Introductory notes on **Independent Component Analysis** and the **Blind Separation of Sources** problem

Timothy Corbett-Clark, 12.2.1999
SP&NN Research Group, Oxford

1 Introduction

Independent Component Analysis (ICA) is a solution to the problem of determining the matrix which will unmix the (unknown) linear combination of (unknown) independent sources. The only information available is a set of observations of the linearly combined signals. This problem is also known as Blind Separation of Sources (BSS).

The need for blind separation of sources arises in diverse applications, including

- *Speech separation.* Here the samples consist of several speech signals that have been linearly mixed together, and the requirement is to separate them back into individual speakers [3]. Such a situation occurs for example in a teleconferencing environment, and also the infamous "cocktail party".

- *Multisensor biomedical records.* Here the samples consist of recordings made by a multitude of sensors used to monitor biological signals. For example, the requirement may be to separate the heartbeat of a fetus from that of the mother using different leads of an Electrocardiograph (ECG) [6]. Another example is removing artifacts such as eye movement, periodic muscle spiking, line noise, and cardiac contamination from Electroencephalograph (EEG) recordings [10].

- *Exploratory data analysis and visualisation.* ICA is closely related (but distinct from) principal component analysis and factor analysis. It is probably very similar to projection pursuit.

There are several good on-line ICA resources: [2] contains an extremely comprehensive list of individuals and publications; [8] provides an introduction, some demonstrations, and some EEG analysis; and [12] provides a list of individuals, the FastICA MATLAB package (now in `/users/tcc/matlab/fastICA`), some EEG analysis, and details of their fixed-point algorithm.

2 Approaches

There are several approaches to ICA, including: maximising a suitable likelihood function; maximising entropy [3]; and maximising a criterion for statistical independence [7, 1]. The maximum

entropy approach has been shown several times to be equivalent to the maximum likelihood approach [5]; both are different from maximising statistical independence [9].

We will concentrate on the maximum likelihood approach.

2.1 Maximum Likelihood ICA

Let the vector $\mathbf{s}$ represents m independent sources (or latent variables), the square mixing matrix $\mathbf{A}$ represents the linear mixing of the sources, and the vector $\mathbf{x}$ represents the m components of the observed signals. The model is simply

$$\begin{aligned} \mathbf{x} &= \mathbf{A}\mathbf{s} \\ p(\mathbf{s}) &= \prod_{i=1}^{m} p(s_i) \end{aligned} \tag{1}$$

and makes the following assumptions: there are m independent sources and m observations; the mixing matrix is invertible; the components of $\mathbf{s}$ are independent and identically distributed; there is no "noise" term; there is no time dependence, so each observation is independent of previous and future observations and the mixing matrix is constant; the signals are instantaneously mixed. For simplicity we have also made the assumption that the distributions $p(s_i)$ have no parameters. In particular we will assume that they have zero mean and a fixed, usually unit, variance.

Using the property $p(\mathbf{s}) = |\det \mathbf{A}| p(\mathbf{x})$, the average log likelihood of a sample of N independent observations is then

$$\begin{aligned} L &= \frac{1}{N} \sum_{n=1}^{N} \log p(\mathbf{x}^n) \\ &= \frac{1}{N} \sum_{n=1}^{N} \log |\det \mathbf{A^{-1}}| + \log p(\mathbf{s^n}) \\ &= \log |\det \mathbf{A^{-1}}| + \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{m} \log p(s_i^n) \end{aligned} \tag{2}$$

where[1] $s_i = \mathbf{A}_{ij}^{-1} x_j$.

So to fit the model, the average log likelihood L is maximised with respect to the parameters $\mathbf{A}^{-1}$. Differentiating with respect to $\mathbf{A}^{-1}$ provides the gradient information required for optimisation,

$$\frac{\partial L}{\partial \mathbf{A}_{ij}^{-1}} = \mathbf{A}_{ij}^{T} + \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial}{\partial s_i} \log p(s_i^n) \right]_i x_j^n. \tag{3}$$

[1]The standard index convention implies that there is an implicit summation over j.

