

Research Article

The LOST Algorithm: Finding Lines and Separating Speech Mixtures

Paul D. O'Grady¹ and Barak A. Pearlmutter²

¹ Complex and Adaptive Systems Laboratory, University College Dublin, Belfield, Dublin 4, Ireland

² Hamilton Institute, National University of Ireland Maynooth, Co. Kildare, Ireland

Correspondence should be addressed to Paul D. O'Grady, paul.d.ogradey@ucd.ie

Received 26 November 2007; Revised 3 April 2008; Accepted 2 July 2008

Recommended by S. Makino

Robust clustering of data into linear subspaces is a frequently encountered problem. Here, we treat clustering of one-dimensional subspaces that cross the origin. This problem arises in blind source separation, where the subspaces correspond directly to columns of a mixing matrix. We propose the LOST algorithm, which identifies such subspaces using a procedure similar in spirit to EM. This line finding procedure combined with a transformation into a sparse domain and an L_1 -norm minimisation constitutes a blind source separation algorithm for the separation of instantaneous mixtures with an arbitrary number of mixtures and sources. We perform an extensive investigation on the general separation performance of the LOST algorithm using randomly generated mixtures, and empirically estimate the performance of the algorithm in the presence of noise. Furthermore, we implement a simple scheme whereby the number of sources present in the mixtures can be detected automatically.

Copyright © 2008 P. D. O'Grady and B. A. Pearlmutter. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

When presented with a set of observations from sensors such as microphones, the process of extracting the underlying sources is called source separation. Doing so without strong additional information about the individual sources, or constraints on the mixing process, is called *blind source separation* (BSS). Here, we consider instantaneous mixing, where the sources arrive instantly at the sensors with differing amplitude, which is described as follows. A set of T observations of M sensors, $\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T)]$, consist of a linear mixture of N source signals, $\mathbf{S} = [\mathbf{s}(1) | \cdots | \mathbf{s}(T)]$, by way of an unknown linear mixing process characterised by an $M \times N$ mixing matrix \mathbf{A} ,

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad (1)$$

where $\mathbf{x}(t) = [x_1(t) \cdots x_M(t)]^T$ and $\mathbf{s}(t) = [s_1(t) \cdots s_N(t)]^T$ are time-indexed vectors that contain observations and sources, respectively. When $M = N$, the underlying sources, \mathbf{S} , can be separated if one can find an unmixing matrix \mathbf{W} such that $\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$, where $\hat{\mathbf{s}}(t)$ holds the estimated sources at time t , and $\mathbf{W} = \mathbf{A}^{-1}$ up to permutation and

scaling of the rows. This problem can also be described in probabilistic terms:

$$P(\mathbf{s}) = P(s_1, \dots, s_N) = \prod_{i=1}^N P(s_i), \quad (2)$$

where the sources are assumed to be mutually independent, and BSS is achieved after factoring each source's probability density function, $P(s_i)$.

Many source separation algorithms are based on the premise that the sources are independent and identically distributed, for example, independent component analysis (ICA) [1], and achieve separation by making an assumption about the nature of the sources' probability densities. One increasingly popular and powerful assumption is that the sources have a parsimonious representation in a given basis. This assumption has come to be known as the *sparseness assumption*. A signal is said to be sparse when it is zero, or nearly zero, more than might be expected from its variance. Such a signal has a probability density function or distribution of values with a sharp peak at zero and fat tails. This shape can be contrasted with a Gaussian distribution, which has a smaller peak and tails that taper quite rapidly

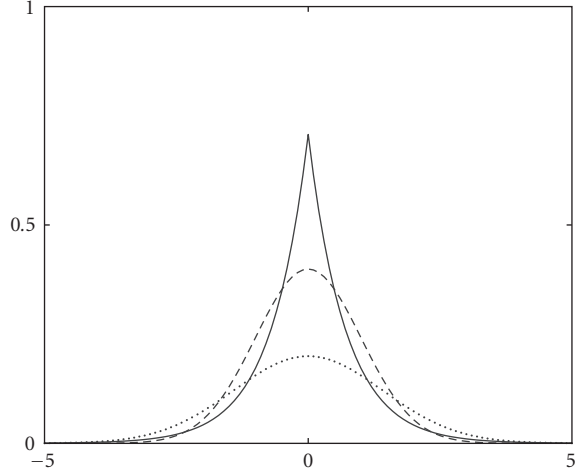


FIGURE 1: A plot of the probability densities of a selection of probability distributions. Solid line: Laplacian distribution; dashed line: Gaussian distribution; dotted line: sub-Gaussian distribution.

