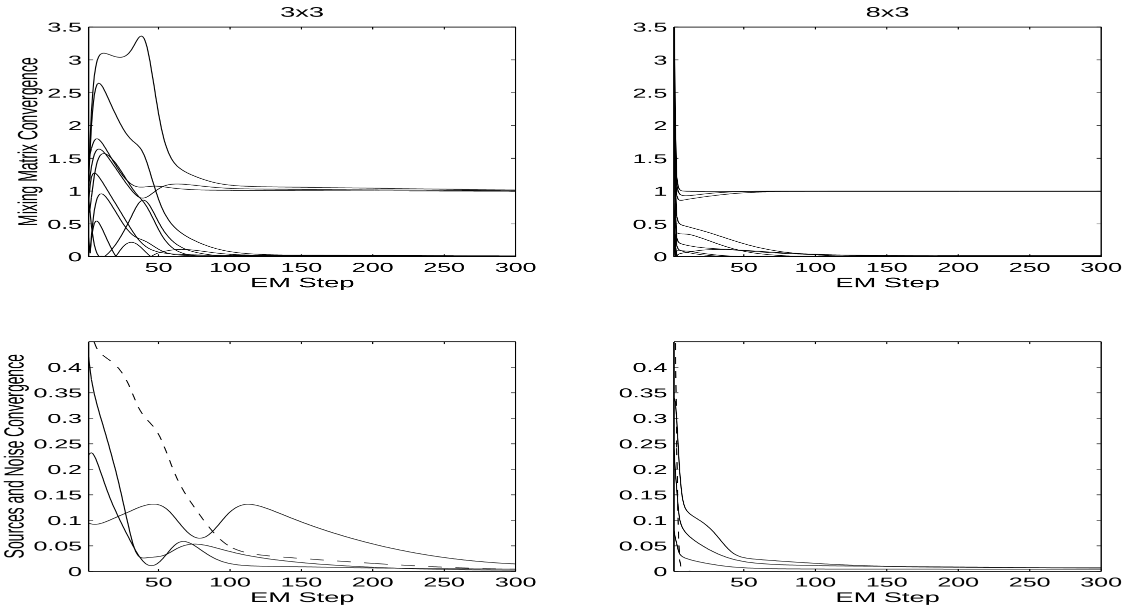

Figure 4: Top: convergence of the mixing matrix $\mathbf{H}$ with $L = 3$ sources, for $L' = 3$ (left) and $L' = 8$ (right) sensors and SNR = 5dB. Plotted are the matrix elements of $\mathbf{J}$ (40) (solid lines) against the EM step number. Bottom: convergence of the noise and source densities. Plotted are the KL distance K_n (41) between the estimated and true noise densities (dashed line), and the KL distances K_i (42) between the estimated source densities $p(x_i)$ and the true ones (solid lines).

at each EM step was then estimated via

$$K_i = \int dx_i \; p^0(x_i) \; \log \frac{p^0(x_i)}{p(x_i)} \approx \sum_{t=1}^{T} \log \frac{p^0(x_i^{(t)})}{p(x_i^{(t)})} \,, \tag{42}$$

where $p(x_i)$ was computed using the parameters values w_{i,q_i}, μ_{i,q_i}, ν_{i,q_i} obtained by IFA at that step; $x_i^{(t)}$ denotes the value of source i at time point t. Figure 4 (bottom, solid lines) shows the convergence of K_i towards zero for $L' = 3, 8$ sensors.

Figure 2 illustrates the accuracy of the source densities $p(x_i)$ learned by IFA. The histogram of the three sources used in this experiment is compared to its MOG description, obtained by adding up the corresponding 3 weighted Gaussians using the final IFA estimates of their parameters. The agreement is very good, demonstrating that the IFA algorithm successfully learned the source densities.

Figure 5 examines more closely the precision of the IFA estimates as the noise level increases. The mixing matrix error $\epsilon_{\mathbf{H}}$ quantifies the distance of the final value of $\mathbf{J}$ (40) from $\mathbf{I}$; we define it as the mean square non-diagonal elements of $\mathbf{J}$ normalized by its mean square diagonal elements:

$$\epsilon_{\mathbf{H}} = \left(\frac{1}{L^2 - L} \sum_{i \neq j = 1}^{L} J_{ij}^2 \right) \left(\frac{1}{L} \sum_{i=1}^{L} J_{ii}^2 \right)^{-1} \,. \tag{43}$$

The signal-to-noise ratio (SNR) is obtained by noting that the signal level in sensor i is $E(\sum_j H_{ij}^0 x_j)^2 = \sum_j (H_{ij}^0)^2$ (recall that $E\mathbf{x}\mathbf{x}^T = \mathbf{I}$), and the corresponding noise level is $Eu_i^2 = \Lambda_{ii}^0$. Averaging over the sensors, we get

$$\mathrm{SNR} = \frac{1}{L'} \sum_{i=1}^{L'} \left[\sum_{j=1}^{L} (H_{ij}^0)^2 \right] / \Lambda_{ii}^0 \,. \tag{44}$$

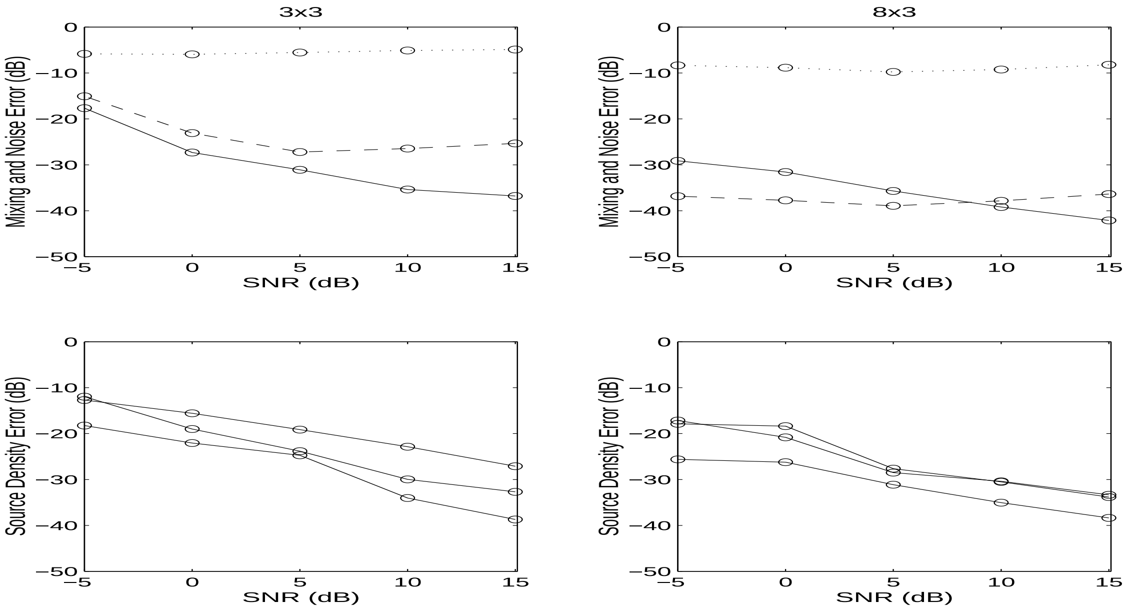

Figure 5: Top: estimate errors of the mixing matrix, $\epsilon_{\mathbf{H}}$ (43) (solid line), and noise covariance, K_n (41) (dashed line), against the signal-to-noise ratio (44), for $L' = 3$ (left) and $L' = 8$ (right). For reference, the errors of the ICA estimate of the mixing matrix (dotted line) are also plotted. Bottom: estimate errors K_i (42) of the source densities.

We plot the mixing matrix error against the SNR in Figure 5 (top, solid line), both measured in dB (i.e., $10 \log_{10} \epsilon_{\mathbf{H}}$ vs. $10 \log_{10} \mathrm{SNR}$), for $L' = 3, 8$ sensors. For reference, we also plot the error of the ICA (Bell and Sejnowski 1995) estimate of the mixing matrix (top, dotted line). Since ICA is formulated for the square ($L' = L$) noiseless case, we employed a two-step procedure: (i) the first L principal components (PC's) $\mathbf{y}_1 = \mathbf{P}_1^T \mathbf{y}$ of the sensor data $\mathbf{y}$ are obtained; (ii) ICA is applied to yield $\hat{\mathbf{x}}^{ICA} = \mathbf{G}\mathbf{y}_1$. The resulting estimate of the mixing matrix is then $\mathbf{H}^{ICA} = \mathbf{P}_1 \mathbf{G}^{-1}$. Notice that this procedure is exact for zero noise, since in that case the first L PC's are the only non-zero ones and the problem reduces to one of square noiseless mixing, described by $\mathbf{y}_1 = \mathbf{P}_1 \mathbf{H} \mathbf{x}$ (see also the discussion at the end of Section 7.1).

Also plotted in Figure 5 is the error in the estimated noise covariance $\mathbf{\Lambda}$ (top, dashed line), given by the KL distance K_n (41) for the final value of $\mathbf{\Lambda}$. (Measuring the KL distance in dB is suggested by its mean-square-error interpretation (20)). Figure 5 (bottom) shows the estimate errors of the source densities $p(x_i)$, given by their KL distance (42) from the true densities after the IFA was completed.

As expected, these errors decrease with increasing SNR and also with increasing L'. The noise error K_n forms an exception, however, by showing a slight increase with the SNR, reflecting the fact that a lower noise level is harder to estimate to a given precision. In general, convergence is faster for larger L'.

We conclude that the estimation errors for the IF model parameters are quite small, usually falling in the range of $20 - 40$dB and never larger than 15dB as long as the noise level is not higher than the signal level (SNR $\geq$ 0dB). Similar results were obtained in other simulations we performed. The small values of the estimate errors suggest that those errors originate from the finite sample size, rather than from convergence to undesired local minima.

