Hierarchical IFA Belief Networks

H. Attias
hagai@gatsby.ucl.ac.uk
Gatsby Computational Neuroscience Unit
University College London
17 Queen Square
London WC1N 3AR, U.K.

Abstract

We introduce a new real-valued belief network, which is a multilayer generalization of independent factor analysis (IFA). At each level, this network extracts real-valued latent variables that are non-linear functions of the input data with a highly adaptive functional form, resulting in a hierarchical distributed representation of these data. The network is based on a probabilistic generative model, constructed by cascading single-layer IFA models. Whereas exact maximum-likelihood learning for this model is intractable, we present and demonstrate an algorithm that maximizes a lower bound on the likelihood. This algorithm is developed by formulating a variational approach to hierarchical IFA networks.

1 INTRODUCTION

This paper introduces a new Bayesian network model for real-valued data, and presents an algorithm for learning and inference in this network. The purpose of this algorithm is to discover, in an unsupervised manner, explanations of data in terms of a small number of unobserved variables, whose relation to those data is non-linear, stochastic and may be highly complex.

Many belief networks have been proposed that are composed of binary units. The hidden units in such networks represent latent variables that explain different features of the data, and whose relation to the data is highly non-linear. However, for tasks such as object and speech recognition which produce real-valued data, the models provided by binary networks are often inadequate. Independent component analysis (ICA) learns a generative model from real data, and extracts real-valued latent variables that are mutually statistically independent. Unfortunately, this model is restricted to a single layer and the latent variables are simple linear functions of the data; hence, underlying degrees of freedom that are non-linear cannot be extracted by ICA. In addition, the requirement of equal numbers of hidden and observed variables and the assumption of noiseless data render the ICA model inappropriate. Fitting a mixture of independent component analyzers to data would, indeed, allow extracting latent variables that are non-linear functions of the data. However, since the expected data conditioned on the latent variables would still be linear, an ICA mixture model cannot describe situations where the observed data is a non-linear function of some unobserved quantity.

Nevertheless, ICA emerges in this paper as a suitable starting point for developing multilayer non-linear probabilistic models for real data. This algorithm was originally designed to solve a simplified form of the problem of auditory scene analysis, known in the field of statistical signal processing as ‘blind source separation’ [1,2]. In this problem, one considers L independent signal sources (e.g., different speakers in a room) and L' sensors (e.g., microphones at several locations). Each sensor receives a linear mixture of the source signals. In realistic situations, the sensor signals at time t depend on both present and past source signals, reflecting reverberation and multipath propagation, and are corrupted by noise. The task is to recover the unobserved sources from the observed sensor signals, in the absence of any information about the mixing process or the sources, apart from their mutual statistical independence. In the simplified version addressed by ICA, $L = L'$, the mixing is instantaneous (history-independent), the data are noise-free, and the sources are described by a temporally-independent density model. Hence, in this case, the sources are the explanations of the data and can be discovered by a linear belief network with non-Gaussian priors.

This paper begins by reviewing the independent factor analysis (IFA) technique [11], an extension of ICA that allows different numbers of latent and observed variables and can handle noisy data. It proceeds to create a multilayer network by cascading single-layer IFA models. The resulting generative model produces a hierarchical distributed representation of the input data, where the latent variables extracted at each level are *non-linear* functions of the data with a highly adaptive functional form. Whereas exact maximum-likelihood (ML) learning in this network is intractable due to the difficulty in computing the posterior density over the hidden layers, we present an algorithm that maximizes a lower bound on the likelihood. This algorithm is based on a general variational approach developed here for the IFA network.

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 $\mathbf{A}^T$ ($A^T_{ij} = A_{ji}$). The multi-variable Gaussian distribution for a random vector $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted by $\mathcal{G}(\mathbf{x} - \boldsymbol{\mu}, \boldsymbol{\Sigma})$.

2 INDEPENDENT FACTOR ANALYSIS

2.1 Blind Separation and ICA

Although the concept of ICA originated in the field of signal processing, it is actually a density estimation problem. Given an $L' \times 1$ observed data vector $\mathbf{y}$, the task is to explain it in terms of an $L \times 1$ vector $\mathbf{x}$ of hidden variables, which we henceforth term 'factors', that are mutually statistically independent. The relation between the two is assumed linear,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{u} \,, \tag{1}$$

where $\mathbf{H}$ is the 'mixing' matrix; the noise vector $\mathbf{u}$ is usually assumed zero-mean Gaussian with a covariance matrix $\boldsymbol{\Lambda}$. In the context of blind source separation [1–7], the factor and noise signals are unobservable; the factor signals $\mathbf{x}$ should be recovered from the mixed noisy signals $\mathbf{y}$ with no knowledge of $\mathbf{H}$, $\boldsymbol{\Lambda}$, or the factor densities $p(x_i)$, hence the term 'blind'. In the density estimation approach, one regards (1) as a probabilistic generative model for the observed $p(\mathbf{y})$, with the mixing matrix, noise covariance, and factor densities serving as model parameters. In principle, these parameters should be learned by ML, followed by inferring the factors via a MAP estimator.

One might expect that, since linear models have been analyzed and applied extensively for many years, the solution to the blind separation problem can be found in some textbook or review article. However, this is not the case. In the case of Gaussian factors, for example, (1) becomes the well known factor analysis (FA) model [8]. Its parameters can be estimated using an efficient expectation-maximization (EM) algorithm, and the optimal estimate of the factors is linear in the data. However, FA uses only second-order statistics of the data and cannot perform separation, because the resulting likelihood function is invariant under factor rotation (i.e., $\mathbf{H}$ is indistinguishable from $\mathbf{HP}$ for any orthogonal $\mathbf{P}$). Hence, capturing the non-Gaussian nature of the factors is crucial for achieving separation. Beyond FA, more modern statistical analysis methods, such as projection pursuit [9] and generalized additive models [10], 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 separation problem.

