Independent Component Analysis of Incomplete Data

Max Welling *Markus Weber*

California Institute of Technology 136-93
Pasadena, CA 91125
{welling,rmw}@vision.caltech.edu

Keywords: EM, Missing Data, ICA

Abstract

Realistic data often exhibit arbitrary patterns of missing features, due to occluded or imperfect sensors. We contribute a constrained version of the expectation maximization (EM) algorithm which fits a model of independent components to the data. This raises the possibility to perform independent component analysis (ICA) on incomplete observations. In the case of complete data, our algorithm represents an alternative to *independent factor analysis*, without the requirement of a large number of Gaussian mixture components that grows exponentially with the number of data dimensions. The performance of our algorithm is demonstrated experimentally.

1 Introduction

Independent component analysis has recently grown popular as a technique of estimating distributions of multivariate random variables which can be modelled as linear combinations of independent sources. In this sense, it is an extension of PCA and factor analysis which takes higher order statistical information into account.

Many approaches to estimating independent components have been put forward, a few of which are Comon (Comon, 1994), Hyvärinen (Hyvärinen, 1997), Girolami and Fyfe (Girolami and Fyfe, 1997), Pearlmutter and Parra (Pearlmutter and Parra, 1996) and Bell and Sejnowski (Bell and Sejnowski, 1995).

ICA has also proven useful as a practical tool in signal processing. Applications can be found in the field of blind source separation, denoising, pattern recognition, image processing and medical signal processing.

In this paper we address the problem of estimating independent components from incomplete data. This problem is important when only sparse data are available because occlussions or noise have corrupted the data. We previously introduced a constrained EM algorithm to estimate mixing matrices. In this paper, we will extend this method to handle incomplete data, and show its performance on real and artificial datasets.

2 Independent Component Analysis

Independent component analysis is typically employed to analyze data from a set of statistically independent sources. Let $s_i, \quad i = 1, ..., D$, denote a scalar random variable representing source i, which we assume to be distributed according to a probability density $p_i(s_i)$. Instead of observing the sources directly, we only have access to the data $x_j, \quad j = 1, \ldots, K$, produced by K sensors which are assumed to capture a *linear* mixture of the source signals,

$$\mathbf{x} = M\,\mathbf{s}, \tag{1}$$

where M is the *mixing matrix*. The task of ICA can be formally stated as follows.

Given a sequence of N data vectors $\mathbf{x}_n,\ \ n = 1, ..., N$, retrieve the mixing matrix, M, and the original source sequence, $\mathbf{s}_n,\ \ n = 1, ..., N$.

This is only possible up to a permutation and scaling of the original source data.

If $\mathbf{u}$ is an estimate of the unmixed sources, then the Kullback-Leibler distance between $p(\mathbf{u})$ and $\prod_{i=1}^{D} p_i(u_i)$ is a natural measure of the independence of the sources. Most methods minimize this *contrast function*, either directly or indirectly. In (Hyvärinen, 1997) a fixed point algorithm was used to this end. Another possibility is to expand the KL-distance in cumulants and use the tensorial properties of the cumulants to device a Jacobi algorithm, as was done in (Comon, 1994) and (Cardoso, 1999). In a third approach, the source estimates are passed through a nonlinearity, while the entropy of the resulting data serves as an objective to be maximized (Bell and Sejnowski, 1997). If the nonlinearity is chosen as to resemble the cumulative distribution functions of the sources, then the correctly unmixed data follow a uniform density and have therefore maximal entropy.

We will adopt the point of view put forward in (Pearlmutter and Parra, 1996) in that we postulate a factorial model for the sources densities. During ICA, this *generative model* is fit to the data using *expectation maximization*.