Finally, we studied how the noise level affects the separation performance, as measured by the quality of source reconstructions obtained from $\hat{\mathbf{x}}^{LMS}$ (36) and $\hat{\mathbf{x}}^{MAP}$ (38). We quantified it by the mean square reconstruction error ϵ^{rec}, which measures how close the reconstructed sources are to the original ones. This error is composed of two components, one arising from the presence of noise and the other from interference of the other sources ('cross-talk'); the additional component arising from IF parameter estimation errors is

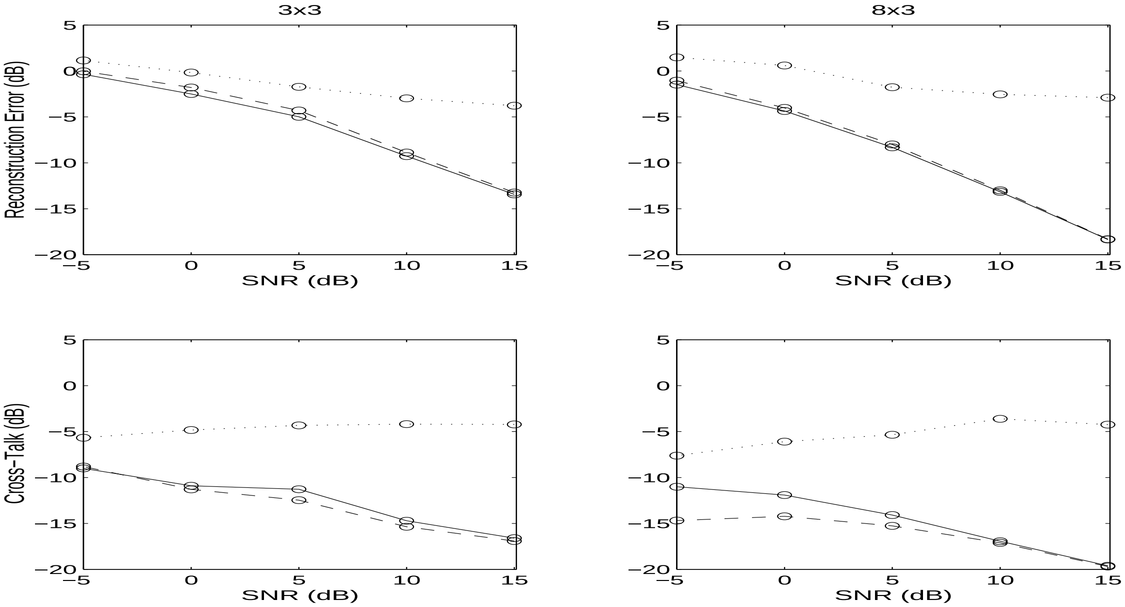

Figure 6: Source reconstruction quality with $L = 3$ sources for $L' = 3$ (left) and $L' = 8$ (right) sensors. Plotted are the reconstruction error ϵ^{rec} (top) and the cross-talk level ϵ^{xtalk} (45) (bottom) vs. signal-to-noise ratio, for the LMS (solid lines), MAP (dashed lines), and ICA (dotted lines) estimators.

negligible in comparison. The amount of cross-talk is measured by ϵ^{xtalk}:

$$\epsilon^{rec} = \frac{1}{L}\sum_{i=1}^{L} E(\hat{x}_i - x_i)^2 \ , \qquad \epsilon^{xtalk} = \frac{1}{L^2 - L}\sum_{i \neq j=1}^{L} \mid E\hat{x}_i x_j \mid \ . \tag{45}$$

Note that for zero noise and perfect separation ($\hat{x}_i = x_i$), both quantities approach zero in the infinite sample limit.

The reconstruction error (which is normalized since $Ex_i^2 = 1$) and the cross-talk level are plotted in Figure 6 against the SNR for both the LMS (solid lines) and MAP (dashed lines) source estimators. For reference, we also plot the ICA results (dotted lines). As expected, ϵ^{rec} and ϵ^{xtalk} decrease with increasing SNR and are significantly higher for ICA. Notice that the LMS reconstruction error is always lower than the MAP one, since it is derived by demanding that it minimizes precisely ϵ^{rec}. In contrast, the MAP estimator has a lower cross-talk level.

6 IFA with Many Sources: The Factorized Variational Approximation

Whereas the EM algorithm (29,30) is exact and all the required calculations can be done analytically, it becomes intractable as the number of sources in the IF model increases. This is because the conditional means computed in the E-step (84–86) involve summing over all $\prod_i n_i$ possible configurations of the source states, i.e., $\sum_{\mathbf{q}} = \sum_{q_1}\sum_{q_2}\cdots\sum_{q_L}$, whose number grows exponentially with the number of sources. As long as we focus on separating a small number L of sources (treating the rest as noise) and describe each source by a small number n_i of states, the E-step is tractable, but separating, for example, $L = 13$ sources with $n_i = 3$ states each would involve $3^{13} \approx 1.6 \times 10^6$-element sums at each iteration.

The intractability of exact learning is, of course, not a problem unique to the IF model but is shared by many probabilistic models. In general, approximations must be made. A suitable starting point for approximations is the function $\mathcal{F}$ (23), which is bounded from below by the exact error $\mathcal{E}$ for an arbitrary p'.

The density p' is a posterior over the hidden variables of our generative model, given the values of the visible variables. The root of the intractability of EM is the choice (24) of p' as the *exact* posterior, which is derived from p via Bayes' rule and is parametrized by the generative parameters W. Several approximation schemes were proposed (Hinton et al. 1995; Dayan et al. 1995; Saul and Jordan 1995; Saul et al. 1996; Ghahramani and Jordan 1997) where p' has a form that generally differs from that of the exact posterior and has its own set of parameters $\boldsymbol{\tau}$, which are learned separately from W by an appropriate procedure. Of crucial significance is the functional form of p', which should be chosen so as to make the E-step tractable, while still providing a reasonable approximation of the exact posterior. The parameters $\boldsymbol{\tau}$ are then optimized to minimize the distance between p' and the exact posterior.

In the case of the IF model, we consider the function

$$\mathcal{F}(\boldsymbol{\tau}, W) = -\sum_{\mathbf{q}} \int d\mathbf{x}\; p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, \boldsymbol{\tau}) \;\log \frac{p(\mathbf{q}, \mathbf{x}, \mathbf{y} \mid W)}{p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, \boldsymbol{\tau})} \geq \mathcal{E}(W) \;, \tag{46}$$

where averaging over the data is implied. We shall use a variational approach, first formulated in the context of feedforward probabilistic models by Saul and Jordan (1995). Given the chosen form of the posterior p' (see below), $\mathcal{F}$ (46) will be minimized iteratively with respect to both W and the variational parameters $\boldsymbol{\tau}$.

This minimization leads to the following approximate EM algorithm for IFA, which we derive in the remaining part of this section. Assume that the previous iteration produced W'. The E-step of the current iteration consists of determining the values of $\boldsymbol{\tau}$ in terms of W' by solving a pair of coupled 'mean-field' equations (53,54). It is straightforward to show that this step minimizes the KL distance between the variational and exact posteriors, $KL[p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, \boldsymbol{\tau}), p(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, W')]$. In fact, this distance equals the difference $\mathcal{F}(\boldsymbol{\tau}, W') - \mathcal{E}(W')$. Hence, this E-step approximates the exact one in which this distance actually vanishes.

Once the variational parameters have been determined, the new generative parameters W are obtained in the M-step using (29,30), where the conditional source means can be readily computed in terms of $\boldsymbol{\tau}$.

6.1 Factorized Posterior

We begin with the observation that, whereas the sources in the IF model are independent, the sources *conditioned on a data vector* are correlated. This is clear from the fact that the conditional source correlation matrix $\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{q}, \mathbf{y} \rangle$ (83) is non-diagonal. More generally, the joint source posterior density $p(\mathbf{q}, \mathbf{x} \mid \mathbf{y})$ given by (80,82) does not factorize, i.e., cannot be expressed as a product over the posterior densities of the individual sources.

In the factorized variational approximation we assume that even when conditioned on a data vector, the sources are independent. Our approximate posterior source density is defined as follows. Given a data vector $\mathbf{y}$, the source x_i at state q_i is described by a Gaussian distribution with a $\mathbf{y}$-dependent mean ψ_{i,q_i} and variance ξ_{i,q_i}, weighted by a mixing proportion κ_{i,q_i}. The posterior is defined simply by the product

$$p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, \boldsymbol{\tau}) = \prod_{i=1}^{L} \kappa_{i,q_i}(\mathbf{y})\; \mathcal{G}\left[x_i - \psi_{i,q_i}(\mathbf{y}), \xi_{i,q_i}\right] \;, \qquad \tau_i = \{\kappa_{i,q_i}, \psi_{i,q_i}, \xi_{i,q_i}\} \;. \tag{47}$$

As alluded to by (47), the variances ξ_{i,q_i} will turn out to be $\mathbf{y}$-independent.

To gain some insight into the approximation (47), notice first that it implies a MOG form for the posterior of x_i,

$$p'(x_i \mid \mathbf{y}, \tau_i) = \sum_{q_i=1}^{n_i} \kappa_{i,q_i}(\mathbf{y})\; \mathcal{G}(x_i - \psi_{i,q_i}(\mathbf{y}), \xi_{i,q_i}) \;, \tag{48}$$

which is in complete analogy with its *prior* (8). Thus, conditioning the sources on the data is approximated simply by allowing the variational parameters to depend on $\mathbf{y}$. Next, compare (47) to the exact posterior

$p(\mathbf{q}, \mathbf{x} \mid \mathbf{y}, W)$ (80,82). The latter also implies a MOG form for $p(x_i \mid \mathbf{y})$, but one that differs from (47); and in contrast with our approximate posterior, the exact one implies a MOG form for $p(x_i \mid q_i, \mathbf{y})$ as well, reflecting the fact that the source states and signals are all correlated given the data.

Therefore, the approximation (47) can be viewed as the result of shifting the source prior towards the true posterior for each data vector, with the variational parameters $\boldsymbol{\tau}$ assuming the shifted values of the source parameters $\boldsymbol{\theta}$. Whereas this shift, of course, cannot capture correlations between the sources, it can be optimized to allow (47) to best approximate the true posterior while maintaining a factorized form. A procedure for determining the optimal values of $\boldsymbol{\tau}$ is derived in the next section.

The factorized posterior (47) is advantageous since it facilitates performing the E-step calculations in polynomial time. Once the variational parameters have been determined, the data-conditioned mean and covariance of the sources, required for the EM learning rule (29) are

$$\begin{aligned}
\langle x_i \mid \mathbf{y} \rangle &= \sum_{q_i=1}^{n_i} \kappa_{i,q_i} \psi_{i,q_i} \;, \\
\langle x_i^2 \mid \mathbf{y} \rangle &= \sum_{q_i=1}^{n_i} \kappa_{i,q_i} (\psi_{i,q_i}^2 + \xi_{i,q_i}) \;, \quad \langle x_i x_{j \neq i} \mid \mathbf{y} \rangle = \sum_{q_i q_j} \kappa_{i,q_i} \kappa_{j,q_j} \psi_{i,q_i} \psi_{j,q_j} \;,
\end{aligned} \tag{49}$$

whereas those required for the rules (30), which are further conditioned on the source states, are given by

$$p(q_i \mid \mathbf{y}) = \kappa_{i,q_i} \;, \quad \langle x_i \mid q_i, \mathbf{y} \rangle = \psi_{i,q_i} \;, \quad \langle x_i^2 \mid q_i, \mathbf{y} \rangle = \psi_{i,q_i}^2 + \xi_{i,q_i} \;. \tag{50}$$

Recovering the sources. In Section 4, the LMS (35,36) and MAP (37,38) source estimators were given for exact IFA. Notice that, being part of the E-step, computing the LMS estimator exactly quickly becomes intractable as the number of sources increases. In the variational approximation it is replaced by $\hat{x}_i^{LMS}(\mathbf{y}) = \langle x_i \mid \mathbf{y} \rangle$ (49), which depends on the variational parameters and avoids summing over all source state configurations. In contrast, the MAP estimator remains unchanged (but the parameters W on which it depends are now learned by variational IFA); note that its computational cost is only weakly dependent on L.

6.2 Mean-Field Equations

For fixed $\boldsymbol{\tau}$, the learning rules for W (29,30) follow from $\mathcal{F}(\boldsymbol{\tau}, W)$ (46) by solving the equations $\partial \mathcal{F} / \partial W = 0$. These equations are linear, as is evident from the gradients given in Appendix A.2, and their solution $W = W(\boldsymbol{\tau})$ is given in closed form.