Most of the work on blind separation (e.g., [1–6]), focused on a simplified case where the number of hidden factors L equals the number of observed data variables L' (square mixing), the data are noise-free ($\mathbf{u} = 0$), and the mixing is instantaneous. Convolutive mixing, where $\mathbf{H}$ becomes a matrix of filters operating on the factors, is discussed in [7]. The factor densities $p(x_i)$ are usually fixed using prior knowledge. Learning $\mathbf{H}$ occurs via gradient-ascent maximization of the likelihood. Factor density parameters can also be adapted in this way [6,7], but the resulting gradient-ascent learning is rather slow. This state of affairs presented a problem to ICA algorithms, since the ability to learn arbitrary factor densities that are not known in advance is crucial: using an inaccurate $p(x_i)$ often leads to a bad $\mathbf{H}$ estimate and failed separation.

2.2 IFA with Zero Noise

The solution I gave in [11] was based on a factor model that (i) is capable of approximating arbitrary densities, and (ii) can be learned efficiently from data by EM. A simple semi-parametric model satisfying both requirements in a mixture of Gaussians (MOG). In that case,

$$p(x_i) = \sum_q \pi^i_q \, \mathcal{G}(x_i - \mu^i_q, \gamma^i_q) \tag{2}$$

is a weighted sum of n_i Gaussian densities labeled by q, with means μ^i_q, variances γ^i_q, and mixing proportions π^i_q. These Gaussians can be viewed as hidden 'states' of the factors. Denoting the state of factor i by q_i, its signal x_i is generated by selecting a state q with probability $p(q_i = q)$ independently at each time point t, followed by sampling from the corresponding Gaussian; the data are then generated via $y_i = \sum_j H_{ij} x_j$, where **H** is square and invertible. This is a probabilistic generative model for the data, defined by

$$\begin{aligned} p(q_i = q) &= \pi^i_q = \frac{\exp(a^i_q)}{\sum_{q'} \exp(a^i_{q'})} , \\ p(x_i \mid q_i = q) &= \mathcal{G}(x_i - \mu^i_q, \gamma^i_q) , \\ p(\mathbf{y}) &= \mid \det \mathbf{G} \mid \prod_{i=1}^{L} p(x_i) , \end{aligned} \tag{3}$$

where the last equation follows from $\mathbf{x} = \mathbf{G}\mathbf{y}$ with $\mathbf{G} = \mathbf{H}^{-1}$ (to within a factor ordering permutation). The softmax parametrization of π^i_q using a^i_q in (3) ensured positivity and normalization of the mixing proportions. A graphical representation of this model is provided by Fig. 2, if we set $n = 1$ and $y^0_j = b^1_{j,q} = \nu^1_{j,q} = \Lambda^1_{ij} = 0$.

The model density $p(\mathbf{y} \mid W)$ defined by (3) is parametrized by $W = \{G_{ij}, \mu^i_q, \gamma^i_q, a^i_q\}$. An EM algorithm can be derived, following [13], by bounding the log-likelihood $\mathcal{L} = \log p(\mathbf{y})$ from below:

$$\begin{aligned} \mathcal{L} &\geq \log \mid \mathbf{G} \mid \\ &+ \sum_i E[\log p(q_i, x_i) - \log p'(q_i \mid x_i)] . \end{aligned} \tag{4}$$

The bound makes use of the joint factor and state density $p(\mathbf{q}, \mathbf{x}) = \prod_i p(q_i, x_i)$ defined by the model (3), where $\mathbf{q} = (q_1, ..., q_L)^T$ denotes a hidden state configuration. The other component of the bound is a posterior density $p'(\mathbf{q} \mid \mathbf{x}) = \prod_i p'(q_i \mid x_i)$. The operator E averages over the hidden states $\mathbf{q}$ using the posterior p'. The inequality in (4), obtained from Jensen's inequality, holds for an arbitrary p'.

E-Step. In EM, p' is computed at each iteration from (3) via Bayes' rule, but using the parameters W' obtained in the previous iteration. As noted above, this posterior factorizes into a product of $v^i_q = p(q_i = q \mid x_i)$ over i, which depends on the data via $x_i = \sum_j G_{ij} y_j$.

M-Step. Following the calculation of v^i_q, the lower bound above is maximized with respect to the new parameters W. The maximization with respect to **G** is performed by gradient ascent using

$$\delta\mathbf{G} = \epsilon\mathbf{G} - \epsilon\overline{\phi(\mathbf{x})\mathbf{x}^T}\mathbf{G} , \tag{5}$$

where $\phi(x_i) = \sum_q v^i_q (x_i - \mu^i_q)/\gamma^i_q$, and ϵ is a properly chosen learning rate; the overline denotes averaging over the observed data **y**. The relative gradient [4,5] was used to derive (5). For the factor parameters we obtain the update rules

$$\begin{aligned} &\pi^i_q = \overline{v^i_q} , \\ &\mu^i_q = \frac{1}{\pi^i_q}\overline{v^i_q x_i} , \quad \gamma^i_q = \frac{1}{\pi^i_q}\overline{v^i_q x_i^2} - \mu^2_{i,q} . \end{aligned} \tag{6}$$

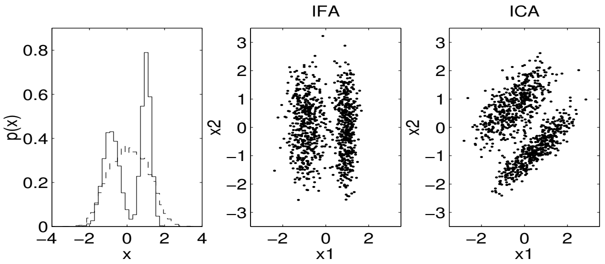

Figure 1: Left: factor densities. The two factors were mixed by a random 2 × 2 matrix. Middle: the outputs of the EM algorithm (5–6) are nearly independent. Right: the outputs of ICA [3] are correlated.