3 Model of Independent Sources

We describe in this section the model fit to the data for the purpose of ICA. As mentioned above, the sources, s_i, are assumed to be independent random variables. The pdf of every source is modeled through a mixture of M Gaussians,

$$p(\mathbf{s}) = \prod_{i=1}^{D} p_i(s_i) = \prod_{i=1}^{D} \sum_{a=1}^{M} \pi_i^a \ \mathcal{G}_{s_i}[\mu_i^a, (\sigma_i^a)^2]. \tag{2}$$

Here, $\mathcal{G}_x[\mu, \sigma^2]$ stands for a Gaussian pdf over x with mean μ and variance σ^2. Without loss of generality, we assume that the densities $p(s_i)$ have unit variance, since any scale factor can be absorbed into the elements on the diagonal of the mixing matrix. Aside from this constraint, the choice of the parameters for the mixture coefficients is entirely free. Thus every source density can be different, including super-Gaussian and sub-Gaussian densities. Once chosen, these parameters are not updated during the EM procedure.

In the general ICA setting, finding the mixing matrix is equivalent to first recovering a *sphering matrix*, L, and then finding an orthonormal *rotation matrix*, A, such that the mixing matrix can be written as $M = L^{-1}A$. In our model we assume that data is generated by adding zero-mean isotropic Gaussian noise with variance β^2 after applying an orthogonal matrix A to the source data. The observed data are then obtained by multiplying with L^{-1}. This process is summarized by the following equality,

$$\mathbf{z} = L\mathbf{x} = A\,\mathbf{s} + \mathbf{n}, \qquad \mathbf{n} \sim \mathcal{G}_{\mathbf{n}}[0, \beta^2 I]. \tag{3}$$

Note that the noise is not added to simulate actual noise, but is rather a necessary ingredient for the proper functioning of the EM algorithm, which would be stuck at a fixed point in the limit of $\beta \to 0$. It was found that the estimation of the noise parameter, β, does not influence the estimation of the mixing matrix A for a wide range of values. We therefore fix this parameter before EM.

Note also that the $\mathbf{x}$ are still zero-mean which is not a limitation, since we can center the data before we perform ICA.

From (3) it is rather obvious that A is indeed orthogonal,

$$\mathbf{E}[\mathbf{z}\mathbf{z}^T] = A\,\mathbf{E}[\mathbf{s}\mathbf{s}^T]\,A^T + \mathbf{E}[\mathbf{n}\mathbf{n}^T] \quad \Rightarrow \quad I = AA^T + \beta^2 I \tag{4}$$

Assuming invertibility for A (the number of sources has to be equal to the number of sensors) we find,

$$AA^T = A^T A = (1 - \beta^2) I. \tag{5}$$

This constraint is crucial for the following exposition, for it allows to derive a *factorized* posterior density $p(\mathbf{s}|\mathbf{x})$ in the case of complete data.

For EM, instead of directly maximizing the log-likelihood over the observed data, one maximizes the expectation of the log of the joint density, $\log p(\mathbf{x}, \mathbf{s})$, over the posterior density $p(\mathbf{s}|\mathbf{x})$. Calculating the posterior is typically the most challenging part in deriving an EM algorithm, and often one has to resort to Gibbs sampling or other approximation methods. In our case, we can compute this posterior analytically.

We start with Bayes rule,

$$p(\mathbf{s}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{s})\ p(\mathbf{s})}{\int d\mathbf{s}\ p(\mathbf{x}|\mathbf{s})\ p(\mathbf{s})}, \tag{6}$$

where

$$p(\mathbf{x}|\mathbf{s}) = \mathcal{G}_{L\mathbf{x}}[A\mathbf{s}, \beta^2 I]\ \det L = \frac{\det L}{\sqrt{1-\beta^2}}\ \mathcal{G}_{\mathbf{s}}[\mathbf{u}, \frac{\beta^2}{1-\beta^2} I], \qquad \mathbf{u} = A^{-1} L\mathbf{x}. \tag{7}$$

