Blind Signal Separation on Real Data: Tracking and Implementation

Paul Baxter, Geoff Spence, and John McWhirter

QinetiQ, St Andrews Road, Malvern, Worcs, WR14 3PS, United Kingdom
p.baxter@signal.qinetiq.com

Abstract. There are numerous algorithms available for blind signal separation (BSS) of multiple signals, but most of these are optimised for short blocks of data, stationary signals and time invariant mixing matrices. As such, they are unsuitable for real-world applications, which often require tracking BSS carried out in real time with as small a lag as possible. This paper looks at the problems encountered in applying BSS to real data sets and addresses the issue of computationally efficient tracking BSS based on well-understood two-stage block-based approaches. An example is included where the technique is applied to a five-minute section of twin foetal electrocardiogram (ECG) data.

1 Introduction

A commonly desired objective of signal processing is to recover a signal of interest from sensor recordings in which it may be masked by noise and by other, interfering, signals. Often there will be a large number of signals present, and any individual sensor can only receive a mixture of these signals: in general, only limited information about the signals of interest can be recovered from such a mixture. Blind signal separation (BSS) aims to separate signals by utilizing multiple sensors, commonly using the assumption that the signals are independent (Independent Component Analysis). BSS has been successfully used in many different application areas, e.g. artefact suppression in electroencephlogram (EEG) recordings [8], foetal electrocardiogram (ECG) analysis [10] and image enhancement [5].

The term 'blind' is used to indicate that no prior information, either concerning the individual signals (other than the independence assumption), or the manner in which they combine at the sensors is available. Unknown factors generally include the number of signals, the locations of the signal sources and the sensor locations.

Many different techniques have been developed for carrying out BSS. Some of the best performing, or best known, are JADE [3], FastICA [6], BLISS [7], EASI [4], InfoMax [2] and kernelICA [1]. All of these have been developed to solve the BSS problem in the theoretical case, so without modification they are not necessarily suitable for processing real data sets, especially over long periods of time. Although the difficulties arising in processing real data sets are often unique to the type of data, there are many common problems. In this paper we describe some of these common problems.

Most real data sets contain non-stationary signals and time-varying mixing, especially if they are recorded over long periods of time. We describe how to extend

J. Rosca et al. (Eds.): ICA 2006, LNCS 3889, pp. 327 – 334, 2006.

single block-based BSS algorithms to produce an efficient tracking BSS approach. We provide a demonstration of this technique, applied to a twin foetal ECG data set over five minutes. This shows the utility of the technique.

In section 2 we introduce the basic BSS model, investigate the two-stage approach to solving it and observe some of the difficulties encountered when using it on real data. Our tracking BSS approach is developed in section 3, and is demonstrated on foetal twin ECG data in section 4. Conclusions are drawn in section 5.

2 Basic BSS

2.1 Data Model

The basic linear BSS data model, assuming m sensors and T samples is:

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \tag{1}$$

The (m x T) matrix **X** denotes the observed sensor data, so the rows of **X** contain the sensor outputs. The (m x m) matrix **A** denotes the time-invariant mixing matrix and the rows of the (m x T) matrix **S** contain the independent signals, assumed to be stationary. The (m x T) matrix **N** contains the sensor noise, usually assumed to be white Gaussian noise, uncorrelated between the sensors. For the sake of simplicity we assume that the number of signals is equal to the number of sensors (m), although for most techniques it is only necessary for there to be at least as many sensors as signals. The following conditions also apply to the data model:

- The mixing process is assumed to be linear and instantaneous (time delays between sensors can be represented as phase shifts);
- The mixing process is assumed to be time invariant;
- At most one of the signals has a Gaussian distribution (only required for complete signal separation).