Scaling. In the blind separation problem, the factors and mixing matrix can be identified only to within an order permutation and scaling of the factors; in other words, the likelihood is invariant under these transformations. The continuous degrees of freedom added by the scaling invariance may delay convergence and cause numerical problems (e.g., G_{ij} may acquire arbitrarily large values). These effects can be minimized by scaling each factor x_j and row j of **G** at each iteration by a factor σ_j, which is determined by the factor variance or by the norm of the corresponding row. It is easy to show that this scaling leaves the likelihood function unchanged.

The algorithm (5–6) may be used in several possible generalized EM schemes. An efficient one is given by the following two-phase procedure: (i) freeze the factor parameters and learn the separating matrix **G** using (5); (ii) freeze **G** and learn the factor parameters using (6), then go back to (i) and repeat. Notice from the above definition of ϕ that for our factor model (3), $\phi(x_i) = -\partial \log p(x_i)/\partial x_i$. Hence, the rule (5) formally coincides with Bell and Sejnowski's ICA rule [3], which was derived for the special case $p(x_i) \propto \cosh^{-2}(x_i)$. We also recognize (6) as the EM learning rules for a 1-dim MOG. Therefore, in phase (i) our algorithm separates the factors using a generalized ICA rule, whereas in phase (ii) it learns a MOG model for each factor. Figure 1 illustrates its performance on a 2 × 2 mixture. This mixture is inseparable to ICA [3] because the factor model used by the latter does not fit the actual factor densities.

2.3 IFA with Non-Zero Noise

We now turn to the general problem, where the number of factors may differ from the number of sensors and noise is present. We model the noise density by a zero-mean Gaussian with covariance matrix $\mathbf{\Lambda}$. The ML estimation problem is now more difficult, as is evident from examining the likelihood:

$$p(\mathbf{y} \mid W) = \int d\mathbf{x}\, \mathcal{G}(\mathbf{y} - \mathbf{H}\mathbf{x}, \mathbf{\Lambda}) \prod_i p(x_i) \ . \qquad (7)$$

For non-Gaussian $p(x_i)$, one might expect that approximations (see [12] for fixed Laplacian densities) or numerical methods must be used to perform the integration over the factors.

However, MOG factors allow performing all the probabilistic calculations, including the above L-dim integral, analytically and exactly. An EM algorithm is derived by first noting that in the noisy case, both the factor signals x_i and states q_i are hidden variables. This is in contrast to the noise-free case where the x_i are deterministically related to the observed data. We begin, as before, with a lower bound on the log-likelihood:

$$\begin{aligned} \mathcal{L} = \log p(\mathbf{y}) \geq E \log p(\mathbf{y} \mid \mathbf{x}) \\ + \sum_{i=1}^{L} E \log p(q_i, x_i) - E \log p' \ , \end{aligned} \qquad (8)$$

where E denotes averaging using a posterior density $p' = p'(\mathbf{q}, \mathbf{x} \mid \mathbf{y})$ over the hidden variables which, unlike in the zero-noise case, does not factorize. Due to our noise model, $p(\mathbf{y} \mid \mathbf{x}) = \mathcal{G}(\mathbf{y} - \mathbf{H}\mathbf{x}, \mathbf{\Lambda})$; the factor density $p(q_i, x_i)$ is defined by (3).

E-Step. Here we calculate the posterior in terms of the previous iteration parameters W'. First, given a state configuration $\mathbf{q} = (q_1, ..., q_L)^T$, the data have a Gaussian density

$$p(\mathbf{y} \mid \mathbf{q}) = \mathcal{G}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu}_{\mathbf{q}}, \mathbf{H}\mathbf{\Gamma}_{\mathbf{q}}\mathbf{H}^T + \mathbf{\Lambda}) \ , \qquad (9)$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{q}} &= (\mu_{1,q_1}, ..., \mu_{L,q_L})^T \ , \\ \mathbf{\Gamma}_{\mathbf{q}} &= \mathrm{diag}(\gamma_{1,q_1}, ..., \gamma_{L,q_L}) \end{aligned} \qquad (10)$$

are determined by the mean and variances of the individual factors. The probability density $p(\mathbf{y})$ for generating a data vector is therefore a mixture of $\prod_i n_i$ Gaussians, with mixing proportions $p(\mathbf{q}) = \prod_i \pi^i_{q_i}$. The state posterior $v_{\mathbf{q}}(\mathbf{y}) = p(\mathbf{q} \mid \mathbf{y})$ is obtained from (9) via Bayes' rule.

Next, when both the data vector and state configuration are fixed, it can be shown that the factors are jointly Gaussian,

$$p(\mathbf{x} \mid \mathbf{q}, \mathbf{y}) = \mathcal{G}(\mathbf{x} - \boldsymbol{\rho}_{\mathbf{q}}, \mathbf{\Sigma}_{\mathbf{q}}) \ , \qquad (11)$$

with data-independent covariance matrix

$$\mathbf{\Sigma}_{\mathbf{q}} = (\mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{H}^T + \mathbf{\Gamma}_{\mathbf{q}}^{-1})^{-1} \qquad (12)$$

and data-dependent mean

$$\boldsymbol{\rho}_{\mathbf{q}}(\mathbf{y}) = \mathbf{\Sigma}_{\mathbf{q}}(\mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{y} + \mathbf{\Gamma}_{\mathbf{q}}^{-1} \boldsymbol{\mu}_{\mathbf{q}}) \ . \qquad (13)$$

All the quantities required for the M-step below can be expressed in terms of $v_{\mathbf{q}}$, $\boldsymbol{\rho}_{\mathbf{q}}$, and $\mathbf{\Sigma}_{\mathbf{q}}$.