(see Figure 1). A standard sparse distribution is the Laplacian distribution,

$$P(c) = \frac{1}{\sqrt{2}} e^{-\sqrt{2}|c|}, \quad (3)$$

which has led to the sparseness assumption being sometimes referred to as a Laplacian prior.

The sparseness of a distribution can be measured by a variety of methods, such as those based on hyperbolic tangent functions [2] and the Gini index [3, 4]. However, the most commonly used measure for unimodal symmetric sparse distributions is kurtosis, which is the degree of peakedness of a distribution:

$$\text{kurt}(c) = \frac{\langle (c - \mu)^4 \rangle}{\sigma^4} - 3, \quad (4)$$

where μ is the mean and σ is the standard deviation. A random variable, c , drawn from a super-Gaussian distribution such as the Laplacian has $\text{kurt}(c) > 0$.

In most situations, a signal of interest will exhibit an inherent structure that defines the underlying components that compose the signal. Without structure a signal is random and of little interest. Typically, structure is not immediately evident from the data and is discovered by identifying an appropriate generative model that describes the structure, then fitting this model to the data by way of a learning algorithm. In the case of BSS, the problem has a dual geometric interpretation, where separation of sources in an audio mixture is equivalent to the separation of linear subspaces in a mixture of oriented lines. Taking a simple example where there are three sources and two mixtures, the generative model takes the form

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}, \quad (5)$$

which can be described as a linear mixture of N linear subspaces in M -space. From (5), it is evident that if only one source is active, say s_1 , then the resultant mixtures would be

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} s_1(t), \quad (6)$$

therefore the points on the *scatter plot* of $x_1(t)$ versus $x_2(t)$ would lie on the line through the origin whose direction is given by the vector $[a_{11} \ a_{21}]^T$. When the sources are sparse, the probability of multiple sources being nonzero simultaneously is low, which indicates that this scenario occurs frequently. Consequently, a scatter plot of coefficients reveals a mixture of lines, with the lines broadened due to noise and occasional simultaneous activity. These line orientations correspond to the columns of \mathbf{A} . Therefore, the essence of the sparse approach is the identification of line orientation vectors from the observed data [5]. In contrast, traditional nonsparse approaches [6–9] exploit the statistics of the sources as opposed to the structure of the mixtures. Moreover, sparseness may be used to perform source separation in the case when there are more sources than mixtures [10], that is, the *under-determined* case. For speech, a sparse representation can often be achieved by a transformation into a suitable domain such as the short-time Fourier transform (STFT) domain (see Figure 2). However, even though \mathbf{X} and \mathbf{S} are complex in the STFT domain, the elements of \mathbf{A} and \mathbf{W} remain real valued, as the form of each depends on the mixing assumption, which remains instantaneous mixing.

We propose the *line orientation separation technique* (LOST), which separates an arbitrary number of sources from an arbitrary number of mixtures by identifying lines in a scatter plot, consequently factorising a mixture of multivariate Laplacian densities. The orientation of each line is estimated using a procedure that is similar in spirit to *Expectation-Maximisation* (EM), and for the under-determined case sources are estimated using L_1 -norm minimisation. We presented an early incarnation of our algorithm in [11]. Here, we present a number of extensions including automatic detection of the number of sources in the mixtures, and improved line estimation through scaling of transform coefficients. Furthermore, we present a more detailed investigation of the algorithm's separation performance, and provide a freely available C code implementation of the algorithm.