Note that the aforementioned constraint was used to derive (7). Because this conditional density factors into a product of D functions over $\mathbf{s}$, it follows that $p(\mathbf{x})$ can be calculated from the solutions of D one-dimensional integrals. Also note that, for a more complicated noise model, or unsphered data, we need to evaluate M^D D-dimensional integrals. For the same reasons, the posterior density $p(\mathbf{s}|\mathbf{x})$ factors as well and can be calculated with relatively little effort,

$$\begin{aligned}
p(\mathbf{s}|\mathbf{x}) &= \prod_{i=1}^{D} \sum_{a=1}^{M} \alpha_i^a\ \mathcal{G}_{s_i}[b_i^a, (\gamma_i^a)^2] && (8)\\
(\gamma_i^a)^2 &= \frac{\beta^2\ (\sigma_i^a)^2}{(1-\beta^2)(\sigma_i^a)^2 + \beta^2} && (9)\\
b_i^a &= (\gamma_i^a)^2 (\frac{1-\beta^2}{\beta^2} u_i + \frac{\mu_i^a}{(\sigma_i^a)^2}) && (10)\\
\alpha_i^a &= \frac{\pi_i^a\ \mathcal{G}_{u_i}[\mu_i^a, \frac{\beta^2}{1-\beta^2} + (\sigma_i^a)^2]}{\sum_{b=1}^{M} \pi_i^b\ \mathcal{G}_{u_i}[\mu_i^b, \frac{\beta^2}{1-\beta^2} + (\sigma_i^b)^2]} && (11)
\end{aligned}$$

and $\mathbf{u} = A^{-1}\ L\ \mathbf{x}$.

4 Missing Data

In order to include missing data in our generative model we split each data vector into a missing part and an observed part: $\mathbf{x}_n^T = [\mathbf{x}_n^{\mathbf{m}}, \mathbf{x}_n^{\mathbf{o}}]^T$. This split is different for each data point but we will not denote this explicitly. We can introduce the following shift in variables,

$$\begin{aligned}
\mathbf{x}_n^{\mathbf{m}} &\rightarrow \mathbf{y}_n = \mathbf{x}_n^{\mathbf{m}} + L_{\mathbf{m}}^{+}(L_{\mathbf{o}}\mathbf{x}_n^{\mathbf{o}} - A\mathbf{s}_n), && (12)\\
\mathbf{s}_n &\rightarrow \mathbf{s}_n, && (13)
\end{aligned}$$

where $L_{\mathbf{m}}$ and $L_{\mathbf{o}}$ are produced by deleting the columns corresponding to missing and observed data respectively. The matrix $L_{\mathbf{m}}^{+} = (L_{\mathbf{m}}^T L_{\mathbf{m}})^{-1} L_{\mathbf{m}}^T$ is the pseudo-inverse of $L_{\mathbf{m}}$. Note that the Jacobian of this transformation is equal to one. The merit of the change in variables becomes clear when we rewrite $p(\mathbf{x}|\mathbf{s})$,

$$p(\mathbf{x}|\mathbf{s}) = \mathcal{G}_{L\mathbf{x}}[A\mathbf{s}, \beta^2] = (2\pi\beta^2)^{-\frac{D}{2}} \exp[-\frac{1}{2\beta^2}(\mathbf{y}^T\ L_{\mathbf{m}}^T L_{\mathbf{m}}\ \mathbf{y})] \tag{14}$$

$$\exp[-\frac{1}{2\beta^2}(\mathbf{s} - A^{-1}L_{\mathbf{o}}\mathbf{x}^{\mathbf{o}})^T\ A^T P_{\mathbf{o}} A\ (\mathbf{s} - A^{-1}L_{\mathbf{o}}\mathbf{x}^{\mathbf{o}})] \tag{15}$$

where $P_{\mathbf{o}} = I - P_{\mathbf{m}} = I - L_{\mathbf{m}} L_{\mathbf{m}}^{+}$ is the projection operator that projects vectors onto the subspace orthogonal to the subspace spanned by the columns of $L_{\mathbf{m}}$. In the derivation we used the following properties of $P_{\mathbf{o}}$,

$$P_{\mathbf{o}} = P_{\mathbf{o}}^T \qquad P_{\mathbf{o}} = P_{\mathbf{o}}^2 \qquad L_{\mathbf{m}}^T P_{\mathbf{o}} = P_{\mathbf{o}} L_{\mathbf{m}} = 0 \tag{16}$$