The learning rules for $\boldsymbol{\tau}$ are similarly derived by fixing $W = W'$ and solving $\partial \mathcal{F} / \partial \boldsymbol{\tau} = 0$. Unfortunately, examining the gradients given in Appendix B shows that these equations are non-linear and must be solved numerically. We choose to find their solution $\boldsymbol{\tau} = \boldsymbol{\tau}(W')$ by iteration.

Define the $L \times L$ matrix $\bar{\mathbf{H}}$ by

$$\bar{\mathbf{H}} = \mathbf{H}^T \boldsymbol{\Lambda}^{-1} \mathbf{H} \;. \tag{51}$$

The equation for the variances ξ_{i,q_i} does not involve $\mathbf{y}$ and can easily be solved:

$$\xi_{i,q_i} = (\bar{H}_{ii} + \frac{1}{\nu_{i,q_i}})^{-1} \;. \tag{52}$$

The means $\psi_{i,q_i}(\mathbf{y})$ and mixing proportions $\kappa_{i,q_i}(\mathbf{y})$ are obtained by iterating the following mean-field equations for each data vector $\mathbf{y}$:

$$\sum_{j \neq i} \sum_{q_j=1}^{n_j} \bar{H}_{ij} \kappa_{j,q_j} \psi_{j,q_j} + \frac{1}{\xi_{i,q_i}} \psi_{i,q_i} = (\mathbf{H}^T \boldsymbol{\Lambda}^{-1} \mathbf{y})_i + \frac{\mu_{i,q_i}}{\nu_{i,q_i}} \;, \tag{53}$$

$$\log \kappa_{i,q_i} = \log w_{i,q_i} + \frac{1}{2}\left(\log \xi^2_{i,q_i} + \frac{\psi^2_{i,q_i}}{\xi_{i,q_i}}\right) - \frac{1}{2}\left(\log \nu^2_{i,q_i} + \frac{\mu^2_{i,q_i}}{\nu_{i,q_i}}\right) + z_i \equiv \alpha_{i,q_i} + z_i \ , \tag{54}$$

where the z_i are Lagrange multipliers that enforce the normalization conditions $\sum_{q_i} \kappa_{i,q_i} = 1$. Note that Eq. (53) depends non-linearly on $\mathbf{y}$ due to the non-linear $\mathbf{y}$-dependence of κ_{i,q_i}.

To solve these equations, we first initialize $\kappa_{i,q_i} = w_{i,q_i}$. Eq. (53) is a linear $(\sum_i n_i) \times (\sum_i n_i)$ system and can be solved for ψ_{i,q_i} using standard methods. The new κ_{i,q_i} are then obtained from (54) via

$$\kappa_{i,q_i} = \frac{e^{\alpha_{i,q_i}}}{\sum_{q'_i} e^{\alpha_{i,q'_i}}} \ . \tag{55}$$

These values are substituted back into (53) and the procedure is repeated until convergence.

Data-independent approximation. A simpler approximation results from setting $\kappa_{i,q_i}(\mathbf{y}) = w_{i,q_i}$ for all data vectors $\mathbf{y}$. The means ψ_{i,q_i} can then be obtained from (53) in a single iteration for all data vectors at once, since this equation becomes linear in $\mathbf{y}$. This approximation is much less expensive computationally, with a corresponding reduction in accuracy as shown below.

6.3 Variational IFA: Simulation Results

Whereas the factorized form of the true posterior (47) and its data-independent simplification are not exact, the mean-field equations optimize the variational parameters $\boldsymbol{\tau}$ to make the approximate posterior as accurate as possible. Here we assess the quality of this approximation.

First, we studied the accuracy of the approximate error function $\mathcal{F}$ (46). For this purpose we considered a small data set with 100 $L' \times 1$ vectors $\mathbf{y}$ generated independently from a Gaussian distribution. The approximate log-likelihood $-\mathcal{F}(\boldsymbol{\tau}, W)$ of these data were compared to the exact log-likelihood $-\mathcal{E}(W)$ (46), with respect to 5000 IF models with random parameters W. Each realization of W was obtained by sampling the parameters from uniform densities defined over the appropriate intervals, followed by scaling the source parameters according to (31). In the case of the mixing proportions, $\bar{w}_{i,q_i}$ were sampled and w_{i,q_i} were obtained via (89). $n_i = 3$-state MOG densities were used. The relative error in the log-likelihood

$$\epsilon^{like} = \frac{\mathcal{F}(\boldsymbol{\tau}, W)}{\mathcal{E}(W)} - 1 \tag{56}$$

was then computed for the factorized and data-independent approximations. Its histogram is displayed in Figure 7 for the case $L' = 5$, with $L = 3$ (left) and $L = 4$ (middle) sources.

In these examples, as well as in other simulations we performed, the mean error in the factorized approximation is under 3%. The data-independent approximation, as expected, is less accurate and increases the mean error above 8%.

Next, we investigated whether the variational IFA algorithm learns appropriate values for the IF model parameters W. The answer is quantified below in terms of the resulting reconstruction error. 5sec-long source signals, sampled from different densities (like those displayed in Figure 2) at a rate of 8.82kHz, were generated. Noisy linear mixtures of these sources were used as data for the exact IFA algorithm and to its approximations. After learning, the source signals were reconstructed from the data by the LMS source estimator (see the discussion at the end of Section 6.1). For each data vector, the reconstruction error ϵ^{rec} (45) was computed. The histograms of $10 \log_{10} \epsilon^{rec}$ (dB units) for the exact IFA and its approximations in a case with $L' = 5, L = 4$, SNR=10dB are displayed in Figure 7 (right). For reference, the ICA error histogram in this case is also plotted.

Note that the variational histogram is very close to the exact one, whereas the data-independent histogram has a larger mean error. The ICA mean error is the largest, consistent with the results of Figure 6 (top).

We conclude that the factorized variational approximation of IFA is quite accurate. Of course, the real test is in its application to cases with large numbers of sources where exact IFA can no longer be used. In addition, other variational approximations can also be defined. A thorough assessment of the factorial and

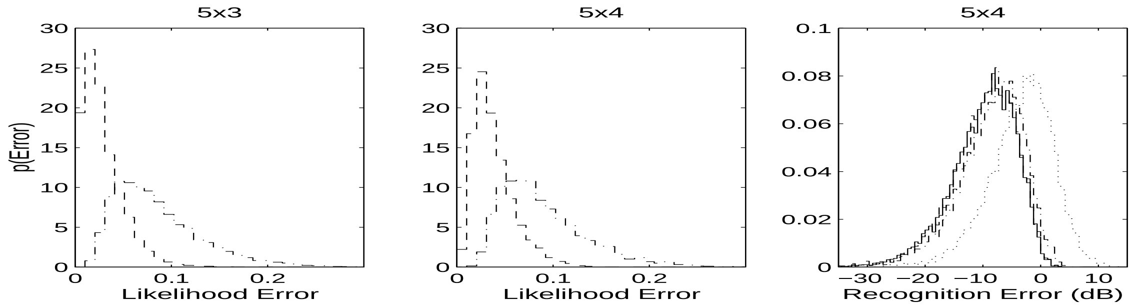

Figure 7: Left, middle: histogram of the relative error in the log-likelihood ϵ^{like} (56) of 100 random data vectors, for the factorized variational approximation (dashed line; mean error = 0.021 (left), 0.025 (middle)) and its data-independent simplification (dashed-dotted line; mean = 0.082, 0.084). The likelihoods were computed with respect to 5000 random IF model parameters with $L' = 5$ sensors and $L = 3$ (left) and $L = 4$ (middle) sources. Right: histogram of the reconstruction error ϵ^{rec} (45) at SNR=10dB for exact IFA (solid line; mean = −10.2dB), the factorized (dashed line; mean = −10.1dB) and data-independent (dashed-dotted line; mean = −8.9dB) variational approximations, and ICA (dotted line; mean = −3.66dB). The LMS source estimator was used.

other variational approximations and their applications is somewhat beyond the scope of the present paper and will be published separately.

7 Noiseless IFA

We now consider the IF model (4) in the noiseless case $\mathbf{\Lambda} = \mathbf{0}$. Here the sensor data depend deterministically on the sources,

$$\mathbf{y} = \mathbf{H}\mathbf{x} \ , \tag{57}$$

hence once the mixing matrix $\mathbf{H}$ is found, the latter can be recovered exactly (rather than estimated) from the observed data using the pseudo-inverse of $\mathbf{H}$ via

$$\mathbf{x} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \ , \tag{58}$$

which reduces to $\mathbf{x} = \mathbf{H}^{-1}\mathbf{y}$ for square invertible mixing. Hence, vanishing noise level results in a linear source estimator which is independent of the source parameters.

One might expect that our EM algorithm (29,30) for the noisy case can also be applied to noiseless mixing, with the only consequence being that the noise covariance $\mathbf{\Lambda}$ would acquire very small values. This, however, is not the case, as we shall show below. It turns out that in the zero-noise limit, that algorithm actually performs *principal component analysis* (PCA); consequently, for low noise, convergence from the PCA to IFA solution is very slow. The root of the problem is that in the noiseless case we have only one type of 'missing data', namely the source states $\mathbf{q}$; the source signals $\mathbf{x}$ are no longer missing, being given directly by the observed sensors via (58). We shall therefore proceed to derive an EM algorithm specifically for this case. This algorithm will turn out to be a powerful extension of Bell and Sejnowski's (1995) ICA algorithm.

7.1 An Expectation-Maximization Algorithm

We first focus on the square invertible mixing ($L' = L$, rank $\mathbf{H} = L$), and write (4) as

$$\mathbf{x} = \mathbf{G}\mathbf{y} \ , \tag{59}$$

where the unmixing (separating) matrix $\mathbf{G}$ is given by $\mathbf{H}^{-1}$ with its columns possibly scaled and permuted.

Unlike the noisy case, here there is only one type of 'missing data', namely the source states $\mathbf{q}$, since the stochastic dependence of the sensor data on the sources becomes deterministic. Hence, the conditional density $p(\mathbf{y} \mid \mathbf{x})$ (14) must be replaced by $p(\mathbf{y}) = \mid \det \mathbf{G} \mid p(\mathbf{x})$ as implied by (59). Together with the factorial MOG model for the sources $\mathbf{x}$ (11), the error function (23) becomes

$$\begin{aligned}\mathcal{E}(W) &= -\log p(\mathbf{y} \mid W) = -\log \mid \det \mathbf{G} \mid - \log p(\mathbf{x} \mid W) \\ &\leq -\log \mid \det \mathbf{G} \mid - \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{x}, W') \, \log \frac{p(\mathbf{x}, \mathbf{q} \mid W)}{p(\mathbf{q} \mid \mathbf{x}, W')} \, . \end{aligned} \tag{60}$$

As in the noisy case (25), we have obtained an approximated error $\mathcal{F}(W', W)$ that is bounded from below by the true error, and is given by a sum over the individual layer contributions (see Figure 1),

$$\mathcal{E}(W) \leq \mathcal{F}(W', W) = \mathcal{F}_V + \mathcal{F}_B + \mathcal{F}_T + \mathcal{F}_H \, . \tag{61}$$

Here, however, the contributions of the visible and bottom hidden layers both depend on the visible layer parameters $\mathbf{G}$,

$$\begin{aligned}\mathcal{F}_V(W', \mathbf{G}) &= -\log \mid \det \mathbf{G} \mid \, , \\ \mathcal{F}_B(W', \mathbf{G}, \{\mu_{i,q_i}, \nu_{i,q_i}\}) &= -\sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid x_i, W') \log p(x_i \mid q_i) \, , \\ \mathcal{F}_T(W', \mathbf{G}, \{w_{i,q_i}\}) &= -\sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid x_i, W') \log p(q_i) \, , \end{aligned} \tag{62}$$

whereas the top layer contribution remains separated (compare with (26), noting that $p(\mathbf{q} \mid \mathbf{y}) = p(\mathbf{q} \mid \mathbf{x})$ due to (59)). The entropy term

$$\mathcal{F}_H(W') = \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{x}, W') \, \log p(\mathbf{q} \mid \mathbf{x}, W') \tag{63}$$

is W-independent. We point out that the complete form of expressions (62) includes replacing $\mathbf{x}$ by $\mathbf{Gy}$ and averaging over the observed $\mathbf{y}$.