This may be written more simply without the indexes,

$$\frac{\partial L}{\partial \mathbf{A}^{-1}} = \left(\mathbf{I} + \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{s}^n)(\mathbf{s}^n)^T \right) \mathbf{A}^T, \tag{4}$$

where the i^{th} component of the activation function $\boldsymbol{\phi}$ is

$$\phi_i(\mathbf{s}^n) = \frac{\partial}{\partial s_i} \log p(s_i^n). \tag{5}$$

2.2 Choosing a source distribution

The maximum likelihood approach to ICA requires the form of the source distributions $p(s_i)$ to be specified/assumed. The Gaussian and the logistic distributions are considered below.

2.2.1 Gaussian sources

Although the most obvious choice for the source distributions, assuming Gaussian sources fails because the unmixing matrix is only recoverable upto a rotation. To see this, consider that if

$$p(s_i) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{s_i^2}{2} \right) \tag{6}$$

then equation (5) becomes

$$\phi_i(s_i) = -s_i, \tag{7}$$

and hence the gradient given by equation (4) becomes

$$\frac{\partial L}{\partial \mathbf{A}^{-1}} = \left(\mathbf{I} - \frac{1}{N} \sum_{n=1}^{N} (\mathbf{s}^n)(\mathbf{s}^n)^T \right) \mathbf{A}^T. \tag{8}$$

This gradient is the zero for any matrix $\mathbf{A}^{-1}$ which spheres/whitens/decorrelates[2] the data $\{\mathbf{x}^n\}_{n=1}^N$. Stacking all the observations by column in the matrix $\mathbf{X}$ and using eigenvector/eigenvalue decomposition,

$$\mathbf{X}\mathbf{X}^T = \mathbf{V}\mathbf{D}\mathbf{V}^T, \tag{9}$$

it is easily verified that

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{x} \qquad \text{where } \mathbf{A}^{-1} = \mathbf{R}N^{1/2}\mathbf{D}^{-1/2}\mathbf{V}^T \tag{10}$$

makes equation (8) equal to zero for any matrix $\mathbf{R}$ with the property $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. An intuitive way of understanding the presence of an arbitrary rotation is to consider that decorrelation is the transformation which makes the projected variance in all directions equal to unity; it is clear that this property is unchanged by further rotations.

To conclude, the unmixing matrix resulting from assuming Gaussian sources is arbitrary up to a rotation, and therefore clearly unhelpful. We can only hope that the sources really aren't (time) independent Gaussians !

[2]Decorrelate is probably the preferred term; such data may not look anything like a sphere.

2.2.2 Logistic sources

Observe that the sigmoid function is a valid cumulative density function because it monotonically increases from 0.0 to 1.0 as the input increases from $-\infty$ to $+\infty$. Thus modelling the sources with logistic distributions can be written

$$\begin{aligned} P(s_i < a) &= \mathrm{sig}(a) \\ p(s_i) &= \mathrm{sig}(a)(1 - \mathrm{sig}(a)) \end{aligned} \tag{11}$$

where

$$\mathrm{sig}(a) = \frac{1}{1 + e^{-a}}. \tag{12}$$

This distribution has a variance of $\frac{\pi^2}{3}$, and so must be scaled to make the variance equal to unity. However it is convenient[3] to absorb this scaling into the unmixing matrix.

When $p(s_i)$ is the logistic defined above, it is easy to show that the activation function, equation (5), has the following simple form:

$$\phi_i(\mathbf{s}) = 1 - 2\mathrm{sig}(s_i), \tag{13}$$

and hence the gradient, equation (4), becomes

$$\frac{\partial L}{\partial \mathbf{A}^{-1}} = \left(\mathbf{I} - \frac{1}{N} \sum_{n=1}^{N} (\mathbf{1} - 2\mathrm{sig}(\mathbf{s}^n))(\mathbf{s}^n)^T \right) \mathbf{A}^T. \tag{14}$$

3 MATLAB code

Minimum functionality MATLAB code which implements the maximum likelihood method of ICA with logistic sources is shown in figure 1. Note that the function `fminu`, which performs BFGS quasi-Newton optimisation, is in the optimisation toolbox[4]. This takes the names of two functions, one which evaluates the minimand, and one which evaluates its gradient. These functions are given in figures 2 and 3 respectively.

4 Two illustrative examples

A simple test of the above method is to generate source data **S** from the assumed source distributions (in this case, independent logistics), make a set of observations **X** by linearly combining

[3] Convenient because it is simple. Actually it is probably not what we want because the likelihood then changes with the variance of the sources.