Rewriting the problem in this particular form is useful, because the random vector $\mathbf{y}$ is decoupled from the sources $\mathbf{s}$ as well as the observed part of the data, $\mathbf{x}^{\mathbf{o}}$. However, the fact that the projection operator $P_{\mathbf{o}}$ is not proportional to the identity introduces correlations between the source components, *given* the observed data. This implies that we cannot avoid an exponential increase in the number of Gaussian mixture components, as the number of data dimensions grows. A naive approximation, where $A^T P_{\mathbf{o}} A$ is replaced by its closest diagonal matrix under L_2-norm, did not work satisfactorily in our experiments. More sophisticated approximations, like the one explored in (Attias, 1999), could be very valuable if we want to deal with a large number of sources. In this paper we stayed with the exact formulation for the incomplete data. The complete data points can of course be treated by the discussion of the previous section.

Let us now derive expressions for the posterior densities that are required for the EM algorithm described in the next section. We again use Bayes rule,

$$p(\mathbf{y}, \mathbf{s}|\mathbf{x}^{\mathbf{o}}) = p(\mathbf{y}) \, \frac{p(\mathbf{x}^{\mathbf{o}}|\mathbf{s}) \, p(\mathbf{s})}{\int d\mathbf{s} \; p(\mathbf{x}^{\mathbf{o}}|\mathbf{s}) \, p(\mathbf{s})}. \tag{17}$$

The density over $\mathbf{y}$ is simply a Gaussian: $p(\mathbf{y}) = \mathcal{G}_{\mathbf{y}}[0, (L_{\mathbf{m}}^T L_{\mathbf{m}})^{-1}]$.

The second term in (17) is more difficult due to the integral in the denominator. Remember that $p(\mathbf{s})$ consists of a product of one dimensional Gaussian mixtures (2). However, because $P_{\mathbf{o}}$ is not proportional to the identity, the second exponential in (15) cannot be factorized. Instead, we have to expand this product into a sum over M^D Gaussian components. The mixing coefficients, means and variances of these components are generated by combining the parameters of all source densities. In the following we will use two Gaussian mixture components per source ($M = 2$) and denote them by an index $a = \{0, 1\}$. We then introduce a new index J which is an integer from 0 to $2^D - 1$. We may write this integer in binary notation and denote by J_i the i'th bit. Using this notation we can write the 2^D parameters of the Gaussian mixtures in terms of the parameters of the marginal densities $p(s_i)$,

$$\begin{aligned}
p(\mathbf{s}) &= \sum_{J=0}^{2^D-1} \pi_J \, \mathcal{G}_{\mathbf{s}}[\mu_J, \Sigma_J] && (18)\\
\Sigma_J &= \mathrm{Diag}[(\sigma_1^{J_1})^2, (\sigma_2^{J_2})^2, ..., (\sigma_D^{J_D})^2] && (19)\\
\mu_J^T &= [\mu_1^{J_1}, \mu_2^{J_2}, ..., \mu_D^{J_D}] && (20)\\
\pi_J &= \pi_1^{J_1} \pi_2^{J_2} ... \pi_D^{J_D} && (21)
\end{aligned}$$

Note that, although we have 2^D mixture components, their degrees of freedom are highly constrained due to the parametrization (21). Written this way, the integrals in (17) are 2^D D-dimensional integrals over Gaussians. After some further algebra we are left with the following posterior density $p(\mathbf{s}|\mathbf{x}^{\mathbf{o}})$,

$$\begin{aligned}
p(\mathbf{s}|\mathbf{x}^{\mathbf{o}}) &= \sum_{J=0}^{2^D-1} \alpha_J \, \mathcal{G}_{\mathbf{s}}[\mathbf{b}_J, \Gamma_J] && (22)\\
\Gamma_J^{-1} &= \beta^{-2} A^T P_{\mathbf{o}} A + \Sigma_J^{-1} && (23)\\
\mathbf{b}_J &= \Gamma_J \{\beta^{-2} A^T P_{\mathbf{o}} L_{\mathbf{o}} \mathbf{x}^{\mathbf{o}} + \Sigma_J^{-1} \mu_J\} && (24)\\
\alpha_J &= \frac{\pi_J \sqrt{\frac{\det[\Gamma_J]}{\det[\Sigma_J]}} \exp(\frac{1}{2} \mathbf{b}_J^T \Gamma_J^{-1} \mathbf{b}_J - \frac{1}{2} \mu_J^T \Sigma_J^{-1} \mu_J)}{\sum_{K=0}^{2^D-1} \pi_K \sqrt{\frac{\det[\Gamma_K]}{\det[\Sigma_K]}} \exp(\frac{1}{2} \mathbf{b}_K^T \Gamma_K^{-1} \mathbf{b}_K - \frac{1}{2} \mu_K^T \Sigma_K^{-1} \mu_K)} && (25)
\end{aligned}$$