2.2 Algorithms

Many of the basic BSS algorithms operate on the whole data block at once, using a two-stage approach to achieve signal separation. Firstly, in the *second-order stage*, the sensor outputs are decorrelated and normalised using a method such as the singular value decomposition (SVD) as shown in equation (2). Here, the columns of the (m x m) matrix **U** contain the orthonormal steering (spatial) vectors. The estimated orthonormal signals are contained in the rows of the (m x T) matrix $\mathbf{V}^T$. The (m x m) diagonal matrix α contains the singular values. The orthonormal signals are related to the independent signals by a (m x m) 'hidden' rotation matrix **R**. The *higher-order stage* of separation determines **R**, as shown in equation 3. The matrix $\mathbf{RV}^T = \hat{\mathbf{S}}$ contains the estimates of the signals and $\mathbf{U}\alpha\mathbf{R}^T$ denotes the estimated mixing matrix.

$$\mathbf{X}=\mathbf{U}\alpha\mathbf{V}^T \tag{2}$$

$$\mathbf{X}=\mathbf{U}\alpha\mathbf{R}^T\mathbf{R}\mathbf{V}^T \tag{3}$$

Many of the block-based BSS algorithms differ only in the method used for computing **R**. JADE, BLISS, FastICA and KernelICA all use this two-stage approach, with the same second-order stage but different methods for the higher-order stage.

Whichever BSS algorithm is used, certain difficulties tend to arise when they are applied to real data sets. Some of the commonest of these are:

- **Computational Cost:** In general the computational cost associated with applying BSS to a data block is high, at least $O(m^2)$ and much higher in the case of some algorithms such as KernelICA. This leads to two, more specific, problems:
  - **Real Time Processing:** In many cases real time processing is required, so computationally expensive procedures require high processing power, which is expensive and possibly unobtainable;
  - **Small Processing Lag:** Even if processing power is available to process data in real time, computationally expensive procedures lead to a large lag between the data arriving at the sensors and the processed data being available;
- **Time variation:** Although short sections of many real data sets are sufficiently time-invariant for the basic BSS algorithms to run, longer sections of such data sets are time-varying, e.g. foetal ECG recordings are often time-invariant over 10 second blocks, but the mixing is time-varying and the signals are non-stationary over 1 hour blocks. Three types of time variation are commonly seen:
  - **Non-Stationary Signals:** Usually the signal power varies, this includes the onset of interfering (jamming) signals and signal births and deaths;
  - **Non-Stationary Noise Levels:** Can lead to portions of the data where the signals are swamped by noise and signal separation may not be possible. Such events should not be allowed to bias the overall tracking process;
  - **Time-Varying Steering Vectors:** The relative locations of the signals and sensors can change during the data collection.

In this paper we develop a tracking BSS algorithm for data with significant time variation, and wherever possible try to reduce the computational costs involved.

3 Tracking BSS

We present a method for extending two-stage BSS techniques to time-varying data sets. The basic principle is to use a moving data window, where the signals are separated in each window using a block-based approach. However, the use of a moving window technique alone is not sufficient for many real applications; here we address the following issues:

- **Computational Load:** Processing the individual windows in isolation is inefficient;
- **Signal Swapping:** In each window the signals may be separated in a different order. This is due to both the inherent signal ordering indeterminacy in the higher-order stage, and to the second-order stage ordering the signals by their powers.

Our tracking BSS technique uses two techniques based on second and higher-order statistics, but for the purposes of tracking, these are first **initialised** using past information and then **updated using small rotations**. This is similar in concept to the EASI algorithm [4], but unlike EASI this approach seeks to utilize the fast convergence of block-based approaches, albeit in a tracking context.

The tracking BSS technique we present here can be implemented using either overlapping or contiguous data windows. The first window defines the acquisition phase; in this window the data is processed by a normal two-stage BSS technique. It is inefficient to process the remaining windows in isolation. The use of the SVD, equation (2), in the remaining windows can be problematic. It successfully decorrelates the signals, but also orders the signals according to their power. This can lead to signals being ordered differently in adjacent windows if their powers vary (signal swapping). Similarly, even if initialised near one solution, most higher-order techniques are not guaranteed to converge to this solution, but instead will find a permuted version of this solution.

The tracking BSS technique presented here can overcome these problems by using information estimated from the previous window to initialise the current one and then applying small updates for both the second and higher-order stages.

3.1 Second-Order Stage

Consider the second-order stage in a given tracking window. The signals can be made orthonormal by combining an initialisation process (using the second-order information from the previous window), a decorrelation method (via a Jacobi diagonalisation) for updating the orthogonality of the signals and a new normalization step.