M-Step. Maximization with respect to the model parameters W produces

$$\begin{aligned} \mathbf{H} &= \overline{\mathbf{y}\langle \mathbf{x}^T \rangle} \left(\overline{\langle \mathbf{x}\mathbf{x}^T \rangle} \right)^{-1} \ , \\ \mathbf{\Lambda} &= \overline{\mathbf{y}\mathbf{y}^T} - \overline{\mathbf{y}\langle \mathbf{x}^T} \rangle \mathbf{H}^T \end{aligned} \qquad (14)$$

for the mixing and noise parameters, and

$$\begin{aligned} &\pi^i_q = \overline{v^i_q} \ , \\ &\mu^i_q = \frac{1}{\pi^i_q} \overline{\langle x_i \rangle_q} \ , \qquad \gamma^i_q = \frac{1}{\pi^i_q} \overline{\langle x_i^2 \rangle_q} - \mu^2_{i,q} \end{aligned} \qquad (15)$$

for the factor parameters. In terms of the E-step quantities, the conditional factor mean is given by

$$\langle \mathbf{x} \rangle = \sum_{\mathbf{q}} v_{\mathbf{q}} \boldsymbol{\rho}_{\mathbf{q}} \ , \qquad (16)$$

and the conditional factor covariance by

$$\langle \mathbf{x}\mathbf{x}^T \rangle = \sum_{\mathbf{q}} v_{\mathbf{q}} (\boldsymbol{\rho}_{\mathbf{q}} \boldsymbol{\rho}_{\mathbf{q}}^T + \mathbf{\Sigma}_{\mathbf{q}}) \ . \qquad (17)$$

The state-conditioned averages are given by

$$\begin{aligned} \langle x_i \rangle_q &= \sum_{\mathbf{q}}^{i} v_{\mathbf{q}} (\boldsymbol{\rho}_{\mathbf{q}})_i \ , \\ \langle x_i^2 \rangle_q &= \sum_{\mathbf{q}}^{i} v_{\mathbf{q}} (\boldsymbol{\rho}_{\mathbf{q}} \boldsymbol{\rho}_{\mathbf{q}}^T + \mathbf{\Sigma}_{\mathbf{q}})_{ii} \ , \end{aligned} \qquad (18)$$

where $\sum_{\mathbf{q}}^i$ denotes summation over $\{q_{j \neq i}\}$, holding $q_i = q$ fixed. Finally, $v^i_q = \sum_{\mathbf{q}}^i v_{\mathbf{q}}$.

We point out that in the limit $\mathbf{\Lambda} \to 0$, this algorithm does *not* reduce to the noise-free separation algorithm from the previous section. In fact, the rule for the mixing matrix becomes $\mathbf{H} \to \mathbf{C}_{\mathbf{y}} \mathbf{H} (\mathbf{H}^T \mathbf{C}_{\mathbf{y}} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{H}$, where $\mathbf{C}_{\mathbf{y}} = E\mathbf{y}\mathbf{y}^T$ is the data covariance matrix. This rule can be shown to perform principal component analysis (PCA): after learning, $\mathbf{H}$ will contain the eigenvectors of $\mathbf{C}_{\mathbf{y}}$ that correspond to its largest L eigenvalues. The resulting EM algorithm for PCA has been independently discovered by [14] and by [15].

Factor Reconstruction. There are two special cases where the factors are reconstructed from the data by a linear estimator: the noise-free case discussed above, where $\hat{\mathbf{x}} = \mathbf{G}\mathbf{y}$, and the noisy case with Gaussian factors where $\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{H} + \mathbf{I})^{-1} \mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{y}$. In general, a linear estimator is sub-optimal. Here we consider two non-linear factor estimators, based on different optimality criteria. The first is the maximum a-posteriori probability (MAP) estimator $\hat{\mathbf{x}}^{MAP}$, obtained by maximizing the factor posterior $p(\mathbf{x} \mid \mathbf{y})$ with respect to $\mathbf{x}$. A gradient-ascent learning rule can be derived and is given by

$$\delta\hat{\mathbf{x}} = \epsilon\mathbf{H}^T\mathbf{\Lambda}^{-1}(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) - \epsilon\phi(\hat{\mathbf{x}}) \ , \qquad (19)$$

where ϕ is the negative log-derivative of the factor density as before, and ϵ is a properly chosen learning rate.

Alternatively, one may use the least mean squares (LMS) estimator, which minimizes the error $E(\hat{\mathbf{x}}-\mathbf{x})^2$. The LMS estimator is given by $\hat{\mathbf{x}}^{LMS} = \langle\mathbf{x}\rangle$, where the conditional mean (16) is as a sum over state configurations $\mathbf{q}$ of the terms $\boldsymbol{\rho}_\mathbf{q}$ that are linear in $\mathbf{y}$, weighted by the terms $v_\mathbf{q}$ that are non-linear in $\mathbf{y}$.

Many Factors. As the number L factors increases, the E-step becomes intractable, since the number $\prod_i n_i$ of factor state configurations $\mathbf{q} = (q_1, ..., q_L)$ depends exponentially on L. Such cases are treated in [11] using a variational approximation. Below we present a variational approach for the hierarchical version of IFA, which is intractable even for a small number of factors.

3 HIERARCHICAL EXTENSION

In the following we develop a multilayer generalization of IFA, by cascading duplicates of the generative model reviewed above. Each layer $n = 1, ..., N$ is composed of two sublayers: a factor sublayer which consists of the units x_i^n, $i = 1, ..., L_n$, and an output sublayer which consists of y_j^n, $j = 1, ..., L'_n$. The two are linearly related via $\mathbf{y}^n = \mathbf{H}^n\mathbf{x}^n + \mathbf{u}^n$ as in (1); $\mathbf{u}^n$ is a Gaussian noise vector with covariance $\mathbf{\Lambda}^n$. The nth-layer factor x_i^n is described by a MOG density model with parameters $a_{i,q}^n$, $\mu_{i,q}^n$, and $\gamma_{i,q}^n$, in analogy to the IFA factors above.