The EM learning algorithm for the IF model parameters is derived in Appendix C. A difficulty arises from the fact that the M-step equation $\partial\mathcal{F}/\partial\mathbf{G} = 0$, whose solution is the new value $\mathbf{G}$ in terms of the parameters W' obtained at the previous EM step, is non-linear and cannot be solved analytically. Instead we solve it iteratively, so that each EM step $W' \to W$ is composed of a sequence of iterations on W with W' held fixed.

The noiseless IFA learning rule for the separating matrix is given by

$$\delta\mathbf{G} = \eta\mathbf{G} - \eta E \phi'(\mathbf{x})\mathbf{x}^T\mathbf{G} \, , \tag{64}$$

where $\eta > 0$ determines the learning rate and its value should be set empirically. $\phi'(\mathbf{x})$ is an $L \times 1$ vector which depends on the posterior $p'(q_i \mid x_i) \equiv p(q_i \mid x_i, W')$ (95) computed using the parameters from the previous iteration; its i-th coordinate is given by a weighted sum over the states q_i of source i,

$$\phi'(x_i) = \sum_{q_i=1}^{n_i} p'(q_i \mid x_i) \frac{x_i - \mu_{i,q_i}}{\nu_{i,q_i}} \, . \tag{65}$$

The rules for the source MOG parameters are

$$\begin{aligned}\mu_{i,q_i} &= \frac{Ep'(q_i \mid x_i)x_i}{Ep'(q_i \mid x_i)} \, , \\ \nu_{i,q_i} &= \frac{Ep'(q_i \mid x_i)x_i^2}{Ep'(q_i \mid x_i)} - \mu_{i,q_i}^2 \, , \\ w_{i,q_i} &= Ep'(q_i \mid x_i) \, . \end{aligned} \tag{66}$$

Recall that $\mathbf{x}$ is linearly related to $\mathbf{y}$ and the operator E averages over the observed $\mathbf{y}$.

The noiseless IFA learning rules (64–66) should be used as follows. Having obtained the parameters W' in the previous EM step, the new step starts with computing the posterior $p'(q_i \mid x_i)$ and setting the initial values of the new parameters W to W', except for w_{i,q_i} which can be set to its final value $Ep'(q_i \mid x_i)$. Then a sequence of iterations begins, where each iteration consists of
(i) computing the sources by $\mathbf{x} = \mathbf{G}\mathbf{y}$ using the current $\mathbf{G}$;
(ii) computing the new μ_{i,q_i}, ν_{i,q_i} from (66) using the sources obtained in (i);
(iii) computing the new $\mathbf{G}$ from (64,65) using the sources obtained in (i) and the means and variances obtained in (ii).

The iterations continue until some convergence criterion is satisfied; note that during this process, both $\mathbf{x}$ and W change but $p'(q_i \mid x_i)$ are frozen. Achieving convergence completes the current EM step; the next step starts with updating those posteriors.

We recognize the learning rules for the source densities (66) as precisely the standard EM rules for learning a separate MOG model for each source i, shown on the right column of (32). Hence, our noiseless IFA algorithm combines separating the sources, by learning $\mathbf{G}$ using the rule (64), with simultaneously learning their densities by EM-MOG. These two processes are coupled by the priors $p'(q_i \mid x_i)$. We shall show in the next section that the two can decouple, and consequently the separating matrix rule (64) becomes Bell and Sejnowski's (1995) ICA rule, producing the algorithm shown schematically in Figure 8.

We also point out that the MOG learning rules for the noiseless case (66) can be obtained from those for the noisy case (30) by replacing the conditional source means $\langle x_i \mid q_i, \mathbf{y}\rangle$ by $x_i = \sum_j G_{ij} y_j$, and replacing the source state posteriors $p(q_i \mid \mathbf{y})$ by $p(q_i \mid x_i)$. Both changes arise from the vanishing noise level which makes the source-sensor dependence deterministic.

Scaling. As in the noisy case (31), noiseless IFA is augmented by the following scaling transformation at each iteration:

$$\begin{aligned} \sigma_i^2 &= \sum_{q_i=1}^{n_i} w_{i,q_i}(\nu_{i,q_i} + \mu_{i,q_i}^2) - (\sum_{q_i=1}^{n_i} w_{i,q_i}\mu_{i,q_i})^2 \;, \\ \mu_{i,q_i} &\to \frac{\mu_{i,q_i}}{\sigma_i} \;, \quad \nu_{i,q_i} \to \frac{\nu_{i,q_i}}{\sigma_i^2} \;, \quad G_{ij} \to \frac{1}{\sigma_i} G_{ij} \;. \end{aligned} \tag{67}$$

More sensors than sources. The noiseless IFA algorithm given above assumes that $\mathbf{H}$ is a square invertible $L \times L$ mixing matrix. The more general case of an $L' \times L$ mixing with $L' \geq L$ can be treated as follows.

We start with the observation that in this case, the $L' \times L'$ sensor covariance matrix $\mathbf{C_y} = E\mathbf{y}\mathbf{y}^T$ is of rank L. Let the columns of $\mathbf{P}$ contain the eigenvectors of $\mathbf{C_y}$, so that $\mathbf{P}^T\mathbf{C_y}\mathbf{P} = \mathbf{D}$ is diagonal. Then $\mathbf{P}^T\mathbf{y}$ are the L' principal components of the sensor data, and only L of them are non-zero. The latter are denoted by $\mathbf{y}_1 = \mathbf{P}_1^T\mathbf{y}$, where $\mathbf{P}_1$ is formed by those columns of $\mathbf{P}$ corresponding to non-zero eigenvalues.

The algorithm (64–66) should now be applied to $\mathbf{y}_1$ to find an $L \times L$ separating matrix, denoted $\mathbf{G}_1$. Finally, the $L \times L'$ separating matrix $\mathbf{G}$ required for recovering the sources from sensors via (59) is simply $\mathbf{G} = \mathbf{G}_1\mathbf{P}_1^T$.

It remains to find $\mathbf{P}_1$. This can be done using matrix diagonalization methods. Alternatively, observing that its columns are not required to be the first L eigenvectors of $\mathbf{C_y}$ but only to span the same subspace, the principal component analysis learning rule (74) (with $\mathbf{H}$ replaced by $\mathbf{P}_1$) may be used for this purpose.

7.2 Generalized EM and the Relation to Independent Component Analysis

Whereas the procedure described above for using the noiseless IFA rules (64–66) is a strictly EM algorithm (for a sufficiently small η), it is also possible to use them in a different manner. An alternative procedure can be defined by making either or both of the following changes: (i) complete each EM step and update the posteriors $p'(q_i \mid x_i)$ after some fixed number S of iterations, regardless of whether convergence has been

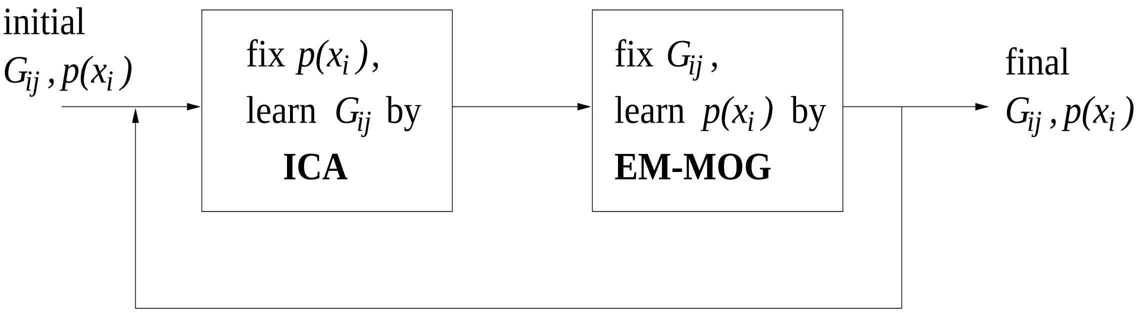

Figure 8: The Seesaw GEM algorithm for noiseless IFA.

achieved; (ii) for a given EM step, select some parameters from the set W and freeze them during that step, while updating the rest; the choice of frozen parameters may vary from one step to the next.

Any procedure that incorporates (i) or (ii) does not minimize the approximate error $\mathcal{F}$ at each M-step (unless S is sufficiently large), but merely reduces it. Of course, the EM convergence proof remains valid in this case. Such a procedure is termed a 'generalized EM' (GEM) algorithm (Dempster et al. 1977; Neal and Hinton 1998). Clearly, there are many possible GEM versions of noiseless IFA. Two particular versions are defined below:

Chase: obtained from the EM version simply by updating the posteriors at each iteration. Each GEM step consists of
(i) a single iteration of the separating matrix rule (64);
(ii) a single iteration of the MOG rules (66);
(iii) updating the posteriors $p'(q_i \mid x_i)$ using the new parameter values.
Hence, the source densities follow $\mathbf{G}$ step by step.

Seesaw: obtained by breaking the EM version into two phases and alternating between them:
(i) freeze the MOG parameters; each GEM step consists of a single iteration of the separating matrix rule (64), followed by updating the posteriors using the new value of $\mathbf{G}$;
(ii) freeze the separating matrix; each GEM step consists of a single iteration of the MOG rule (66), followed by updating the posteriors using the new values of the MOG parameters.
The sequence of steps in each phase terminates after making S steps or upon satisfying a convergence criterion. Hence, we switch back and forth between learning $\mathbf{G}$ and learning the source densities.

Both the Chase and Seesaw GEM algorithms were found to converge faster than the original EM one. Notice that both require updating the posteriors at each step; this operation is not computationally expensive since each source posterior $p(q_i \mid x_i)$ (95) is computed individually and requires summing only over its own n_i states, making the total cost linearly dependent on L. In our noisy IFA algorithm, in contrast, updating the source state posteriors $p(\mathbf{q} \mid \mathbf{y})$ (82) requires summing over the $\prod_i n_i$ collective source states $\mathbf{q}$, and the total cost is exponential in L.