The Jacobi method for diagonalisation can be used as the decorrelation method, where each pairwise rotation is constrained to choose the smallest of the possible angles to rotate the signals by [9]. For example, if **x** and **y** represent the (1 x T) i'th and j'th vectors the pairwise orthogonalisation step of Jacobi diagonalisation can be done by diagonalising their symmetric correlation matrix, i.e. by finding a rotation matrix **Q,** parameterised by θ, that zeros the off-diagonal elements of the matrix **B**:

$$\mathbf{B} = \mathbf{Q}^T \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \begin{bmatrix} \mathbf{x}^T & \mathbf{y}^T \end{bmatrix} \mathbf{Q} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \tag{4}$$

where c=cos(θ) and s=sin(θ). The correct θ therefore satisfies:

$$\tan(2\theta) = 2a_{ij} / (a_{jj} - a_{ii}) \tag{5}$$

The LHS of equation (5) can be expressed as $2t/(1-t^2)$, where t=tan(θ). Thus, there are two solutions for θ in the range [–π/2, π/2]. The orthogonality of two vectors has been initialised using information from the last window, so **x** and **y** are nearly orthogonal to begin with. Thus the two solutions for θ are close to 0 and π/2. A normal SVD chooses between these by ordering the outputs according to their power; this can cause the vectors to be rotated by approximately π/2 and hence introduce signal swapping. In our tracking BSS technique we avoid this by insisting on the solution closest to 0 being used.

A mathematical summary of the second-order update stage is shown in equations (7) to (10), where subscript k denotes values belonging to the k'th data window, e.g. $\mathbf{X}_k$ denotes the data in the k'th window. We first note that the symmetric (unnormalised) correlation matrix of the data has as its eigenvalue decomposition

$$\mathbf{X}_k\mathbf{X}_k^T = \mathbf{U}_k\boldsymbol{\alpha}_k^2\mathbf{U}_k^T = \mathbf{U}_k\boldsymbol{\lambda}_k\mathbf{U}_k^T, \tag{6}$$

where $\mathbf{U}_k$ contains the eigenvectors and $\boldsymbol{\lambda}_k$ contains the eigenvalues.

In equation (7) the eigenvectors for the last window $\mathbf{U}_{k-1}$ are used to initialise the eigenvalue decomposition of the current covariance matrix; for a slowly changing mixing matrix, $\boldsymbol{\lambda}_k'$ will be nearly diagonal. Thus $\boldsymbol{\lambda}_k'$ can be simply diagonalised by $\mathbf{U}_z$, equation (8), found by the Jacobi method with the small rotation constraint described above. The updated eigenvectors are found via equation (9) and the estimated orthonormal vectors are given by equation (10).

Equation (9) shows how $\mathbf{U}_k$ is calculated as the product of an **initialisation process**, $\mathbf{U}_{k-1}$, and a **small update**, $\mathbf{U}_z$.

$$\boldsymbol{\lambda}_k' = \mathbf{U}_{k-1}^T\mathbf{X}_k\mathbf{X}_k^T\mathbf{U}_{k-1} \tag{7}$$

$$\boldsymbol{\lambda}_k = \mathbf{U}_z^T\boldsymbol{\lambda}_k'\mathbf{U}_z = \mathbf{U}_z^T\mathbf{U}_{k-1}^T\left(\mathbf{X}_k\mathbf{X}_k^T\right)\mathbf{U}_{k-1}\mathbf{U}_z \tag{8}$$

$$\mathbf{U}_k = \mathbf{U}_{k-1}\mathbf{U}_z \tag{9}$$

$$\mathbf{V}_k^T = \boldsymbol{\lambda}_k^{-0.5}\mathbf{U}_k^T\mathbf{X}_k \tag{10}$$

3.2 Higher-Order Stage

The higher-order stage is carried out in a similar way to the second-order stage. The independence of the orthonormal vectors $\mathbf{V}_k^T$ can be initialised by the rotation matrix $\mathbf{R}_{k-1}$ derived from the last window, equation (11). For a slowly changing mixing matrix, and where signal swapping has been avoided in the second-order stage, then $\mathbf{R}_{k-1}\mathbf{R}_k \approx \mathbf{I}$. Then the initialized signal estimates, $\hat{\mathbf{S}}_k'$, are nearly separated and need to be updated using small angle rotations to avoid introducing signal swapping. Most two-stage BSS algorithms can be easily modified so they find a rotation matrix, $\mathbf{R}_z$, only using small angles. Equations (12) and (13) show how $\mathbf{R}_z$ is used to find the independent signal estimates and to update the rotation matrix.