These posterior densities will be used in the constrained EM algorithm that we explore in section 5.

5 Constrained Expectation Maximization

As mentioned in section 3, our first task is to estimate the mean and covariance from the incomplete data. The mean is subtracted from the data while the covariance matrix is used to compute $L = \Sigma^{-\frac{1}{2}}$ (the sphering matrix). Estimating mean and covariance from incomplete data is discussed, for example, in (Ghahramani and Jordan, 1994). One proceeds by fitting a Gaussian to the data using yet another, albeit simple EM procedure. In the following we will assume that these preprocessing steps have been performed.

In the second stage of the algorithm we estimate the orthogonal matrix A. Instead of directly maximizing the log-likelihood of the observed data, $\mathcal{X}^{\mathbf{o}} = \{\mathbf{x}_1^{\mathbf{o}}, ..., \mathbf{x}_N^{\mathbf{o}}\}$,

$$\mathcal{L}(A|\mathcal{X}^{\mathbf{o}}) = \sum_{n=1}^{N} \log\{p(\mathbf{x}_n^{\mathbf{o}}|A)\}, \tag{26}$$

EM maximizes the posterior average of the joint log-likelihood, denoted by Q,

$$Q(\tilde{A}|A) = \sum_{n=1}^{N} \int d\mathbf{s}_n d\mathbf{x}_n^{\mathbf{m}} \; p(\mathbf{x}_n^{\mathbf{m}}, \mathbf{s}_n|\mathbf{x}_n^{\mathbf{o}}; A) \; \log\{p(\mathbf{x}_n^{\mathbf{m}}, \mathbf{x}_n^{\mathbf{o}}|\mathbf{s}_n; \tilde{A}) \, p(\mathbf{s}_n)\}, \tag{27}$$

where $\tilde{A}$ is the new mixing matrix with respect to which we optimize $Q(\tilde{A}|A)$, while A is the value from the previous iteration, which we assume to be constant in the M-step. The second term in (27), involving the log-prior $\log\{p(\mathbf{s})\}$ does not depend on $\tilde{A}$ and can be ignored for that matter. The first term,

$$Q_1(\tilde{A}|A) = \sum_{n=1}^{N} \int d\mathbf{x}_n^{\mathbf{m}} d\mathbf{s}_n \; p(\mathbf{x}_n^{\mathbf{m}}, \mathbf{s}_n|\mathbf{x}_n^{\mathbf{o}}; A) \; \log\{p(\mathbf{x}_n^{\mathbf{m}}, \mathbf{x}_n^{\mathbf{o}}|\mathbf{s}_n; \tilde{A})\}, \tag{28}$$

is the part to be maximized with respect to $\tilde{A}$. We will introduce the notation $\langle . \rangle$ for the expectation with respect to the posterior density $p(\mathbf{x}_n^{\mathbf{m}}, \mathbf{s}_n|\mathbf{x}_n^{\mathbf{o}})$. Now Q_1 can be rewritten as follows,

$$Q_1(\tilde{A}|A) = -\frac{1}{2}DN\log(2\pi) - \frac{1}{2}DN\log(\beta^2) + N\log\det L - \frac{1}{2\beta^2}\langle \| L\mathbf{x}_n - A\mathbf{s}_n \|^2 \rangle. \tag{29}$$