We now show that Seesaw combines two well-known algorithms in an intuitively appealing manner. Since the source density learning rules (66) are the EM rules for fitting an MOG model to each source, as discussed in the previous section, the second phase of Seesaw is equivalent to EM-MOG. It will be shown below that its first phase is equivalent to Bell and Sejnowski's (1995) independent component analysis (ICA) algorithm, with their sigmoidal non-linearity replaced by a function related to our MOG source densities. Therefore, Seesaw amounts to learning G_{ij} by applying ICA to the observed sensors y_j while the densities $p(x_i)$ are kept fixed, then fixing G_{ij} and learning the new $p(x_i)$ by applying EM-MOG to the reconstructed sources $x_i = \sum_j G_{ij} y_j$, and repeat. This algorithm described schematically in Figure 8.

In the context of BSS, the noiseless IFA problem for an equal number of sensors and sources had already been formulated before as the problem of ICA by Comon (1994). An efficient ICA algorithm was first proposed by Bell and Sejnowski (1995) from an information-maximization viewpoint; it was soon observed (Mackay 1996; Pearlmutter and Parra 1997; Cardoso 1997) that this algorithm was, in fact, performing a maximum-likelihood (or, equivalently, minimum KL distance) estimation of the separating matrix using a generative model of linearly mixed sources with non-Gaussian densities. In ICA, these densities are fixed throughout.

The derivation of ICA, like that of our noiseless IFA algorithm, starts from the KL error function $\mathcal{E}(W)$ (60). However, rather than approximating it, ICA minimizes the exact error by the steepest descent method using its gradient $\partial\mathcal{E}/\partial\mathbf{G} = -(\mathbf{G}^T)^{-1} + \varphi(\mathbf{x})\mathbf{y}^T$, where $\varphi(\mathbf{x})$ is an $L \times 1$ vector whose i-th coordinate is related to the density $p(x_i)$ of source i via $\varphi(x_i) = -\partial \log p(x_i)/\partial x_i$. The separating matrix $\mathbf{G}$ is incremented at each iteration in the direction of the *relative* gradient (Cardoso and Laheld 1996; Amari et al. 1996; Mackay 1996) of $\mathcal{E}(W)$ by $\delta\mathbf{G} = -\eta(\partial\mathcal{E}/\partial\mathbf{G})\mathbf{G}^T\mathbf{G}$, resulting in the learning rule

$$\delta\mathbf{G} = \eta\mathbf{G} - \eta E\varphi(\mathbf{x})\mathbf{x}^T\mathbf{G} , \tag{68}$$

where the sources are computed from the sensors at each iteration via $\mathbf{x} = \mathbf{G}\mathbf{y}$.

Now, the ICA rule (68) has the form of our noiseless IFA separating matrix rule (64) with $\phi(x_i)$ (65) replaced by $\varphi(x_i)$ defined above. Moreover, whereas the original Bell and Sejnowski (1995) algorithm used the source densities $p(x_i) = \cosh^{-2}(x_i)$, it can be shown that using our MOG form for $p(x_i)$ (8) produces

$$\varphi(x_i) = \sum_{q_i=1}^{n_i} p(q_i \mid x_i)\frac{x_i - \mu_{i,q_i}}{\nu_{i,q_i}} , \tag{69}$$

which has the same form as $\phi(x_i)$ (65); they become identical, $\varphi(x_i) = \phi(x_i)$, when noiseless IFA is used with the source state posteriors updated at each iteration ($S = 1$). We therefore conclude that the first phase of Seesaw is equivalent to ICA.

We remark that, although ICA can sometime accomplish separation using an inaccurate source density model (e.g., speech signals with a Laplacian density $p(x_i) \approx e^{-|x_i|}$ are successfully separated using the model $p(x_i) = \cosh^{-2}(x_i)$), model inaccuracies often lead to failure. For example, a mixtures of negative-kurtosis signals (e.g., with a uniform distribution) could not be separated using the $\cosh^{-2}$ model whose kurtosis is positive. Thus, when the densities of the sources at hand are not known in advance, the algorithm's ability to learn them becomes crucial.

A parametric source model can, in principle, be directly incorporated into ICA (Mackay 1996; Pearlmutter and Parra 1997) by deriving gradient-descent learning rules for the its parameters θ_i via $\delta\theta_i = -\eta\partial\mathcal{E}/\partial\theta_i$, in addition to the rule for $\mathbf{G}$. Unfortunately, the resulting learning rate is quite low, as is also the case when non-parametric density estimation methods are used (Pham 1996). Alternatively, the source densities may be approximated using cumulant methods such as the Edgeworth or Gram-Charlier expansions (Comon 1994; Amari et al. 1996; Cardoso and Laheld 1996); this approach produces algorithms that are less robust since the approximations are not true probability densities, being non-normalizable and sometimes negative.

In contrast, our noiseless IFA algorithm, and in particular its Seesaw GEM version, resolves these problems by combining ICA with source density learning rules in a manner that exploits the efficiency offered by the EM technique.

7.3 Noiseless IFA: Simulation Results

In this section we demonstrate and compare the performance of the Chase and Seesaw GEM algorithms on noiseless mixtures of $L = 3$ sources. We used 5sec-long speech and music signals obtained from commercial CD's, as well as synthetic signals produced by a random number generator, at sampling rate of $f_s = 8.82$kHz. The source signal densities used in the following example are shown in Figure 2. Those signals were scaled to unit variance and mixed by a random $L \times L$ mixing matrix $\mathbf{H}^0$. The learning rules (64–66), used in the manner required by either the Chase or Seesaw procedures, were iterated in batch mode, starting from random parameter values. We used a fixed learning rate $\eta = 0.05$.

Figure 9 shows the convergence of the estimated separating matrix $\mathbf{G}$ (left) and the source densities $p(x_i)$ (right) for Chase (top) and Seesaw (bottom). The distance of $\mathbf{G}^{-1}$ from the true mixing matrix $\mathbf{H}^0$ is

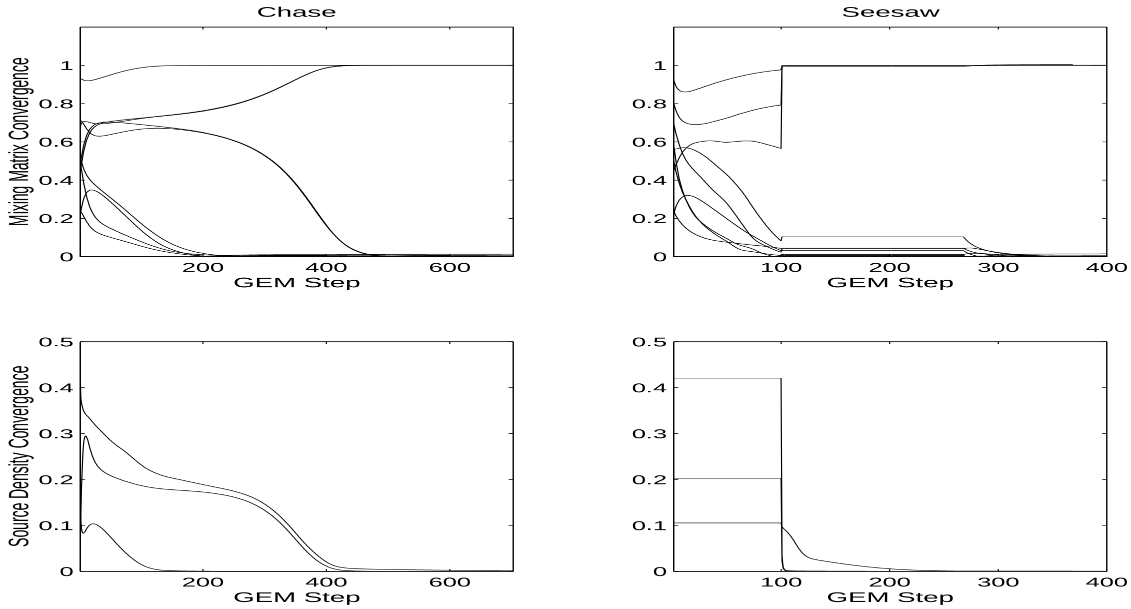

Figure 9: Top: convergence of the separating matrix **G** (left) and the source densities $p(x_i)$ (right) for the Chase algorithm with $L = 3$ sources. For **G** we plot the matrix elements of **J** (70) against GEM step number, whereas for $p(x_i)$ we plot their KL distance K_i (42) from the true densities. Bottom: same for the Seesaw algorithm.

quantified by the matrix elements of

$$\mathbf{J} = \mathbf{G}\mathbf{H}^0 \ . \tag{70}$$

Notice that for the correct estimate $\mathbf{G}^{-1} = \mathbf{H}^0$, **J** becomes the unit matrix **I**. Recall that the effect of source scaling is eliminated by (67); to prevent possible source permutations from affecting this measure, we permuted the columns of **G** such that the largest element (in absolute value) in column i of **J** would be J_{ii}. Indeed, this product is shown to converge to **I** in both cases. For the source densities, we plot their KL distances K_i (42) from the true densities $p^0(x_i)$, which approach zero as the learning proceeds. Notice that Seesaw required a smaller number of steps to converge; similar results were observed in other simulations we performed.

Seesaw was used in the following manner: after initializing the parameters, the MOG parameters were frozen and phase (i) proceeded for $S = 100$ iterations on **G**. Then **G** was frozen (except for the scaling (67)), and phase (ii) proceeded until the maximal relative increment of the MOG parameters decreased below 5×10^{-4}. This phase alternation is manifested in Figure 9 by K_i being constant as **J** changes and vice versa. In particular, the upward jump of one of the elements of **J** after $S = 100$ iterations is caused by the scaling (67), which is performed only in phase (ii).

To demonstrate the advantage of noiseless IFA over Bell and Sejnowski's (1995) ICA, we applied both algorithms to a mixture of $L = 2$ sources whose densities are plotted in Figure 10 (left). The Seesaw version of IFA was used. After learning, the recovered sources were obtained; their joint densities are displayed in Figure 10 for IFA (middle) and ICA (right). The sources recovered by ICA are clearly correlated, reflecting the fact that this algorithm uses a non-adaptive source density model that is unsuitable for the present case.

