Single Channel and Space-Time ICA

By Mike Davies

IDCOM and JRISIP, School of Engineering and Electronics, The University of Edinburgh, UK

Abstract

We consider ICA codebooks and identify when these can be used to extract independent components from a stationary scalar time series. This is motivated by recent empirical work that suggests that single channel ICA can sometimes be used in this way. Here we show that if the sources are spectrally disjoint we can identify and separate them. We will finally consider a space-time ICA codebook which can be used to perform convolutive blind source separation and in some circumstances extract more sources than sensors.

1. Introduction

While Independent Component Analysis (ICA) was originally proposed for the blind separation of vector-valued observations recent empirical work suggests that it can sometimes be used to separate out independent sources from a single time series. Here we will call this type of analysis "Single channel ICA" (SCICA). SCICA can be regarded as an extreme case of underdetermined ICA (one sensor!) and it is interesting to explore when and how applying a standard ICA solution can solve this problem.

Here we set out a mathematical framework that explains, under certain assumptions, how and when standard ICA can perform source separation on a single sensor and we show that SCICA can only separate out sources whose power spectra have disjoint support. We conclude with two possible algorithmic solutions and some numerical experiments. The results follow our predicted theory.

2. Single Channel ICA

A simple but popular model of a signal, $x(t) \in \mathbb{R}^N$ is to represent it as a linear superposition of functions $a_i(t)$: $x(t) = \sum_i s_i a_i(t)$, where s_i are the function weights or coefficients, or in vector-matrix form: $\mathbf{x} = A\mathbf{s}$, where $\mathbf{x} = [x(t), x(t-1), \ldots, x(t-N+1)]^T$ and $A = [a_1(t), \cdots, a_N(t)]$. Often A is chosen to be invertible so that the inverse (analysis) equation is uniquely defined: $\mathbf{s} = \mathbf{W}\mathbf{x}$. where $W = A^{-1}$. This popular framework includes the DFT, wavelets, etc. If s_i are treated as *independent* then we also have the classic ICA model.

In practice one might break up a signal into a series of contiguous analysis blocks and treat these as a sequence of vector observations. A standard ICA algorithm can then be applied to the data matrix to learn a 'well matched' representation. This is also closely linked with learning sparse factorial codes.

2.1. *Separation & Reconstruction*

In standard ICA, each source can be separated and reconstructed in the observation domain through the operation:

$$\mathbf{x}_{\mathbf{s_i}} = A_{(:,i)} W_{(i,:)} \mathbf{x}$$

where $\mathbf{x_{s_i}}$ is the i-th source in the observation domain. With a single channel of data we can apply the same formula to non-overlapping data blocks giving:

$$x_{s_i}(nN - k + 1) = A_{(:,i)} \sum_{j=1}^{N} W_{(i,j)} x(nN - j + 1)$$

However the resulting source estimates are highly dependent on the block alignment. For shift-invariance we need to use 'cycle-spinning': essentially a different estimate is made for each possible block alignment. The 'cycle-spinning' estimate is then the average of all of these fixed block estimates.

In our case the cycle-spinning separation is given by:

$$\begin{aligned} x_{s_i}(t) &= \frac{1}{N} \sum_{k=1}^{N} A_{(k,i)} \sum_{j=1}^{N} W_{(i,j)} x(t - j + k) \\ &= a_i(-t) * w_i(t) * x(t) \end{aligned} \tag{2.1}$$

where $a_i(t)$ is the filter associated with the column vector $A_{(:,i)}$ and $w_i(t)$ is similarly defined.

Furthermore if we use a pre-whitening filter then the separation and reconstruction involves application of: the whitening filter; the separation filter in equation (2.1); and the inverse whitening filter. However, since filters commute this is equivalent to just the application of the separation filter. Whitened data also implies that the separating matrix W is orthogonal and therefore $a_i(t) = w_i(t)$, so the **separating filters are symmetric and has zero phase**.

2.2. *Independent or merely interesting?*

What is not clear is when we can really expect the transform coefficients $s_i(t)$ to be independent. If we merely wish to learn 'interesting' features we can relax the notion of independence (e.g. Topographical ICA). In contrast, when identifying real sources independence is important. We will see next that the appropriate model is a special verion of Cardoso's Multi-dimensional Independent Component Analysis (MICA). Recall MICA assumes that the observed data $\mathbf{x}$ can be decomposed into $\mathbf{x} = \sum_p \mathbf{x_p}$ where each $\mathbf{x_p} \in \mathbb{R}^N$ lie in an n_p dimensional subspace E_p and the set $\{E_1, \ldots, E_c\}$ are linearly independent.

3. Independent Mixtures of Random Processes

We now determine when a single channel signal can be modelled as a MICA system. Suppose that $x(t)$, admits a decomposition into the sum of mutually independent random processes, $x_{s_i}(t)$:

$$x(t) = \sum_i x_{s_i}(t) \tag{3.1}$$

Furthermore if $x_{s_i}(t)$ is a filtered i.i.d. process $x_{s_i} = h_i * s_i$ then we can form an ICA-type model:

$$\mathbf{x} = \sum_i H_i \mathbf{s}_i \tag{3.2}$$

where H_i is the Toeplitz matrix associated with the filter h_i. Note for a single source, solving for $s_1(t)$ is the blind deconvolution problem.