Taking the derivative with respect to $\tilde{A}$ and equating with zero yields the following update rule,

$$\tilde{A} = \frac{1}{N}\sum_{n=1}^{N} L_{\mathbf{m}}\langle \mathbf{x}_n^{\mathbf{m}} \mathbf{s}_n^T \rangle + L_{\mathbf{o}}\mathbf{x}_n^{\mathbf{o}}\langle \mathbf{s}_n^T \rangle. \tag{30}$$

We still need to project $\tilde{A}$ onto the space of orthogonal matrices satisfying the constraint (5),

$$\tilde{A} \to \sqrt{(1-\beta^2)}\tilde{A}(\tilde{A}^T\tilde{A})^{-\frac{1}{2}} \tag{31}$$

Previously we showed that this rule can be derived using a Lagrange multiplier. In section 4 we performed a shift in variables (12), which implies that we view $\mathbf{x}^{\mathbf{m}}$ as a function of $\mathbf{y}$ and $\mathbf{s}$. Using this in (30), we find

$$\langle \mathbf{x}_n^{\mathbf{m}} \mathbf{s}_n^T \rangle = \langle \mathbf{y}_n \mathbf{s}_n^T \rangle - L_{\mathbf{m}}^{+}(L_{\mathbf{o}}\mathbf{x}_n^{\mathbf{o}}\langle \mathbf{s}_n^T \rangle - A\langle \mathbf{s}_n \mathbf{s}_n^T \rangle). \tag{32}$$

The first term vanishes because $\mathbf{y}_n$ is independent of $\mathbf{s}_n$ and has mean zero. Combining (32) and (30) we finally obtain for the M-step,

$$\tilde{A} = \frac{1}{N}\sum_{n=1}^{N} P_{\mathbf{m}}A\langle \mathbf{s}_n \mathbf{s}_n^T \rangle + P_{\mathbf{o}}L_{\mathbf{o}}\mathbf{x}_n^{\mathbf{o}}\langle \mathbf{s}_n^T \rangle, \tag{33}$$

after which we project to an orthogonal matrix using (31). Here, $P_{\mathbf{m}}$ and $P_{\mathbf{o}}$ are operators projecting onto the missing and observed dimensions respectively.

The E-step consists in the calculation of the sufficient statistics $\langle \mathbf{s}_n \rangle$ for complete and incomplete data vectors and $\langle \mathbf{s}_n \mathbf{s}_n^T \rangle$ only for incomplete data vectors. This calculation is straightforward, given the posterior densities (8) for a complete data vector and (22) for an incomplete data vector. Ignoring dependence on the sample index n we find for a complete data vector,

$$\langle s_i \rangle = \sum_{a=1}^{M} \alpha_i^a b_i^a, \tag{34}$$

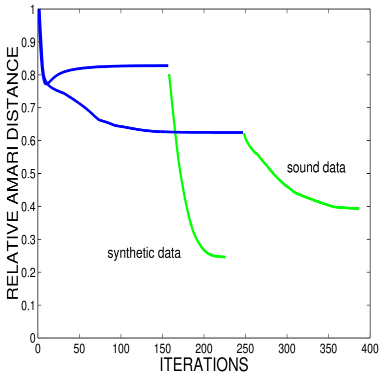
and for an incomplete data vector,

$$\langle \mathbf{s} \rangle = \sum_{J=0}^{2^D-1} \alpha_J \, \mathbf{b}_J \tag{35}$$

$$\langle \mathbf{s}\mathbf{s}^T \rangle = \sum_{J=0}^{2^D-1} \alpha_J \, (\Gamma_J + \mathbf{b}_J \mathbf{b}_J^T) \tag{36}$$

Alternating M-step and E-step will produce an ML estimate for the matrix A.