[4] Type `help optim` for information on MATLAB's optimisation toolbox. Alternatively with MATLAB5 use the command `helpwin`. On-line documentation is also available at `/data/apps/matlab-5/help/helpdesk.html`. Much of it is in pdf format so you may wish to start netscape with the acroread plug-in by using (for example) `/users/tcc/bin/netscape`.

```
function Ahatinv = ica(X)
% Ahatinv = ica(X)
%    Independent Component Analysis on data in the columns of X.
%    Use maximum likelihood method. Assume logistic source distributions.
%
%    Return the unmixing matrix Ahatinv

% useful global variables (can't be helped)
global ICA_X ICA_DIMENSION ICA_NPATTERNS
ICA_X = X;
[ICA_DIMENSION, ICA_NPATTERNS] = size(ICA_X);

% initialise Ahatinv to decorrelating transformation
% (helps to prevent ill-conditioning)
[V,D]=eig(X*X'); Ahatinv0 = diag(1./sqrt(diag(D)))*V';

% set options for optimisation algorithm
options(1) = 1; % display info
options(6) = 0; % BFGS quasi-Newton (default)
options(7) = 1; % with cubic line search

% minimise the average negative log likelihood
Ahatinv = fminu('evalf', Ahatinv0, options, 'gradf');

% tidy
clear ICA_X ICA_DIMENSION ICA_NPATTERNS
```

Figure 1: MATLAB code implementing ICA using the maximum likelihood method, assuming logistic distributed sources. In file "ica.m".

```
function avnegloglikelihood = evalf(Ahatinv)
global ICA_X ICA_NPATTERNS

S = Ahatinv*ICA_X;
temp = sigmoid(S);
avnegloglikelihood = -(log(abs(det(Ahatinv))) +
                       (sum(sum(log(temp.*(1.0-temp))))) ./ ICA_NPATTERNS);
```

Figure 2: MATLAB function to evaluate the average negative log likelihood. In file "evalf.m".

```
function grad = grad_f(Ahatinv)
global ICA_X ICA_NPATTERNS ICA_DIMENSION

S = Ahatinv*ICA_X;
grad = -(eye(ICA_DIMENSION)+(1-2*sigmoid(S))*S' ./ ICA_NPATTERNS) * inv(Ahatinv');
```

Figure 3: MATLAB function to evaluate the gradient. In file "gradf.m".

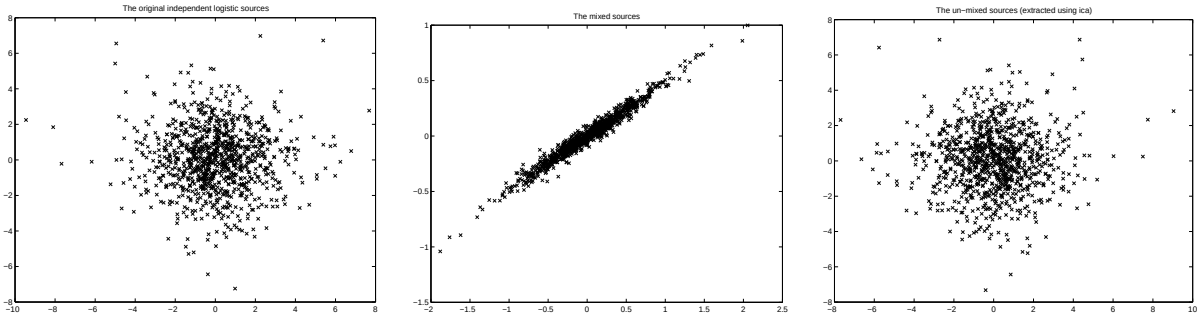

Figure 4: Example showing data generated from two independent logistic distributions (shown left), mixed together using the matrix given in the text (shown middle), and unmixed using the ICA maximum likelihood algorithm (shown right).

the source data with a known mixing matrix $\mathbf{A}$, and then use the above MATLAB code to try to recover the unmixing matrix $\widehat{\mathbf{A}^{-1}}$ *only* from the set of observations $\mathbf{X}$.