$$\hat{\mathbf{S}}_k' = \mathbf{R}_{k-1}\mathbf{V}_k^T = \mathbf{R}_{k-1}\mathbf{R}_k\hat{\mathbf{S}}_k \tag{11}$$

$$\hat{\mathbf{S}}_k = \mathbf{R}_z\left(\mathbf{R}_{k-1}\mathbf{V}_k^T\right) \tag{12}$$

$$\mathbf{R}_k = \mathbf{R}_z\mathbf{R}_{k-1} \tag{13}$$

4 Demonstration of Concept

In this section the results of applying the tracking BSS technique using the BLISS algorithm [7] for the higher-order stage to foetal ECG recordings are presented. It is possible to monitor single or multiple foetuses by placing ECG sensors on the mother's abdomen and analysing the signals. It is hard to observe the weak foetal signals in the outputs from a single sensor due to maternal signals and other electrical interference. BSS offers a way to separate out the weak foetal signals, but this analysis needs to be carried out over long periods of time where the stationarity hypothesis is not valid.

Figure 1 shows the first 10 seconds of the 12 sensor recordings, sampled at 512Hz, demonstrating the small magnitude of the foetal signals in the sensor outputs. This data set is relatively time-invariant has very few interfering signals e.g. muscle noise; this means that the signals are clear and good separation should be achievable.

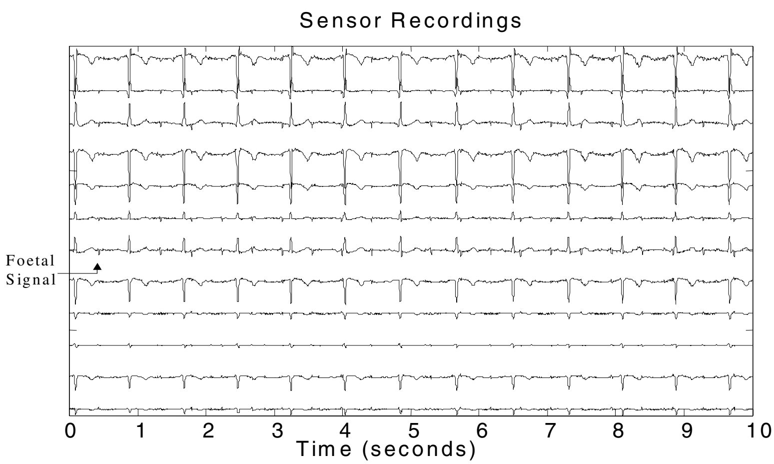

Fig. 1. Section of the sensor recordings

As in many real data analysis problems, qualitative performance measures are hard to find, so qualitative assessments on the quality of the separation must be made.

In figure 2, the first ten seconds of the signals separated by the EASI algorithm (learning rate 0.001) are shown. Note a larger learning rate caused the EASI algorithm to introduce signal swapping. The convergence problems of the EASI algorithm can be seen; signal breakthrough occurs up to 4000 samples into the data set. This effect will follow any sudden change in signal powers or steering vectors. The EASI algorithm did provide good separation on the remaining 290 seconds of this data set.

Figure 3 shows the first ten seconds of the signals separated by the tracking BSS technique (block size 5120 samples, 50% block-to-block overlap); the separation is clearer as no breakthrough is visible, and the F2 is more clearly separated. The time taken to process 1 second of input data by the tracking BSS algorithm was 0.04 seconds – demonstrating that the algorithm can lead to real-time processing. The algorithm was coded in C++ and run on a Dell Latitude 1.2GHz computer, and the figure quoted is for the average of 30 trials. Other experiments have shown that the tracking BSS technique can work on heavily artifact-corrupted data sets.

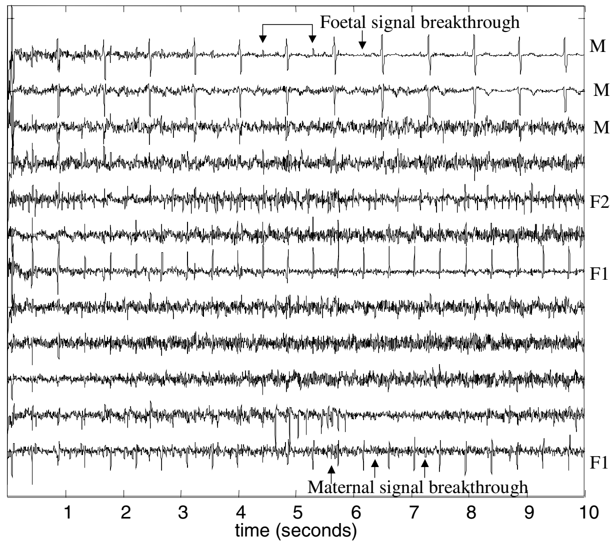

Fig. 2. First ten seconds extracts of the signals separated by EASI. The signals are denoted by M – maternal, F1 – foetus 1 and F2 – foetus 2.