7.4 Relation to Principal Component Analysis

As mentioned at the beginning of this section, the EM algorithm for IFA presented in Section 3.2 fails to identify the mixing matrix **H** in the noiseless case. This can be shown by taking the zero-noise limit

$$\mathbf{\Lambda} = \eta \mathbf{I} \ , \qquad \eta \to 0 \ , \tag{71}$$

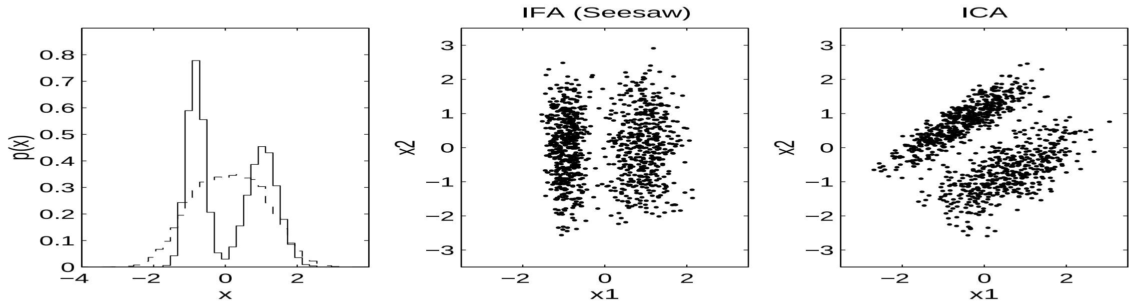

Figure 10: Noiseless IFA vs. ICA. Left: source densities histograms. These sources were mixed by a random 2 × 2 matrix. Middle: joint density of the sources recovered from the mixtures by Seesaw. Right: same for ICA.

where $\mathbf{I}$ is the $L \times L$ unit matrix, and examine the learning rule for $\mathbf{H}$ (first line in (29)). Using (71) in (80,81), the source posterior becomes singular,

$$p(\mathbf{x} \mid \mathbf{q}, \mathbf{y}) = \delta\left[\mathbf{x} - \rho(\mathbf{y})\right] \ , \quad \rho(\mathbf{y}) = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \ , \tag{72}$$

and loses its dependence of the source states $\mathbf{q}$. This simply expresses the fact that, for a given observation $\mathbf{y}$, the sources $\mathbf{x}$ are given by their conditional mean $\langle \mathbf{x} \mid \mathbf{y} \rangle$ with zero variance,

$$\langle \mathbf{x} \mid \mathbf{y} \rangle = \rho(\mathbf{y}) \ , \quad \langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y} \rangle = \rho(\mathbf{y})\rho(\mathbf{y})^T \ , \tag{73}$$

as indeed is expected for zero noise.

The rule for $\mathbf{H}$ (29) now becomes

$$\mathbf{H} = \mathbf{C_y}\mathbf{H}'(\mathbf{H}'^T\mathbf{C_y}\mathbf{H}')^{-1}\mathbf{H}'^T\mathbf{H}' \ , \tag{74}$$

where $\mathbf{H}'$ is the mixing matrix obtained in the previous iteration and $\mathbf{C_y} = E\mathbf{y}\mathbf{y}^T$ is the covariance matrix of the observed sensor data. This rule contains no information about the source parameters; in effect, the vanishing noise disconnected the bottom hidden layer from the top one. The bottom+visible layers now form a separate generative model of *Gaussian* sources (since the only source property used is their vanishing correlations) that are mixed linearly without noise.

In fact, if the columns of $\mathbf{H}'$ are L of the orthogonal L' directions defined by the principal components of the observed data (recall that this matrix is $L' \times L$), the algorithm will stop. To see that, assume $\mathbf{H}^T\mathbf{C_y}\mathbf{H} = \mathbf{D}$ is diagonal and the columns of $\mathbf{H}$ are orthonormal (namely, $\mathbf{H}^T\mathbf{H} = \mathbf{I}$). Then $\mathbf{D}$ contains L eigenvalues of the data covariance matrix, which itself can be expressed as $\mathbf{C_y} = \mathbf{H}\mathbf{D}\mathbf{H}^T$. By direct substitution, the rule (74) reduces to $\mathbf{H} = \mathbf{H}'$. Hence, the M-step contributes nothing towards minimizing the error since $W = W'$ is already a minimum of $\mathcal{F}(W', W)$ (22), so $\mathcal{F}(W', W) = \mathcal{F}(W', W')$ in (28). Mathematically, the origin of this phenomenon lies in the sensor density conditioned on the sources (14) becoming non-analytic, i.e., $p(\mathbf{y} \mid \mathbf{x}) = \delta(\mathbf{y} - \mathbf{H}\mathbf{x})$.

A more complete analysis of the generative model formed by linearly mixing uncorrelated Gaussian variables (Tipping and Bishop 1997) shows that any $\mathbf{H}$, whose columns span the L-dimensional space defined by any L principal directions of the data, is a stationary point of the corresponding likelihood; in particular, when the spanned space is defined by the *first* L principal directions, the likelihood is maximal at that point.

We conclude that in the zero-noise case, the EM algorithm (29,30) performs PCA rather than IFA, with the top layer learning a factorial MOG model for some linear combinations of the first L principal components. For non-zero but very low noise, convergence from the PCA to IFA solution will therefore be rather slow, and the noiseless IFA algorithm may become preferable.

It is also interesting to point out that the rule (74), obtained as a special case of noiseless IFA, has been discovered quite recently by Tipping and Bishop (1997) and independently by Roweis (1998) as an EM algorithm for PCA.

8 Conclusion

This paper introduced the concept of independent factor analysis, a new method for statistical analysis of multi-variable data. By performing IFA, the data are interpreted as arising from independent, unobserved sources that are mixed by a linear transformation with added noise. In the context of the blind source separation problem, IFA separates non-square, noisy mixtures where the sources, mixing process, and noise properties are all unknown.

To perform IFA we introduced the hierarchical IF generative model of the mixing situation, and derived an EM algorithm that learns the model parameters from the observed sensor data; the sources are then reconstructed by an optimal non-linear estimator. Our IFA algorithm reduces to the well-known EM algorithm for ordinary FA when the model sources become Gaussian. In the noiseless limit it reduces to the EM algorithm for PCA. As the number of sources increases, the exact algorithm becomes intractable; an approximate algorithm, based on a variational approach, has been derived and its accuracy demonstrated.

An EM algorithm specifically for noiseless IFA, associated with a linear source estimator, has also been derived. This algorithm and, in particular, its generalized EM versions, combine separating the sources by Bell and Sejnowski's (1995) ICA with learning their densities using the EM rules for mixtures of Gaussians. In the Chase version, the source densities are learned simultaneously with the separating matrix, whereas the Seesaw version learns the two parameter sets in alternating phases. Hence, an efficient solution is provided for the problem of incorporating adaptive source densities into ICA.

A generative model similar to IF were recently proposed by Lewicki and Sejnowski (1998). In fact, their model was implicit in Olshausen and Field's (1996) algorithm, as exposed in Olshausen (1996). This model uses a Laplacian source prior $p(x_i) \propto e^{-|x_i|}$, and the integral over the sources required to obtain $p(\mathbf{y})$ in (7) is approximated by the value of the integrand at its maximum; this approximation can be improved upon by incorporating Gaussian corrections (Lewicki and Sejnowski 1998). The resulting algorithm was used to derive efficient codes for images and sounds (Lewicki and Olshausen 1998), and was put forth as a computational model for interpreting neural responses in V1 in the efficient coding framework (Olshausen and Field 1996, 1997). In contrast with IFA, this algorithm use a non-adaptive source density model and may perform poorly on non-Laplacian sources; it uses gradient ascent rather than the efficient EM method; and the approximations involved in its derivation must be made even for a small number of sources, where exact IFA is available. It will be interesting to compare the performance of this algorithm with variational IFA on mixtures of many sources with arbitrary densities.

An EM algorithm for noisy BSS, which was restricted to discrete sources whose distributions are known in advance, was developed in Belouchrani and Cardoso (1994). Moulines et al. (1997) proposed an EM approach to noisy mixing of continuous sources. They did not discuss source reconstruction, and their method was restricted to a small number of sources and did not extend to noiseless mixing; nevertheless, they had essentially the same insight as the present paper regarding the advantage of mixture source models. A related idea was discussed in Roweis and Ghahramani (1997).

An important issue that deserves a separate discussion is the determination of the number L of hidden sources, assumed known throughout this paper. L is not a simple parameter since increasing the number of sources increases the number of model parameters, resulting, in effect, in a different generative model. Hence, to determine L one should use model comparison methods, on which extensive literature is available (see, e.g., Mackay's (1992) discussion of Bayesian model comparison using the evidence framework). A much simpler but imprecise method would exploit the data covariance matrix $\mathbf{C_y} = E\mathbf{y}\mathbf{y}^T$, and fix the number of sources at the number of its 'significant' (with respect to some threshold) eigenvalues. This method is suggested by the fact that in the zero-noise case, the number of positive eigenvalues is precisely L; however, for the noisy case the result will depend strongly on the threshold (which there is no systematic way to determine), and the accuracy of this method is expected to decrease with increasing noise level.

Viewed as a data modeling tool, IFA provides an alternative to factor analysis on the one hand and to mixture models on the other, by suggesting a description of the data in terms of a highly constrained mixture of co-adaptive Gaussians, and simultaneously in terms of independent underlying sources which may reflect the actual generating mechanism of those data. In this capacity, IFA may be used for noise removal and completion of missing data. It is also related to the statistical methods of projection pursuit (Friedman and Stuetzle 1981; Huber 1985) and generalized additive models (Hastie and Tibshirani 1990); a comparative study of IFA and those techniques would be of great interest.

Viewed as a compression tool, IFA constitutes a new method for redundancy reduction of correlated multi-channel data into a factorial few-channel representation given by the reconstructed sources. It is well known that the optimal *linear* compression is provided by PCA and is characterized by the absence of second-order correlations among the new channels. In contrast, the compressed IFA representation is a *non-linear* function of the original data, where the non-linearity is effectively optimized to ensure the absence of correlations of arbitrarily high orders.

Finally, viewed as a tool for source separation in realistic situations, IFA is currently being extended to handle noisy *convolutive* mixing, where H becomes a matrix of filters. This extension exploits spatio-temporal generative models introduced by Attias and Schreiner (1998), where they served as a basis for deriving gradient-descent algorithms for convolutive noiseless mixtures. A related approach to this problem is outlined in Moulines et al. (1997). In addition to more complicated mixing models, IFA allows the use of complex models for the source densities, resulting in source estimators that are optimized to the properties of the sources and can thus reconstruct them more faithfully from the observed data. A simple extension of the source model used in the present paper could incorporate the source auto-correlations, following Attias and Schreiner (1998); this would produce a non-linear, multi-channel generalization of the Wiener filter. More powerful models may include useful high-order source descriptions.

A IFA: Derivation of the EM Algorithm

Here we provide the derivation of the EM learning rules (29,30) from the approximate error (26).

A.1 E-Step

To obtain $\mathcal{F}$ in terms of the IF model parameters W, we first substitute $p(\mathbf{y} \mid \mathbf{x}) = \mathcal{G}(\mathbf{y} - \mathbf{Hx}, \mathbf{\Lambda})$ (14) in (26) and obtain, with a bit of algebra,

$$\mathcal{F}_V = \frac{1}{2} \log \mid \det \mathbf{\Lambda} \mid + \frac{1}{2} \mathrm{Tr}\, \mathbf{\Lambda}^{-1} \left(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\langle \mathbf{x}^T \mid \mathbf{y} \rangle \mathbf{H}^T + \mathbf{H} \langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y} \rangle \mathbf{H}^T \right) \ . \tag{75}$$

The integration over the sources $\mathbf{x}$ required to compute $\mathcal{F}_V$ (26) appears in (75) via the conditional mean and covariance of the sources given the observed sensor signals, defined by

$$\langle m(\mathbf{x}) \mid \mathbf{y}, W' \rangle = \int d\mathbf{x}\, p(\mathbf{x} \mid \mathbf{y}, W')\, m(\mathbf{x}) \ , \tag{76}$$

where we used $m(\mathbf{x}) = \mathbf{x}, \mathbf{x}\mathbf{x}^T$; note that these conditional averages depend on the parameters W' produced by the previous iteration. We point out that for a given $\mathbf{y}$, $\langle \mathbf{x} \mid \mathbf{y} \rangle$ is an $L \times 1$ vector and $\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y} \rangle$ is an $L \times L$ matrix.