The left-most plot of figure 4 shows 1000 points sampled from a two dimensional distribution formed from two independent logistic distributions (one per axis). The middle plot shows the result of mixing these sources using the following matrix

$$\mathbf{A} = \begin{pmatrix} 0.2262 & 0.1143 \\ 0.1180 & 0.0332 \end{pmatrix}. \tag{15}$$

The right-most plot shows the result from using the MATLAB code given above to unmix the data. The product of the original mixing matrix $\mathbf{A}$ and the estimated un-mixing matrix $\widehat{\mathbf{A}^{-1}}$ should be the identity up to permutations and axis-aligned reflections of the variables. In fact,

$$\widehat{\mathbf{A}^{-1}}\mathbf{A} = \begin{pmatrix} -0.9712 & -0.0460 \\ -0.0253 & 1.0041 \end{pmatrix}. \tag{16}$$

4.1 ICA is different from PCA

Given the above plots one might be tempted to imagine that ICA is the same as Principal Component Analysis (PCA). This is not the case, and may be understood by realising that unlike ICA, PCA is arbitrary if the data is decorrelated. Recall that PCA finds directions of maximum projected variance, so if the data has equal variance in all directions (*ie* because it is decorrelated), then no one direction is any more a maximum projected variance than any other direction.

Assuming an identity rotation matrix, multiplying the principal component projection matrix $\widehat{\mathbf{A}^{-1}_{\text{PCA}}}$ by $\mathbf{A}$ produces

$$\widehat{\mathbf{A}^{-1}_{\text{PCA}}}\mathbf{A} = \begin{pmatrix} -0.0155 & -0.0071 \\ -0.0075 & 0.0154 \end{pmatrix}, \tag{17}$$

which is much less diagonally dominated than equation (16). Another example which demonstrates the difference between ICA and PCA will be given later.

On a point of implementation, initialising ICA with PCA appears to help prevent ill-conditioning.

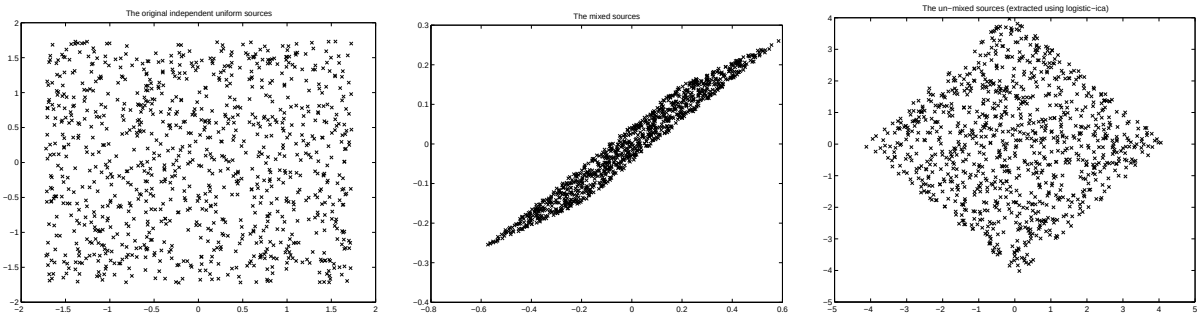

Figure 5: Example showing data generated from two independent *uniform* distributions (shown left), mixed together using the matrix given in the text (shown middle), and unmixed using the ICA maximum likelihood algorithm *but assuming logistic sources* (shown right).

4.2 Robustness to incorrectly assumed source distributions

Recall that above we assumed the sources to have logistic distributions, and then tested the algorithm on data generated from logistic distributions. An obvious question is how robust this algorithm is to sources which do not have logistic distributions. Figure 5 shows what happens if the source distributions are uniform but we still assume logistic distributions in the maximum likelihood model.

This is obviously wrong, as can be additionally verified by looking at the product of the original mixing matrix $\mathbf{A}$ and the estimated unmixing matrix $\widehat{\mathbf{A}^{-1}}$,

$$\widehat{\mathbf{A}^{-1}}\mathbf{A} = \begin{pmatrix} -1.1705 & -1.2325 \\ -1.2084 & 1.1738 \end{pmatrix}. \tag{18}$$

This is not a numerical problem. The algorithm reaches a true (local) minimum, and the 45^o effect is completely reproduceable. In fact there are simple theoretical reasons for why this algorithm has produced the worst conceivable solution on this data. However before explaining these reasons, it is useful to understand what is meant by kurtosis and super/sub-Gaussianity.