The important step is to determine how layer n depends on the previous layers. We choose to introduce a dependence of the ith factor of layer n only on the ith output of layer $n-1$. Notice that matching $L_n = L'_{n-1}$ is now required. This dependence is implemented by making the means and mixture proportions of the Gaussians which compose $p(x_i^n)$ dependent on y_i^{n-1}. Specifically, we make the replacements

$$\begin{aligned} \mu_{i,q}^n &\to \mu_{i,q}^n + \nu_{i,q}^n y_i^{n-1} \ , \\ a_{i,q}^n &\to a_{i,q}^n + b_{i,q}^n y_i^{n-1} \ . \end{aligned} \qquad (20)$$

The resulting joint density for layer n, conditioned on layer $n-1$, is

$$p(\mathbf{q}^n, \mathbf{x}^n, \mathbf{y}^n \mid \mathbf{y}^{n-1}, W^n)$$

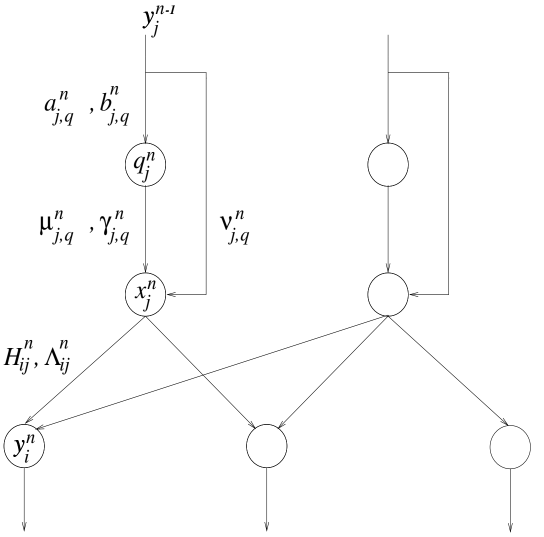
y_j^{n-1}
$a_{j,q}^n$, $b_{j,q}^n$
q_j^n
$\mu_{j,q}^n$, $\gamma_{j,q}^n$
$\nu_{j,q}^n$
x_j^n
H_{ij}^n, Λ_{ij}^n
y_i^n

Figure 2: Layer n of the hierarchical IFA generative model.

$$= \prod_{i=1}^{L_n} p(q_i^n \mid y_i^{n-1}) \ p(x_i^n \mid q_i^n, y_i^{n-1}) \ p(\mathbf{y}^n \mid \mathbf{x}^n) \ , \quad (21)$$

where W^n are the parameters of layer n and

$$\begin{aligned} p(q_i^n = q \mid y_i^{n-1}) &= \pi_{i,q}^n(y_i^{n-1}) \\ &= \frac{\exp(a_{i,q}^n + b_{i,q}^n y_i^{n-1})}{\sum_{q'} \exp(a_{i,q'}^n + b_{i,q'}^n y_i^{n-1})} \ , \\ p(\mathbf{y}^n \mid \mathbf{x}^n) &= \mathcal{G}(\mathbf{y}^n - \mathbf{H}^n\mathbf{x}^n, \mathbf{\Lambda}^n) \ , \\ p(x_i^n \mid q_i^n = q, y_i^{n-1}) &= \mathcal{G}(x_i^n - \mu_{i,q}^n - \nu_{i,q}^n y_i^{n-1}, \gamma_{i,q}^n) \ . \end{aligned}$$

The full model joint density is given by the product of (21) over the layers,

$$\begin{aligned} & p(\mathbf{q}^{1...N}, \mathbf{x}^{1...N}, \mathbf{y}^{1...N} \mid W^{1...N}) \\ & = \prod_{n=1}^{N} p(\mathbf{q}^n, \mathbf{x}^n, \mathbf{y}^n \mid \mathbf{y}^{n-1}, W^n) \ , \end{aligned} \qquad (22)$$

setting $\mathbf{y}^0 = 0$. A graphical representation of layer n of the hierarchical IFA network is given in Fig. 2. All units are hidden except $\mathbf{y}^N$.

To gain some insight into our network, we examine the relation between the nth-layer factor x_i^n and the $n-1$th-layer output y_i^{n-1}. This relation is probabilistic and is determined by the conditional density $p(x_i^n \mid y_i^{n-1}) = \sum_{q_i^n} p(q_i^n \mid y_i^{n-1}) p(x_i^n \mid q_i^n, y_i^{n-1})$. Notice from (21) that this is a MOG density whose parameters depend on y_i^{n-1}. In particular, its mean is given by

$$\overline{x_i^n} = f_i^n(y_i^{n-1}) = \sum_q \pi_{i,q}^n(y_i^{n-1})(\mu_{i,q}^n + \nu_{i,q}^n y_i^{n-1}) \ , (23)$$

and is a non-linear function of y_i^{n-1} due to the softmax form of $\pi_{i,q}^n$. By adjusting the parameters, the function

f_i^n can assume a very wide range of forms: suppose that for state q_i^n, $a_{i,q}^n$ and $b_{i,q}^n$ are set so that $\pi_{i,q}^n(y_i^{n-1})$ is significant only in a small, continuous range of y_i^{n-1} values, with different ranges associated with different q's. In this range, f_i^n will be dominated by the linear term $\mu_{i,q}^n + \nu_{i,q}^n y_i^{n-1}$. Hence, a desired f_i^n can be produced by placing oriented line segments at appropriate points above the y_i^{n-1}-axis, then smoothly join them together by the $\pi_{i,q}^n$. Using the algorithm below, the optimal form of f_i^n will be learned from the data. Therefore, our model describes the data y_i^N as a potentially highly complex function of the top layer factors, produced by repeated application of linear mixing followed by a non-linearity, with noise allowed at each stage:

$$\begin{aligned} y_i^n &= \sum_j H_{ij}^n x_j^n + u_i^n \ , \\ x_j^n &= f_j^n(y_j^{n-1}) + \xi_j^n \ . \end{aligned} \tag{24}$$

Notice that, unlike the 'mixing noise' u_i^n, whose density is Gaussian depends only on the model parameters, the 'non-linearity noise' ξ_j^n has a complex density which depends on y_j^{n-1} as well.