Next, we substitute $p(x_i \mid q_i) = \mathcal{G}(x_i - \mu_{i,q_i}, \nu_{i,q_i})$ in (26) to get

$$\mathcal{F}_B = \sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid \mathbf{y}, W') \left[\frac{1}{2} \log \nu_{i,q_i} + \frac{1}{2\nu_{i,q_i}} \left(\langle x_i^2 \mid q_i, \mathbf{y} \rangle - 2 \langle x_i \mid q_i, \mathbf{y} \rangle \mu_{i,q_i} + \mu_{i,q_i}^2 \right) \right] \ , \tag{77}$$

where the integration over the source x_i indicated in $\mathcal{F}_B$ (26) enters via the conditional mean and variance of this source given both the observed sensor signals and the hidden state of this source, defined by

$$\langle m(x_i) \mid q_i, \mathbf{y}, W' \rangle = \int dx_i\, p(x_i \mid q_i, \mathbf{y}, W')\, m(x_i) \ , \tag{78}$$

and we used $m(x_i) = x_i,\ x_i^2$. Note from (77) that the quantity we are actually calculating is the joint conditional average of the source signal x_i and state q_i, i.e., $\langle x_i, q_i \mid \mathbf{y}, W' \rangle = p(q_i \mid \mathbf{y}, W') \langle m(x_i) \mid q_i, \mathbf{y}, W' \rangle = \int dx_i \; p(x_i, q_i \mid \mathbf{y}, W') \; m(x_i)$. We broke the posterior over those hidden variables as in (77) for computational convenience.

Finally, for the top layer we have

$$\mathcal{F}_T = -\sum_{i=1}^{L} \sum_{q_i=1}^{n_i} p(q_i \mid \mathbf{y}, W') \; \log w_{i,q_i} \; . \tag{79}$$

To complete the E-step we must express the conditional averages (76,78) explicitly in terms of the parameters W'. The key to this calculation are the conditional densities $p(\mathbf{x} \mid \mathbf{q}, \mathbf{y}, W')$ and $p(\mathbf{q} \mid \mathbf{y}, W')$, whose product is the posterior density of the unobserved source signals and states given the observed sensor signals, $p(\mathbf{x}, \mathbf{q} \mid \mathbf{y}, W')$. Starting from the joint (15), it is straightforward to show that, had both the sensor signals and the state from which each source is drawn been known, the sources would have a Gaussian density,

$$p(\mathbf{x} \mid \mathbf{q}, \mathbf{y}) = \mathcal{G}\left[\mathbf{x} - \boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y}), \Sigma_{\mathbf{q}}\right] \; , \tag{80}$$

with covariance matrix and mean given by

$$\Sigma_{\mathbf{q}} = \left(\mathbf{H}^T \boldsymbol{\Lambda}^{-1} \mathbf{H} + \mathbf{V}_{\mathbf{q}}^{-1}\right)^{-1} \; , \qquad \boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y}) = \boldsymbol{\Sigma}_{\mathbf{q}} \left(\mathbf{H}^T \boldsymbol{\Lambda}^{-1} \mathbf{y} + \mathbf{V}_{\mathbf{q}}^{-1} \boldsymbol{\mu}_{\mathbf{q}}\right) \; . \tag{81}$$

Note that the mean depends linearly on the data.

The posterior probability of the source states given the sensor data can be obtained from (12,17) via

$$p(\mathbf{q} \mid \mathbf{y}) = \frac{p(\mathbf{q}) p(\mathbf{y} \mid \mathbf{q})}{\sum_{\mathbf{q}'} p(\mathbf{q}') p(\mathbf{y} \mid \mathbf{q}')} \; . \tag{82}$$

We are now able to compute the conditional source averages. From (80) we have

$$\langle \mathbf{x} \mid \mathbf{q}, \mathbf{y} \rangle = \boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y}) \; , \quad \langle \mathbf{x}\mathbf{x}^T \mid \mathbf{q}, \mathbf{y} \rangle = \Sigma_{\mathbf{q}} + \boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y}) \boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y})^T \; . \tag{83}$$

To obtain the conditional averages given only the sensors (76) we sum (83) over the states $\mathbf{q}$ with probabilities $p(\mathbf{q} \mid \mathbf{y})$ (82) to get

$$\langle m(\mathbf{x}) \mid \mathbf{y} \rangle = \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{y}) \langle m(\mathbf{x}) \mid \mathbf{q}, \mathbf{y} \rangle \; , \tag{84}$$

taking $m(\mathbf{x}) = \mathbf{x},\ \mathbf{x}\mathbf{x}^T$. We point out that the corresponding source posterior density, given by $p(\mathbf{x} \mid \mathbf{y}) = \sum_{\mathbf{q}} p(\mathbf{q} \mid \mathbf{y}) p(\mathbf{x} \mid \mathbf{q}, \mathbf{y})$, is a co-adaptive MOG, just like the sensor density $p(\mathbf{y})$ (16). Notice that the sums over $\mathbf{q}$ in (82) and (84) mean $\sum_{q_1} \sum_{q_2} \cdots \sum_{q_L}$.

Individual source averages (78) appear in (77) together with the corresponding state posterior, and their product is given by summing over all the other sources,

$$p(q_i \mid \mathbf{y}) \langle m(x_i) \mid q_i, \mathbf{y} \rangle = \sum_{\{q_j\}_{j \neq i}} p(\mathbf{q} \mid \mathbf{y}) \langle m(x_i) \mid \mathbf{q}, \mathbf{y} \rangle \; , \tag{85}$$

and using the results (82,83).

Finally, the individual state posterior appearing in (79) is similarly obtained from (82):

$$p(q_i \mid \mathbf{y}) = \sum_{\{q_j\}_{j \neq i}} p(\mathbf{q} \mid \mathbf{y}) \; . \tag{86}$$

We emphasize that all the parameters appearing in (80–86) belong to W'. Substituting these expressions in (75,77,79) and adding them up completes the E-step which yields $\mathcal{F}(W', W)$.

A.2 M-Step

To derive the EM learning rules we must minimize $\mathcal{F}(W', W)$ obtained above with respect to W. This can be done by first computing its gradient $\partial\mathcal{F}/\partial W$ layer by layer. For the visible layer parameters we have

$$\begin{aligned} \frac{\partial \mathcal{F}_V}{\partial \mathbf{H}} &= \mathbf{\Lambda}^{-1}\mathbf{y}\langle \mathbf{x}^T \mid \mathbf{y}\rangle - \mathbf{\Lambda}^{-1}\mathbf{H}\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y}\rangle \ , \\ \frac{\partial \mathcal{F}_V}{\partial \mathbf{\Lambda}} &= -\frac{1}{2}\mathbf{\Lambda}^{-1} + \frac{1}{2}\mathbf{\Lambda}^{-1}\left(\mathbf{y}\mathbf{y}^T - 2\mathbf{y}\langle \mathbf{x}^T \mid \mathbf{y}\rangle\mathbf{H}^T + \mathbf{H}\langle \mathbf{x}\mathbf{x}^T \mid \mathbf{y}\rangle\mathbf{H}^T\right)\mathbf{\Lambda}^{-1} \ , \end{aligned} \tag{87}$$

whereas for the bottom hidden layer

$$\begin{aligned} \frac{\partial \mathcal{F}_B}{\partial \mu_{i,q_i}} &= -\frac{1}{\nu_{i,q_i}} p(q_i \mid \mathbf{y})\left(\langle x_i \mid q_i, \mathbf{y}\rangle - \mu_{i,q_i}\right) \ , \\ \frac{\partial \mathcal{F}_B}{\partial \nu_{i,q_i}} &= -\frac{1}{2\nu_{i,q_i}^2} p(q_i \mid \mathbf{y})\left(\langle x_i^2 \mid q_i, \mathbf{y}\rangle - 2\langle x_i \mid q_i, \mathbf{y}\rangle\mu_{i,q_i} + \mu_{i,q_i}^2 - \nu_{i,q_i}\right) \ . \end{aligned} \tag{88}$$

In computing the gradient with respect to the top hidden layer parameters we should ensure that, being probabilities $w_{i,q_i} = p(q_i)$, they satisfy the non-negativity $w_{i,q_i} \geq 0$ and normalization $\sum_{q_i} w_{i,q_i} = 1$ constraints. Both can be enforced automatically by working with new parameters $\bar{w}_{i,q_i}$, related to the mixing proportions through

$$w_{i,q_i} = \frac{e^{\bar{w}_{i,q_i}}}{\sum\limits_{q_i'} e^{\bar{w}_{i,q_i'}}} \ . \tag{89}$$

The gradient is then taken with respect to the new parameters:

$$\frac{\partial \mathcal{F}_T}{\partial \bar{w}_{i,q_i}} = -p(q_i \mid \mathbf{y}) + w_{i,q_i} \ . \tag{90}$$

Recall that the conditional source averages and state probabilities depend on W' and that the equations (87–90) include averaging over the observed $\mathbf{y}$. We now set the new parameters W to the values that make the gradient vanish, obtaining the IF learning rules (29,30).

B Variational IFA: Derivation of the Mean-Field Equations

To derive the mean-field equations (52–54), we start from the approximate error $\mathcal{F}(\boldsymbol{\tau}, W)$ (46) using the factorial posterior (47). The approximate error is composed of the three layer contributions and the negative entropy of the posterior, as in (25). $\mathcal{F}_V$, $\mathcal{F}_B$, and $\mathcal{F}_T$ are given by (75,77,79) with the conditional source means and densities expressed in terms of the variational parameters $\boldsymbol{\tau}$ via (49,50).

The last term in $\mathcal{F}$ is given by

$$\mathcal{F}_H(\boldsymbol{\tau}) = \sum_{i=1}^{L}\sum_{q_i=1}^{n_i} \kappa_{i,q_i}\left(\frac{1}{2}\log\xi_{i,q_i} - \log\kappa_{i,q_i}\right) + Const. \ , \tag{91}$$

where *Const.* reflects the fact that the source posterior is normalized. $\mathcal{F}_H$ (91) is obtained by using the factorial posterior (47) in (27). Note that since this term does not depend on the generative parameters W, it did not contribute to the exact EM algorithm but is crucial for the variational approximation.