5 Kurtosis, and super/sub-Gaussian distributions

A random variable X is said to be *sub-Gaussian* [4] if it is uniformly distributed, or its probability density function is expressible in the form $\exp -g(x)$ where $g(x)$ is an even function that is differentiable (except possibly at the origin), and both $g(x)$ and $g'(x)/x$ are strictly increasing for $0 < x < \infty$. If however $g'(x)/x$ is strictly decreasing for $0 < x < \infty$ then the random variable X is said to be *super-Gaussian*. A simple example is $g(x) = |x|^\beta$, for which X is sub-Gaussian if $\beta > 2$, super-Gaussian if $\beta < 2$, and of course Gaussian if $\beta = 2$. Thus loosely speaking, super-Gaussian distributions have heavy tails whereas sub-Gaussian distributions have light tails.

A convenient measure of the weight in the tails is the relative or *excess kurtosis*, which for a

	uniform	triangular	Gaussian	logistic
κ_4	-1.2	-0.6	0	1.2

Table 1: The excess in kurtosis for some common distributions.

	uniform	triangular	Gaussian	logistic
av log likelihood	-2.060	-2.038	-2.024	-2.000

Table 2: The average log likelihood of various distributions assuming a logistic model, estimated using a sample size of 1000000.

zero mean random variable X is defined

$$\kappa_4(X) = \frac{E[X^4]}{(E[X^2])^2} - 3. \tag{19}$$

The "3" normalises the definition so that the kurtosis of a Gaussian is zero. The kurtosis of some common distributions are shown in table 1. The kurtosis of a super-Gaussian is positive and the kurtosis of a sub-Gaussian is negative.

5.1 Using kurtosis to explain ICA

Insight into ICA can be gained from observing that a linear combination of random variables is more Gaussian than the original random variables. This is a crude interpretation of the central limit theorem [14]. Thus assuming super-Gaussian distributions with data from sub-Gaussian sources results in the output from the "unmixing" matrix being more Gaussian and therefore *less* separated. In terms of kurtosis, in the example of figure 5, two uniformly distributed variables are linearly combined, thus forming a triangular distribution. From table 1 it can be seen that the kurtosis of a uniform distribution is -1.2 and the kurtosis of a triangular distribution if -0.6. The latter is closer to the kurtosis of a logistic distribution 1.2, and thus one might expect the likelihood of data from a triangular distribution under a logistic model to be greater than the likelihood of data from a uniform distribution. Table 2 confirms this.

It is also enlightening to view plots of super-Gaussian and sub-Gaussian distributions, and then to interpret the maximum likelihood approach as a means of linearly transforming the data to maximise the likelihood that it came from one of these distributions. Figure 6 shows contour-plots of sub-Gaussian, Gaussian, and super-Gaussian distributions. It is now clear why a model with (super-Gaussian) logistic sources fails to unmix a linear mixture of data from two uniform distributions (which are sub-Gaussian). The best "fit" of a mixture of uniform distributions onto a super-Gaussian distribution is at 45^o degrees to the axes.

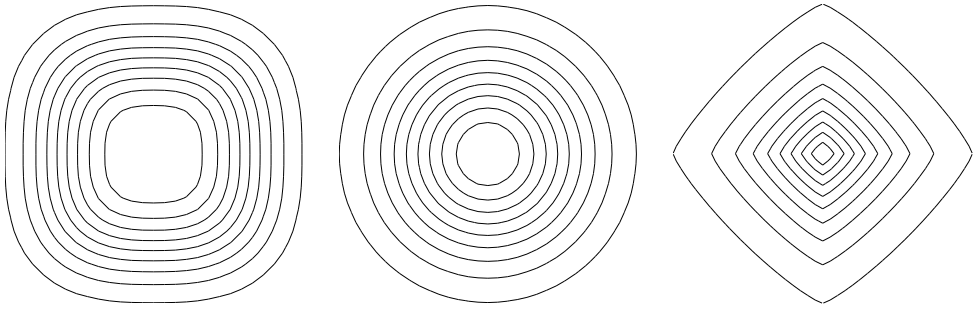

Figure 6: Contour plots of sub-Gaussian, $\beta = 2.8$ (left), Gaussian $\beta = 2$ (middle), super-Gaussian $\beta = 1.2$ (right) distributions.