At this point it appears we will need to consider overcomplete representations. However when h_i is not invertible the matrix H_i may be rank deficient. While this immediately precludes the possibility of extracting $s_i(t)$, it does not necessarily preclude the estimation of $x_{s_i}(t)$. If each matrix H_i is rank deficient: i.e. its columns only span a subspace E_i then $\mathbf{x_{s_i}} = H_i s_i \in E_i$ and, if all the subspaces, $\{E_i\}$, are linearly independent, we have a valid MICA model.

More generally $x_{s_i}(t)$ can lie on a low dimensional subspace E_p characterized by the significant eigenvectors of the correlation matrix C_{x_i}, but without the restriction $x_{s_i} = h_i * s_i$. For the subspaces, $\{E_i\}$, to be linearly independent the components must have disjoint spectral support.

4. Solving SCICA with ICA

Cardoso argued that a standard ICA algorithm can be used to learn MICA by grouping the components based on dependency. Here we note that the learnt basis functions are approximate shifts of the generating filters h_i. The number of shifted versions of h_i will depend on the bandwidth. Furthermore since the components are assumed to have disjoint spectral support. We can group them based on power spectra. We therefore propose the following algorithm:

- Temporally whiten $x(t)$ with possible dimension reduction
- Apply ICA to learn mixing matrix A.
- Calculate Transfer Functions (TFs) of the basis vectors $a_i(t)$ and cluster into groups γ_p, $p = 1, \ldots, c$ using k-means.
- Calculate the separation and reconstruction filters, $\tilde{f}_p(t)$, defined as:

$$\tilde{f}_p(t) = \frac{1}{N} \sum_{i \in \gamma_p} w_i(-t) * w_i(t) \tag{4.1}$$

5. A Quick and Dirty Solution

A faster method that avoids explicit clustering can be developed based on a deflationary approach. It is particularly attractive when there are only a small number of independent processes to be extracted (for example, when using the constrained ICA).

We first note that the separating filter, above, can be approximated using any single independent component associated with the given source:

$$\tilde{f}_p(t) \approx \alpha_p f_i(t) = \frac{\alpha_p}{N} w_i(-t) * w_i(t), \text{ for any } i \in \gamma_p$$

were α_p can be estimated by minimising the norm of the residual $r(t) = x(t) - \alpha_p f_i(t) * x(t)$.

This suggests the following algorithm:

- Temporally whiten signal (with possible dimension reduction);
- Extract one component (deflationary ICA) and calculate: $\tilde{f}_p(t) = \frac{\alpha_p}{N} w_p(-t) * w_p(t)$;
- Calculate the residual $r_p(t) = r_{p-1}(t) - \tilde{f}_p(t) * r_{p-1}(t)$ $(r_0(t) = x(t))$;
- repeat from step 2.

This method tends to give inferior results to the full ICA solution. However it can be much faster.

5.1. *Toy Example*

We give a toy example: the mixture of two filtered sparse i.i.d. signals with approximately disjoint support plus a pure sinusoid: see figure 1. Applying FastICA to the mixture of the

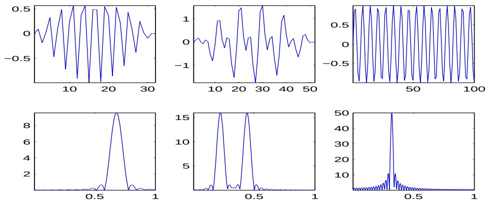

FIGURE 1. *The impulse responses and transfer functions of the two mixing filters (left and centre) and a section from the single sinusoid (right)*

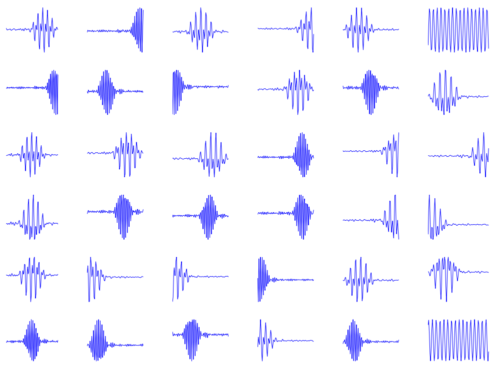

FIGURE 2. *Basis functions from the columns of the mixing matrix, A, learnt using FastICA.*

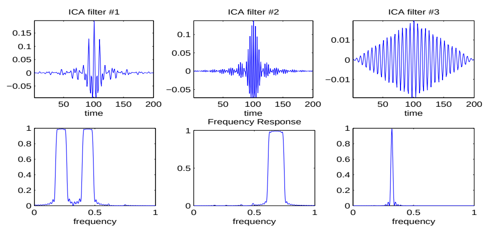

FIGURE 3. *The zero-phase reconstruction filters for the three different sources*

sources (delay dimension = 100, reduced to 36 using PCA) resulted in the basis functions (columns of A) shown in figure 2. The filter transfer functions were then grouped into 3 clusters using k-means. The separating filters are plotted in figure 3. Note that they act as frequency adaptive bandpass filters. The signals were well separated by these filters.

6. Extensions to Space-Time ICA

Following along the same lines it is possible to sort multi-channel data into space-time vectors in a similar way to Space-Time Adaptive Processing (STAP) in radar. So far our experiments in this direction have shown that it can be used to solve convolutive blind source separation and under certain circumstances extract more sources than sensors. Examples of these will be given in the presentation.