6 Experiments



	N	D	q	Amari Distance
ES	3000	5	0.6	$10.2 \rightarrow 6.4$
E1	1000	5	0.3	$2.5 \rightarrow 1.3$
E2	1000	5	0.5	$9.0 \rightarrow 3.3$
E3	700	5	0.4	$9.5 \rightarrow 4.1$
E4	500	4	0.5	$4.8 \rightarrow 2.3$
E5	500	5	0.5	$9.2 \rightarrow 2.8$
E6	400	6	0.2	$9.2 \rightarrow 5.5$
E7	300	3	0.6	$4.7 \rightarrow 2.55$
E8	300	3	0.3	$1.4 \rightarrow 1.3$
E9	200	2	0.5	$0.4 \rightarrow 0.5$
E10	100	2	0.2	$0.5 \rightarrow 0.5$

In order to explore the feasibility of our method we performed experiments on real sound data as well as artificial data. The sounds were CD recordings[1] which we subsampled by a factor of 5. The artificial data were generated using the Laplace distribution, $p(x) = \frac{1}{2}\exp(-|x|)$, for all sources. To measure the goodness of fit we used the Amari distance (Amari et al., 1996) which is invariant to permutations and scaling between the true mixing matrix and the estimated one:

$$\mathcal{N} = \sum_{i=1}^{N} (\sum_{j=1}^{N} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1) + \sum_{j=1}^{N} (\sum_{i=1}^{N} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1). \tag{37}$$

The matrix P is defined as $P = A^* A^{-1}$, where A^* is the true mixing matrix and A is the estimated mixing matrix. The Gaussian mixture that was used to model the source densities has the following parameters: $\pi_i^a = \frac{1}{2}$, $\mu_i^a = 0$ and $\sigma_i^1 = \sqrt{1.99}$, $\sigma_i^2 = \sqrt{0.01}$. To simulate missing features we deleted entries of the data matrix at random with probablity q. After the preprocessing steps (calculation of mean and covariance from incomplete data),

[1]The recordings can be found at http://sweat.cs.unm.edu/ bap/demos.html

we initialized the algorithm by the mixing matrix estimated from the subset of *complete* data points. This allows to observe the improvement over a strategy, where incomplete data is simply discarded.

In the table we list some results obtained with artificial and sound data for different values of D (number of sources), q (probability of deleting a data feature) and N (number of data points). As expected, the gain is more important in higher dimensions, since the fraction of complete data points will decrease with dimensionality, given a fixed probability that a feature is missing. In the figure we plot the Amari distance as a function of the number of iterations for experiment **E5** (synthetic data curve) and for the sound data **ES**. The sound data consisted of 3000 samples from 5 sources with positive kurtosis (between 0.5 and 3). The incomplete data are taken into consideration after the first plateau in each curve. Clearly, the estimate of mixing matrix is significantly improved by our algorithm.

To verify whether we could match these results by naive imputations we adopted two ways to fill in the missing data. First we completed the data with the mean value of all observed values of the dimension corresponding to a missing feature. As a second strategy, we filled in missing features in a data point with their expected values, given the observed dimensions in the same data point. (To compute expected values, we fit a Gaussian to the complete data.) The result is, that all filled in features lie in a hyperplane, which introduces a strong bias. In all cases these methods were vastly inferior to the EM solution.

7 Discussion

To estimate ICA components from incomplete data vectors we proposed a constrained EM algorithm. It was shown that significant improvement was gained over using only complete data or naive imputation methods. We also observed that estimation from incomplete data becomes more important in higher dimensions. Approximations to speed up the algorithm in higher dimensions will be adressed in future research.

References

Amari, S., Cichocki, A., and Yang, H. (1996). A new algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763.

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

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

Bell, A. and Sejnowski, T. (1997). The independent components of natural scenes are edge filters. *Vision Research*, 37:3327–3338.

Cardoso, J. (1999). High-order constrast for independent component analysis. *Neural Computation*, 11:157–192.

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

Ghahramani, Z. and Jordan, M. (1994). Learning from incomplete data. Technical Report A.I. Memo 1509, Massachusetts Institute of Technology, Artifical Intelligence Laboratory.

Girolami, M. and Fyfe, C. (1997). An extended exploratory projection pursuit network with linear and nonlinear anti-hebbian lateral connections applied to the cocktail party problem. *Neural Networks*, 10:1607–1618.

Hyvärinen, A. (1997). Independent component analysis by minimization of mutual information. Technical report, Helsinki University of Technology, Laboratory of Computer and Information Science.

Pearlmutter, B. and Parra, L. (1996). A context sensitive generalization of ica. *International conference on neural information processing*, pages 151–157.