5.2 ICA for sub-Gaussian distributions

Additional verification of the above explanation is obtained by using a model with *sub*-Gaussian sources to try to unmix data originating from either super or sub-Gaussian distributions. One expects the data to be correctly separated in the former case but not the latter. Using a model with sub-Gaussian sources, $p(s_i) = \alpha \exp -s_i^4$, and repeating the experiments of figures 4 and 5, one indeed obtains the expected solutions.

Figure 7 shows another example where 1000 samples are generated from two independent sources which have quite different distributions. The first signal is a modulated sinusoid and the second is random uniform noise. The figures show these signals being successfully unmixed using the sub-Gaussian ICA described above. This should be compared with figure 8 which shows that a model assuming super-Gaussian (logistic) source distributions completely fails to recover the original signals. In fact the kurtosis of the two signals is -0.74 and -1.2, so this result is not surprising. It is also worth observing that PCA does not separate these signals particularly well either, see figure 9.

In summary, using the likelihood model with assumed logistic sources will not separate sub-Gaussian data. Similarly, assuming sub-Gaussian sources only enables sub-Gaussian data to be separated, but does not separate super-Gaussian data. Obviously the more accurately the assumed source distributions are able to fit the actual sources, the more likely the model will be able to separate the mixture.

6 Practical ICA

Independent Component Analysis can be succinctly described as a linear non-Gaussian latent variables model. It is been shown that the assumed distributions of the latent variables *is* important to the success of the algorithm. In particular, correctly choosing a sub-Gaussian or super-Gaussian distribution is vital, and an intuitive explanation for this was given in terms of a crude version of the central limit theorem – a linear combination of either sub or super-Gaussian dis-

tributions tends to be more Gaussian. Clearly it would also be useful to avoid having to assume either all super-Gaussian or all sub-Gaussian sources. One possible method is suggested in [11]. Another possibility is to assume a family of distributions of the form

$$p(s_i) = \alpha(\beta_i) \exp -|s_i|^{\beta_i}, \tag{20}$$

and maximising the likelihood not only with respect to the unmixing matrix but also with respect to the parameters β_i. This might automatically select an appropriate sub or super-Gaussian distribution for each source.

The model described in this introduction is extremely simple, and constrained by the assumptions given in section 2.1. Probably the most unrealistic assumption is that the samples are independently sampled. In other words, the data is not a time series. Context-ICA [13] is one possible method of implementing ICA "through time". Adding a "noise" term would also increase the realism of the model, and possibly enable the extraction of fewer independent components than there are number of components in each observation.

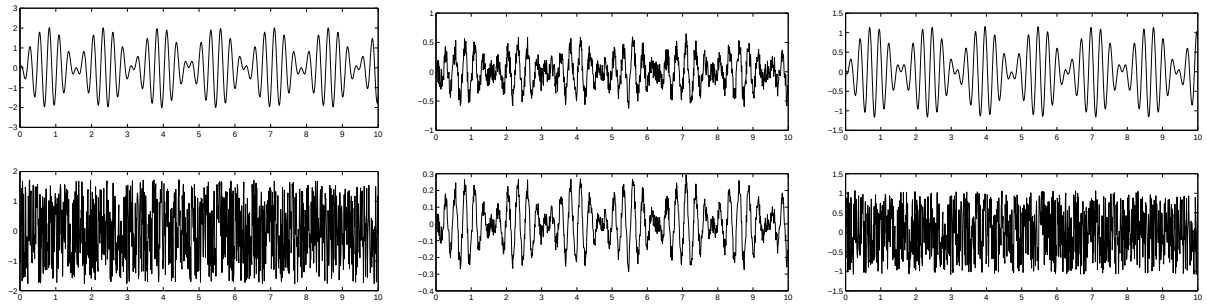

Figure 7: Sub-Gaussian ICA successfully separating two signals.