4 VARIATIONAL LEARNING AND INFERENCE

The need for summing over an exponentially large number of factor state configurations $(q_1^n, ..., q_L^n)$, and integrating over the softmax functions $\pi_{i,q}^n(y_i^{n-1})$, makes exact learning intractable in our network. Thus, approximations must be made. In the following we develop a variational approach to hierarchical IFA. Our approach is inspired by the work of [18,19,20] on variational approximations of learning and inference in several intractable probabilistic models, and the related work [16,17] on the Helmholtz machine. However, the approach presented here is more powerful, in that it does not rely on complete factorization and can handle more complex non-linearities (e.g., the softmax function).

We begin, following [13], by bounding the log-likelihood of the observed data from below:

$$\mathcal{L} = \log p(\mathbf{y}^N) \geq \sum p' \log \frac{p}{p'} \ , \tag{25}$$

where $p = p(\mathbf{q}^{1 \cdots N}, \mathbf{x}^{1 \cdots N}, \mathbf{y}^{1 \cdots N})$ is the generative model density defined by (21–22), and $p' = p'(\mathbf{q}^{1 \cdots N}, \mathbf{x}^{1 \cdots N}, \mathbf{y}^{1 \cdots N-1} \mid \mathbf{y}^N)$ is an arbitrary posterior density over the hidden layers. If, for any function $g(\mathbf{q}^{1 \cdots N}, \mathbf{x}^{1 \cdots N}, \mathbf{y}^{1 \cdots N})$, we denote by Eg the outcome of averaging over the hidden layers using p', we have

$$\begin{aligned} \mathcal{L} = \log p(\mathbf{y}^N) &\geq \sum_n E \log p(\mathbf{y}^n \mid \mathbf{x}^n) \\ &+ \sum_{n,i,q_i^n} [E \log p(x_i^n \mid q_i^n, y_i^{n-1}) + E \log p(q_i^n \mid y_i^{n-1})] \\ &- E \log p' \ . \end{aligned} \tag{26}$$

In exact EM, p' at each iteration is the true posterior, parametrized by $W^{1 \cdots N}$ from the previous iteration. In variational EM, p' is chosen to have a form which makes learning tractable, and is parametrized by a separate set of 'variational' parameters $V^{1 \cdots N}$. These are optimized to bring p' as close to the true posterior as possible. The optimization is done by maximizing the lower bound (25) on the likelihood (as for the generative parameters $W^{1 \ldots N}$), or, equivalently, by minimizing the Kullback-Leibler divergence between p' and the true posterior.

E-step. We use a variational posterior that is factorized across layers, but not within layers: within layer n it has the form

$$\begin{aligned} & p'(\mathbf{q}^n, \mathbf{x}^n, \mathbf{y}^n \mid \mathbf{y}^N, V^n) \\ & = \prod_{i=1}^{L_n} v_{i,q_i}^n \ \mathcal{G}(\mathbf{z}^n - \boldsymbol{\rho}^n, \boldsymbol{\Sigma}^n) \ , \qquad \mathbf{z}^n = (\mathbf{x}^n, \mathbf{y}^n)^T \end{aligned} \tag{27}$$

for layers $n < N$, and

$$p'(\mathbf{q}^N, \mathbf{x}^N \mid \mathbf{y}^N, V^N) = \prod_i v_{i,q_i}^N \mathcal{G}(\mathbf{x}^N - \boldsymbol{\rho}^N, \boldsymbol{\Sigma}^N) \tag{28}$$

for the bottom layer. The variational parameters $V^n = (\boldsymbol{\rho}^n, \boldsymbol{\Sigma}^n, \{v_{i,q}^n\})$ depend on the data $\mathbf{y}^N$. The full N-layer posterior is simply the product of (27,28) over n,

$$\begin{aligned} p' &= p'(\mathbf{q}^N, \mathbf{x}^N \mid \mathbf{y}^N, V^N) \\ &\times \prod_{n=1}^{N-1} p'(\mathbf{q}^n, \mathbf{x}^n, \mathbf{y}^n \mid \mathbf{y}^N, V^n) \ . \end{aligned} \tag{29}$$

Hence, given the data, the nth-layer factors and outputs are jointly Gaussian whereas the states q_i^n are independent.

We point out that it is possible to introduce more structure into (27) by allowing the means and diagonal covariances of the Gaussians to depend on the states q_i^n. The mean simply becomes $\boldsymbol{\rho}^n \to \boldsymbol{\rho}_{\mathbf{q}}^n$. It is more difficult to make the covariance $\mathbf{q}$-dependent since simply changing $\boldsymbol{\Sigma}^n \to \boldsymbol{\Sigma}_{\mathbf{q}}^n$ would render the algorithm intractable due to the need to sum over all state configurations. Instead, one can use the substitution $\boldsymbol{\Sigma}^n \to \boldsymbol{\Theta}_{\mathbf{q}}^n \boldsymbol{\Sigma}^n \boldsymbol{\Theta}_{\mathbf{q}}^n$, where $\boldsymbol{\Theta}_{\mathbf{q}}^n$ is a diagonal matrix. In this case, the correlation between factors i, j in layer n, given the data $\mathbf{y}^N$ and the states $\mathbf{q}^n$ of all layer n factors, would depend only on their own states q_i^n, q_j^n. This variational approximation would be more accurate than (27) (but also more complex) while maintaining tractability.