To minimize $\mathcal{F}$ with respect to $\boldsymbol{\tau}$ we compute its gradient $\partial\mathcal{F}/\partial\boldsymbol{\tau}$:

$$\frac{\partial \mathcal{F}}{\xi_{i,q_i}} = -\frac{1}{2}\left(\bar{H}_{ii} + \frac{1}{\nu_{i,q_i}} - \frac{1}{\xi_{i,q_i}}\right)\kappa_{i,q_i} \ ,$$

$$\begin{aligned}\frac{\partial \mathcal{F}}{\psi_{i,q_i}} &= \left[(\mathbf{H}^T\mathbf{\Lambda}^{-1}\mathbf{y})_i + \frac{\mu_{i,q_i}}{\nu_{i,q_i}} - \sum_{j\neq i}\sum_{q_j=1}^{n_j} \bar{H}_{ij}\kappa_{i,q_j}\psi_{j,q_j} - \left(\bar{H}_{ii} + \frac{1}{\nu_{i,q_i}}\right)\psi_{i,q_i}\right]\kappa_{i,q_i} \ , \\ \frac{\partial \mathcal{F}}{\kappa_{i,q_i}} &= -\log\kappa_{i,q_i} + \log w_{i,q_i} + \frac{1}{2}\left(\log\xi_{i,q_i} + \frac{\psi^2_{i,q_i}}{\xi_{i,q_i}}\right) - \frac{1}{2}\left(\log\nu_{i,q_i} + \frac{\mu^2_{i,q_i} + \xi_{i,q_i}}{\nu_{i,q_i}}\right) \\ & -\frac{1}{2}\bar{H}_{ii}\xi_{i,q_i} + z_i \ . \end{aligned} \tag{92}$$

The first equation leads directly to (52). The second and third equations, after a bit of simplification using (52), lead to (53,54). The z_i reflect the normalization of the mixing proportions κ_{i,q_i}: to impose normalization, we actually minimize $\mathcal{F} + \sum_i z_i(\sum_{q_i}\kappa_{i,q_i} - 1)$ using the method of Lagrange multipliers.

C Noiseless IFA: Derivation of the GEM Algorithm

In this appendix we derive the GEM learning rules for the noiseless case (57). This derivation follows the same steps as the one in Appendix A.

C.1 E-Step

By substituting $p(x_i \mid q_i) = \mathcal{G}(x_i - \mu_i, \nu_i)$ in (62) we get for the bottom layer

$$\mathcal{F}_B = \sum_{i=1}^{L}\sum_{q_i=1}^{n_i} p(q_i \mid x_i, W')\left[\frac{1}{2}\log\nu_{i,q_i} + \frac{(x_i - \mu_{i,q_i})^2}{2\nu_{i,q_i}}\right] \ , \tag{93}$$

whereas for the top layer we have

$$\mathcal{F}_T = -\sum_{i=1}^{L}\sum_{q_i=1}^{n_i} p(q_i \mid x_i, W') \ \log w_{i,q_i} \ . \tag{94}$$

Note that, unlike $\mathcal{F}_B$ in the noisy case, no conditional source means should be computed. The posterior probability of the i-th source states is obtained from Bayes' rule:

$$p(q_i \mid x_i) = \frac{p(x_i \mid q_i)p(q_i)}{\sum\limits_{q'_i} p(x_i \mid q'_i)p(q'_i)} \ . \tag{95}$$

C.2 M-Step

To derive the learning rule for the unmixing matrix $\mathbf{G}$ we use the error gradient

$$\frac{\partial \mathcal{F}}{\partial \mathbf{G}} = -\left(\mathbf{G}^T\right)^{-1} + \phi(\mathbf{x})\mathbf{y}^T \ , \tag{96}$$

where $\phi(\mathbf{x})$ is given by (65). To determine the increment of $\mathbf{G}$ we use the relative gradient of the approximate error,

$$\delta\mathbf{G} = -\eta\frac{\partial \mathcal{F}}{\partial \mathbf{G}}\mathbf{G}^T\mathbf{G} = \eta\mathbf{G} - \eta\phi(\mathbf{x})\mathbf{x}^T\mathbf{G} \ . \tag{97}$$

Since the extremum condition $\delta\mathbf{G} = 0$, implying $E\phi(\mathbf{G}\mathbf{y})\mathbf{y}^T\mathbf{G}^T = \mathbf{I}$, is not analytically solvable, (97) leads to the iterative rule (64).

As explained in (Amari et al. 1996; Cardoso and Laheld 1996; Mackay 1996), the relative gradient has an advantage over the ordinary gradient since the algorithm it produces is equivariant, i.e., its performance is independent of the rank of the mixing matrix, and its computational cost is lower since it does not require matrix inversion.

The learning rules (30) for the MOG source parameters are obtained from the gradient of the bottom and top layer contributions,

$$\begin{aligned}
\frac{\partial \mathcal{F}_B}{\partial \mu_{i,q_i}} &= -\frac{1}{\nu_{i,q_i}} p(q_i \mid x_i)(x_i - \mu_{i,q_i}) \ , \\
\frac{\partial \mathcal{F}_B}{\partial \nu_{i,q_i}} &= -\frac{1}{2\nu_{i,q_i}} p(q_i \mid x_i) \left[(x_i - \mu_{i,q_i})^2 - \nu_{i,q_i} \right] \ , \\
\frac{\partial \mathcal{F}_T}{\partial \bar{w}_{i,q_i}} &= -p(q_i \mid x_i) + w_{i,q_i} \ , \qquad (98)
\end{aligned}$$

where the last line was obtained using (89).

Acknowledgements

I thank B. Bonham, K. Miller, S. Nagarajan, T. Troyer, and especially V. deSa, for useful discussions. Thanks are also due to two anonymous referees for very helpful suggestions. Research was supported by The Office of Naval Research (N00014-94-1-0547), NIDCD (R01-02260), and the Sloan Foundation.

References

Amari, S., Cichocki, A., and Yang, H.H. (1996). A new learning algorithm for blind signal separation. In Touretzky, D.S., Mozer, M.C., and Hasselmo, M.E. (Eds.), *Advances in Neural Information Processing Systems 8.* MIT Press, Cambridge, MA.

Attias, H. and Schreiner, C.E. (1998). Blind source separation and deconvolution: the dynamic component analysis algorithm. *Neural Computation* **10**, 1373-1424.

Bell, A.J. and Sejnowski, T.J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* **7**, 1129-1159.

Belouchrani, A. and Cardoso, J.-F. (1994). Maximum likelihood source separation for discrete sources. In *Proc. EUSIPCO*, 768-771.

Bishop, C.M., Svensén, M., and Williams, C.K.I. (1998). GTM: the generative topographic mapping. *Neural Computation* **10**, 215-234.

Cardoso, J.-F. and Laheld, B.H. (1996). Equivariant adaptive source separation. *IEEE Transactions on Signal Processing* **44**, 3017-3030.

Cardoso, J.-F. (1997). Infomax and maximum likelihood for source separation. *IEEE Signal Processing Letters* **4**, 112-114.

Comon, P., Jutten, C., and Herault, J. (1991). Blind separation of sources, Part II: Problem Statement. *Signal Processing* **24**, 11-20.

Comon, P. (1994). Independent component analysis: a new concept? *Signal Processing* **36**, 287-314.

Cover, T.M. and Thomas, J.A. (1991). *Elements of Information Theory.* John Wiley, New York.

Dayan, P., Hinton, G., Neal, R., and Zemel, R. (1995). The Helmholtz machine. *Neural Computation* **7**, 889-904.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B* **39**, 1-38.

Everitt, B.S. (1984). *An Introduction to Latent Variable Models.* Chapman and Hall, London.

Friedman, J.H. and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association* **76**, 817-823.

Ghahramani, Z. (1995). Factorial learning and the EM algorithm. In Tesauro, G., Touretzky, D.S., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems 7*, 617-624. Morgan Kaufmann, San Francisco, CA.

Ghahramani, Z. and Jordan, M.I. (1997). Factorial hidden Markov models. *Machine learning* **29**, 245-273.

Hastie, T.J., and Tibshirani, R.J. (1990). *Generalized Additive Models.* Chapman & Hall, London.

Hinton, G.E., Williams, C.K.I., and Revow, M.D. (1992). Adaptive elastic models for hand-printed character recognition. In Moody, J.E., Hanson, S.J., and Lippmann, R.P. (Eds.), *Advances in Neural Information Processing Systems 4*, 512-519. Morgan Kaufmann.

Hinton, G.E. and Zemel, R.S. (1994). Autoencoders, minimum description length, and Helmholtz free energy. In Cowan, J.D., Tesauro, G., and Alspector, J. (Eds.), *Advances in Neural Information Processing Systems 6.* Morgan Kaufmann, San Francisco, CA.

Hinton, G.E., Dayan, P., Frey, B.J., and Neal, R.M. (1995). The "wake-sleep" algorithm for unsupervised neural networks. *Science* **268**. 1158-1161.

Huber, P.J. (1985). Projection pursuit. *Annals of Statistics* **13**, 435-475.

Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation* **9**, 1483-1492.

Jordan, M.I., and Jacobs, R.A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* **6**, 181-214.

Jutten, C., and Herault, J. (1991). Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing* **24**, 1-10.

Lee, T.-W., Bell, A.J., and Lambert, R. (1997). Blind separation of delayed and convolved sources. In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*, 758-764. MIT Press, Cambridge, MA.

Lewicki, M.S. and Sejnowski, T.J. (1998). Learning nonlinear overcomplete representations for efficient coding. In *Advances in Neural Information Processing Systems 10*, in press.

Lewicki, M.S. and Olshausen, B.A. (1998). Inferring sparse, overcomplete image codes using an efficient coding framework. In *Advances in Neural Information Processing Systems 10*, in press.

MacKay, D.J.C. (1992). Bayesian interpolation. *Neural Computation* **4**, 415-447.

MacKay, D.J.C. (1996). Maximum likelihood and covariant algorithms for independent component analysis. Technical report, Cavendish Laboratory, Cambridge University.

Moulines, E., Cardoso, J.-P., Gassiat, E. (1997). Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In *Proceedings of IEEE Conference on Acoustics, Speech, and Signal Processing 1997*, vol. 5, 3617-3620. IEEE.

Neal, R.M. and Hinton, G.E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M.I. (Ed.), *Learning in Graphical Models.* Kluwer Academic Press, in press.

Olshausen, B.A. and Field, D.J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607-609.

Olshausen, B.A. (1996). Learning linear, sparse, factorial codes. Technical Report AI Memo 1580, CBCL 138, Artificial Intelligence Lab, MIT.

Olshausen, B..A. and Field, D.J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research* **37**, 3311-3325.

Pearlmutter, B.A. and Parra, L.C. (1997). Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*, 613-619. MIT Press, Cambridge, MA.

Pham, D.T. (1996). Blind separation of instantaneous mixture of sources via an independent component analysis. *IEEE Transactions on Signal Processing* **44**, 2768-2779.

Roweis, S. and Ghahramani, Z. (1997). A unifying review of linear Gaussian models. Technical report, dept. of computer science, U. of Toronto.

Roweis, S. (1998). EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*, in press.

Rubin, D. and Thayer, D. (1982). EM algorithms for ML factor analysis. *Psychometrika* **47**, 69-76.

Saul, L. amd Jordan, M.I. (1995). Exploiting tractable structures in intractable networks. In Touretzky, D.S., Mozer, M.C., and Hasselmo, M.E. (Eds.), *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA.

Saul, L.K., Jaakkola, T., and Jordan, M.I. (1996). Mean field theory of sigmoid belief networks. *Journal of Artificial Intelligence Research* **4**, 61-76.

Tipping, M.E. and Bishop, C.M. (1997). Probabilistic principal component analysis. Technical report NCRG/97/010.

Torkkola, K. (1996). Blind separation of convolved sources based on information maximization. In *Neural Networks for Signal Processing VI*, IEEE, New York.