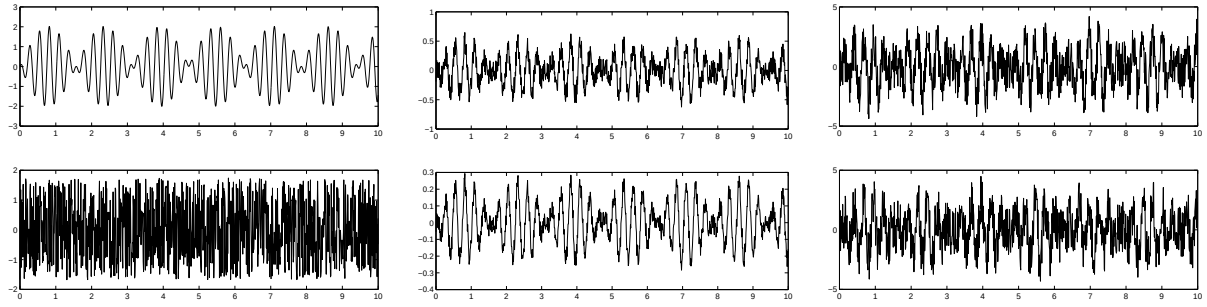

Figure 8: Super-Gaussian ICA failing to separate two signals.

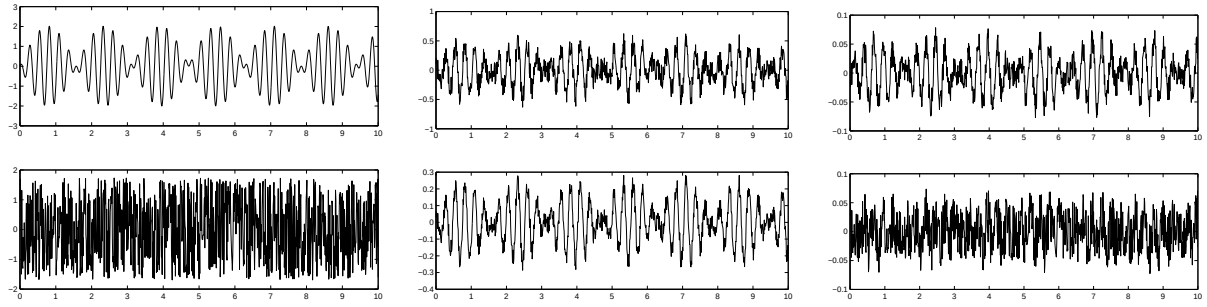

Figure 9: PCA failing to properly separate two signals.

References

[1] S Amari, A Cichoki, and H H Yang. A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8:757–763, 1996. Cambridge, MA: MIT Press.

[2] Allan Kardec Barros. Independent component analysis. `http://www.bmc.riken.go.jp/sensor/Allan/ICA/`.

[3] A J Bell and T J Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 6:1129–1159, 1995.

[4] A Benveniste, M Metivier, and P Priouret. *Adaptive Algorithms and Stochastic Approximations*. New York:Springer-Verlag, 1987.

[5] J-F Cardoso. Infomax and maximum likelihood for blind source separation. *IEEE Signal Porcessing Letters*, 4:112–114, 1997.

[6] J-F Cardoso. Multidimensional independent component analysis. *Proceedings IEEE ICASSP, Seatle, WA*, May 1999.

[7] P Comon. Independent component analysis. *Proceedings of International Signal Processing Workshop on Higher-order Statistics*, pages 111–120, 1991. Chamrousse, France.

[8] La Jolla Computational Neuroscience Laborabory at Salk Institute. Independent component analysis - cnl. `http://www.cnl.salk.edu/~tewon/ica_cnl.html`.

[9] Simon Haykin. *Neural Networks, A Comprehensive Foundation. Second edition*. Macmillan, 1999.

[10] T P Jung, C Humphries, T W Lee, S Makeig, M McKeown, V Iragui, and T Sejnowski. Extended ica removes artifacts from electroencephalographic recordings. *Advances in Neural Information Processing Systems*, May 1997.

[11] Te-Won Lee and Terrence J Sejnowski. Independent component analysis for mixed subgaussian and super-gaussian sources. Technical report, Computational Neurobiology Lab, The Salk Institute, La Jolla, California, USA, 1998.

[12] Helsinki University of Technology. Independent component analysis research group at the helsinki university of technology. `http://www.cis.hut.fi/projects/ica/`.

[13] Barak A Pearlmutter and Lucas C Parra. A context-sensitive generalisation of ica. Technical report, Dept. of Cog. Sci., UCSD, La Jolla, California, USA. Siemens Corporate Research, Princeton, New Jersey, USA, 1998.

[14] John A Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 1995.