Even with the variational posterior (27), the term $E \log p(q_i^n \mid y_i^{n-1})$ in the lower bound cannot be calculated analytically, since it involves integration over the softmax function. Instead, we calculate yet a lower bound on this term. Let us define

$$c_{i,q}^n = a_{i,q}^n + b_{i,q}^n y_i^{n-1} \tag{30}$$

and drop the unit and layer indices i, n, then

$$\log p(q \mid y) = -\log(1 + e^{-c_q^n} \sum_{q' \neq q} e^{c_{q'}}) \ . \tag{31}$$

Borrowing an idea from [19], we multiply and divide by e^{η_q} under the logarithm sign and use Jensen's inequality to get

$$\begin{aligned} &E \log p(q \mid y) \qquad\qquad\qquad\qquad\qquad\qquad (32) \\ &\geq -\eta_q E c_q - \log E \left[e^{-\eta_q c_q} + e^{-(1+\eta_q) c_q} \sum_{q' \neq q} e^{c_{q'}} \right] \ . \end{aligned}$$

This results in a bound that can be calculated in closed form:

$$\begin{aligned} &E \log p(q_i^n = q \mid y_i^{n-1}) \qquad\qquad\qquad\qquad (33) \\ &\geq -v_q^n \eta_q^n \bar{c}_q^n - v_q^n \log \left(e^{f_q^n} + \sum_{q' \neq q} e^{f_{qq'}^n} \right) \equiv \mathcal{F}_{i,q}^n \ , \end{aligned}$$

where

$$\begin{aligned} \bar{c}_q^n &= a_q^n + b_q^n \rho_y^{n-1} \ , \\ f_q^n &= -\eta_q^n \bar{c}_q^n + (\eta_q^n b_q^n)^2 \Sigma_{yy}^{n-1}/2 \ , \\ f_{qq'}^n &= -(1 + \eta_q^n) \bar{c}_q^n + \bar{c}_{q'}^n \\ &\quad + [(1 + \eta_q^n) b_q^n - b_{q'}^n]^2 \Sigma_{yy}^{n-1}/2 \ , \end{aligned} \tag{34}$$

and the subscript i is omitted. We also defined

$$\boldsymbol{\rho}^n = (\boldsymbol{\rho}_x^n, \boldsymbol{\rho}_y^n)^T \ , \tag{35}$$

and similarly $\mathbf{\Sigma}_{xx}^n, \mathbf{\Sigma}_{yy}^n, \mathbf{\Sigma}_{xy}^n = \mathbf{\Sigma}_{yx}^{n\ T}$ are the subblocks of $\mathbf{\Sigma}^n$ corresponding to (35). Since (33) holds for arbitrary $\eta_{i,q}^n$, the latter are treated as additional variational parameters which are optimized to tighten this bound.

An alternative approach to handle $E \log p(q_i^n \mid y_i^{n-1})$ is to approximate the required integral by, e.g., the maximum value of the integrand, possibly including Gaussian corrections. The resulting approximation is simpler than (33); however, it is no longer guaranteed to bound the log-likelihood from below.

To optimize the variational parameters $V^{1\cdots N}$, we equate the gradient of the lower bound on $\mathcal{L}$ to zero and obtain

$$\begin{aligned} & \begin{pmatrix} (\mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{H})^n + \mathbf{A}^n & -(\mathbf{H}^T \mathbf{\Lambda}^{-1})^n \\ -(\mathbf{\Lambda}^{-1} \mathbf{H})^n & (\mathbf{\Lambda}^{-1})^n + \mathbf{B}^{n+1} \end{pmatrix} \boldsymbol{\rho}^n \\ - & \begin{pmatrix} 0 & \mathbf{B}^n \\ \mathbf{A}^{n+1} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\rho}_x^{n+1} \\ \boldsymbol{\rho}_y^{n-1} \end{pmatrix} \\ = & \begin{pmatrix} \boldsymbol{\alpha}^n \\ \boldsymbol{\beta}^{n+1} + \mathcal{F}_\rho^{n+1} \end{pmatrix} \ , \end{aligned} \tag{36}$$

$$\begin{aligned} &\mathbf{\Sigma}^n = \\ &\begin{pmatrix} (\mathbf{H}^T \mathbf{\Lambda}^{-1} \mathbf{H})^n + \mathbf{A}^n & -(\mathbf{H}^T \mathbf{\Lambda}^{-1})^n \\ -(\mathbf{\Lambda}^{-1} \mathbf{H})^n & (\mathbf{\Lambda}^{-1})^n + \mathbf{B}^{n+1} - \mathcal{F}_\Sigma^{n+1} \end{pmatrix}^{-1} , \end{aligned}$$

where we define

$$\begin{aligned} A_{ij}^n &= \sum_q \frac{v_{i,q}^n}{\gamma_{i,q}^n} \delta_{ij} \ , \\ B_{ij}^n &= \sum_q \frac{v_{i,q}^n \nu_{i,q}^n}{\gamma_{i,q}^n} \delta_{ij} \ , \\ \alpha_i^n &= \sum_q \frac{v_{i,q}^n \mu_{i,q}^n}{\gamma_{i,q}^n} \ , \\ \beta_i^n &= \sum_q \frac{v_{i,q}^n \mu_{i,q}^n \nu_{i,q}^n}{\gamma_{i,q}^n} \ . \end{aligned} \tag{37}$$

$F_{\rho,\Sigma}^{n+1}$ contain the derivatives of $\mathcal{F}_q^{n+1}$ (33) with respect to $\boldsymbol{\rho}^{n+1}$ and $\mathbf{\Sigma}^{n+1}$, summed over q. For the state posteriors we have

$$\begin{aligned} &v_q^n = \frac{1}{Z^n} \exp \left(\frac{\gamma_q^n}{2} + \frac{\partial \mathcal{F}_q^n}{\partial v_q^n} \right. \qquad\qquad\qquad (38) \\ &\left. + \frac{1}{2\gamma_q^n} [(\rho_x^n - \mu_q^n - \nu_q^n \rho_y^{n-1})^2 + \Sigma_{xx}^n + (\nu_q^n)^2 \Sigma_{yy}^{n-1}] \right) \end{aligned}$$

where the unit subscript i is omitted (i.e., $\Sigma_{xx}^n = \Sigma_{xx,ii}^n$); $Z^n = Z_i^n$ is set such that $\sum_q v_{i,q}^n = 1$. A simple modification of these equations is required for layer $n = N$.