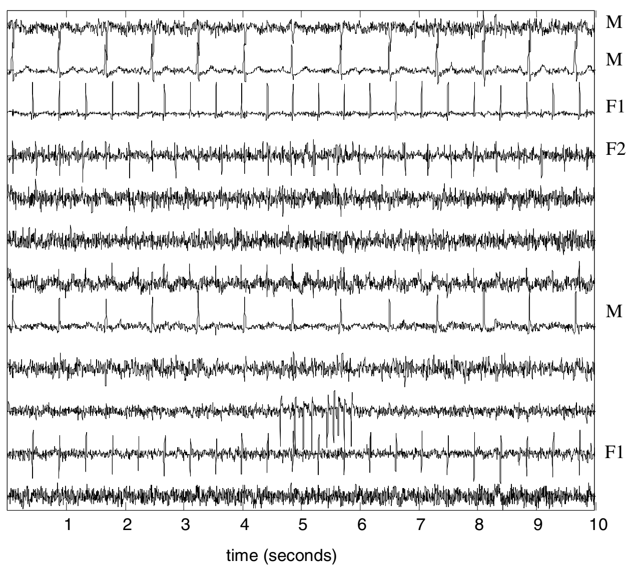

Fig. 3. First ten seconds extracts of the signals separated by the tracking BSS technique developed at QinetiQ. The signals are denoted by M - maternal, F1 - foetus 1 and F2 - foetus 2.

5 Conclusions

In this paper it has been shown that block-based blind signal separation (BSS) methods, combined with a moving window approach, can in principle be used for tracking real, non-stationary signals. However, the use of the moving window principle alone is not sufficient due to the introduction of signal swapping. We overcome this problem and show how past estimates can be efficiently used in the tracking process, to reduce the overall computational cost.

A demonstration of this tracking BSS approach is shown, where it is applied to a five-minute recording of twin foetal ECG data.

References

1. Bach, F., Jordan, M.: Kernel Independent Analysis. Report No UCB/CSD-01-1166, UCSE Berkerley, Presented at ICA2001, November 2001.
2. Bell, A., Sejnowski, T.: An Information-maximisation approach to blind separation and blind deconvolution, Neural Computation 7, 6, 1004-1034, 1995.
3. Cardoso, J., Souloumiac, A.: Blind beamforming for non-Gaussian signals, IEE proc-F, Vol. 140, no. 6, pp. 362-370, Dec 1993.
4. Cardoso, J., Laheld, B.: Equivariant adaptive source separation, IEEE Trans. on Signal Processing, vol. 44, no 12, pp. 3017-3030, Dec. 1996.
5. Haykin, S., Unsupervised adaptive filtering, John Wiley and sons, 2000.
6. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis, Wiley, 2001.
7. McWhirter, J., Clarke, I., Spence, G.: Multi-linear algebra for independent component analysis, SPIE's 44th Annual meeting, The international symposium on Optical Science, Engineering and Instrumentation, Denver USA, 18-23rd July, 1999.
8. Romero, S., Mananas, M., Clos, S., Gimenez, S., Barbanoj, M.: Reduction of EEG artifacts by ICA in different sleep stages, 25th annual conference of the IEEE EMBS, September 2003.
9. Spence, G., Clarke, I., McWhirter, J.: Dynamic Blind Signal Separation, UK Patent Application No. 0326539.4, Patent application filed 14th November 2003.
10. Taylor, M., Smith, M., Thomas, M., Green, A., Chenga, F., Oseku-Affula, S., Wee, L.: Non-invasive fetal electrocardiography in singleton and multiple pregnancies, BJOG: an International Journal of Obstetrics and Gynaecology, Vol. 110, pp. 668–678, July 2003.