The optimal $V^{1\cdots N}$ are obtained by solving the fixed-point equations (36–38) iteratively for each data vector $\mathbf{y}^N$, keeping the generative parameters $W^{1\cdots N}$ fixed. Notice that these equations couple layer n to layers $n \pm 1$, hence although the variational posterior is factorized over the layers, its parameters are determined by the whole network. The additional parameters $\eta_{i,q}^n$ are adjusted using gradient ascent on $\mathcal{F}_{i,q}^n$. Once learning is complete, the inference problem is solved since the MAP estimate of the hidden unit values given the data is readily available from ρ_i^n and $v_{i,q}^n$.

M-Step. In terms of the variational parameters obtained in the E-step, the new generative parameters are given by

$$\begin{aligned} \mathbf{H}^n &= (\boldsymbol{\rho}_y^n \boldsymbol{\rho}_x^{n\ T} + \mathbf{\Sigma}_{yx}^n)(\boldsymbol{\rho}_x^n \boldsymbol{\rho}_x^{n\ T} + \mathbf{\Sigma}_{xx}^n)^{-1} \ , \\ \mathbf{\Lambda}^n &= \boldsymbol{\rho}_y^n \boldsymbol{\rho}_y^{n\ T} + \mathbf{\Sigma}_{yy}^n - \mathbf{H}^n (\boldsymbol{\rho}_x^n \boldsymbol{\rho}_x^{n\ T} + \mathbf{\Sigma}_{xy}^n) \end{aligned} \tag{39}$$

and

$$\begin{pmatrix} \mu_q^n \\ \nu_q^n \end{pmatrix} = \begin{pmatrix} v_q^n & \rho_y^{n-1} v_q^n \\ \rho_y^{n-1} v_q^n & [(\rho_y^{n-1})^2 + \Sigma_{yy}^{n-1}] v_q^n \end{pmatrix}^{-1} \times \begin{pmatrix} \rho_x^n v_q^n \\ \rho_x^n \rho_y^{n-1} v_q^n \end{pmatrix} , \qquad (40)$$

$$\gamma_q^n = \frac{1}{v_q^n} \times \left[(\rho_x^n - \mu_q^n - \nu_q^n \rho_y^{n-1})^2 + \Sigma_{xx}^n + (\nu_q^n)^2 \Sigma_{yy}^{n-1} \right] v_q^n ,$$

omitting the subscript i as in (38), and are slightly modified for layer N. The overlines denoting averaging over the observed data are also omitted; due to the implied averaging, the v_q^n in (40) do not cancel out. Finally, the softmax parameters $a_{i,q}^n, b_{i,q}^n$ are adapted by gradient ascent on the bound (33).

5 DISCUSSION

The hierarchical IFA network presented here constitutes a quite general framework for learning and inference using probabilistic models that are strongly nonlinear but highly adaptive. Our formulation can easily be generalized beyond the multilayer architecture to any directed acyclic graph. Notice that this network includes both continuous x_i^n, y_i^n and multinomial q_i^n units, and can thus extract both types of latent variables. Its single-layer version, the IFA algorithm, generalizes and unifies ICA, PCA and FA. The rectified Gaussian belief networks [21] and non-linear Gaussian belief networks [22], two recently proposed multilayer probabilistic models for real-valued data, can be viewed as special cases of hierarchical IFA, where x_i^n is a prescribed deterministic function (e.g., rectifier) of the previous outputs y_j^{n-1}, i.e., the non-linear function f_j^n (24) is fixed and the non-linearity noise ξ_j^n vanishes.

The learning algorithm presented here has the technical advantage of using a variational posterior that allows correlations among hidden units occupying the same layer, thus providing a more accurate description of the true posterior than in the completely factorized approximation [19,20,22].

We point out that this model can be easily incorporated into a Bayesian classification algorithm, where hierarchical IFA would described the class-conditional densities. Another possible extension includes constructing multilayer networks with more than a single IFA module in each layer, such that different modules describe different regions of the data, e.g., spatial regions in images or spectro-temporal regions in speech.

References

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

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

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

[4] Amari, S., Cichocki, A. & Yang, H.H. (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems* **8**, 757-763 (Touretzky et al, Eds). MIT Press, Cambridge, MA.

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

[6] Pearlmutter, B.A. & Parra, L.C. (1997). Maximum likelihood blind source separation: A context-sensitive generalization of ICA. *Advances in Neural Information Processing Systems* **9**, 613-619 (Mozer, M.C. et al, Eds). MIT Press, Cambridge, MA.

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

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

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

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

[11] Attias, H. (1998). Independent factor analysis. *Neural Computation*, in press.

[12] Lewicki, M.S. & Sejnowski, T.J. (1998). Learning nonlinear overcomplete representations for efficient coding. *Advances in Neural Information Processing Systems* **10**, 556-562 (Jordan M.I. et al, Eds). MIT Press, Cambridge, MA.

[13] Neal, R.M. & 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.

[14] Roweis, S. (1998). EM algorithms for PCA and SPCA. *Advances in Neural Information Processing*

Systems **10**, 626-632. (Jordan M.I. et al, Eds). MIT Press, Cambridge, MA.

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

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

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

[18] Saul, L. & Jordan, M.I. (1995). Exploiting tractable structures in intractable networks. *Advances in Neural Information Processing Systems* **8**, 486-492 (Touretzky, D.S. et al, Eds). MIT Press, Cambridge, MA.

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

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

[21] Ghahramani, Z. & Hinton, G.E. (1998). Hierarchical non-linear factor analysis and topographic maps. *Advances in Neural Information Processing Systems* **10**, 486-492 (Jordan M.I. et al, Eds). MIT Press, Cambridge, MA.

[22] Frey, B.J. & Hinton, G.E. (1998). Variational learning in non-linear Gaussian belief networks. *Neural Computation*, in press.