This paper is organised as follows. In Section 2, we discuss the identification of overlapping linear subspaces in a scatter plot and present the LOST algorithm. Additionally, we implement a simple scheme whereby the LOST algorithm can automatically detect the number of sources in the mixtures. In Section 3, we investigate the general separation performance of the algorithm, and provide an empirical assessment of the algorithm's robustness to noise. Furthermore, we investigate the performance of the algorithm when automatically detecting the number of sources, and compare the performance of the LOST algorithm to that of the geoICA [12] algorithm. We complete the paper with a discussion in

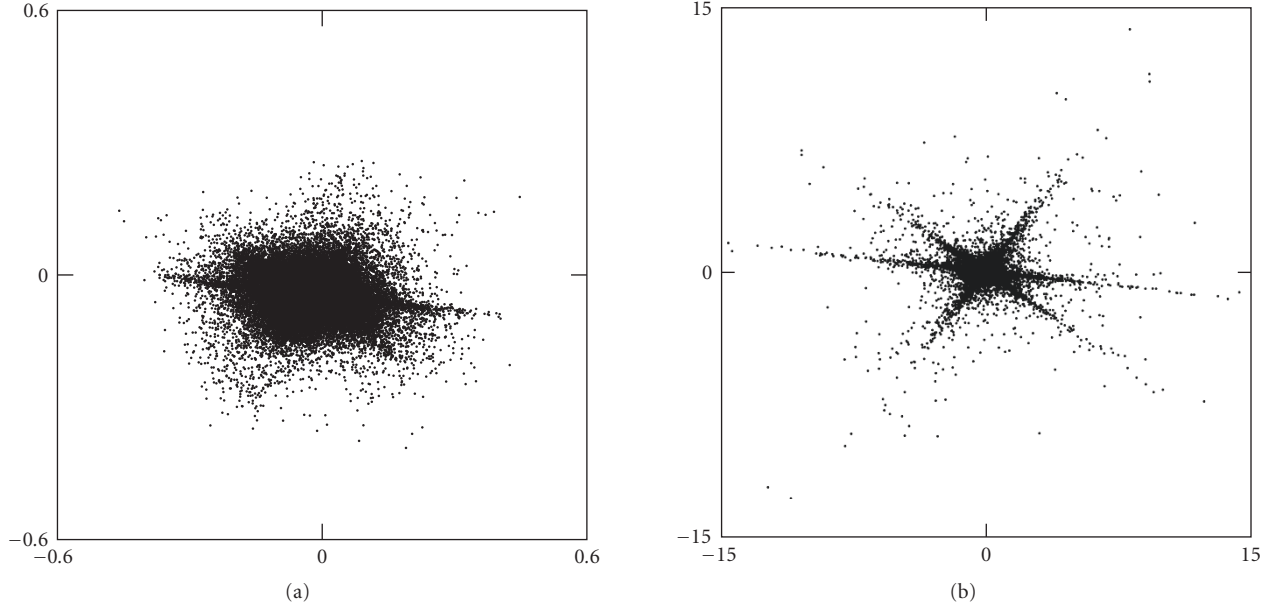


FIGURE 2: Scatter plot of two linear mixtures of three zero-mean speech sources, in both the time domain, (a) $x_1(t)$ versus $x_2(t)$, and the transform domain, (b) $\text{real}(x_1(\omega))$ versus $\text{real}(x_2(\omega))$. The *sparse* transform domain consists of the real coefficients of a 512-point windowed STFT. Each of the figures' axes is measured in arbitrary units of mixture coefficients.

Section 4, and conclusion in Section 5. Details of the sources used in our experiments are presented in the appendix.

2. ORIENTED LINES SEPARATION

It can be seen from the scatter plot of Figure 2 that the columns of \mathbf{A} , which represent the sources, manifest linear subspaces that cross the origin in a sparse domain. Furthermore, it is evident that the points in each linear subspace are drawn from a distribution that is concentrated around the line. Such a distribution resembles a multivariate Laplacian density that is centred *along* the line. Since there are N sources, s_1, \dots, s_N , each characterised by a different Laplacian density, the observations $\mathbf{x}(t)$ are generated by a linear combination of these Laplacian densities; such a model is commonly known as a *Mixture of Laplacians* (MoL) or a *Laplacian mixture model* (LMM). By fitting an LMM to the observed density $P(\mathbf{x})$, the linear subspaces are identified by the Laplacian density centres.

2.1. Mixture of multivariate Laplacians

We propose the following mixture of multivariate Laplacians as a generative model for BSS. The Laplacian density may be expressed by

$$\mathcal{L}(\mathbf{v}|\beta, \mathbf{x}) = \beta e^{-2\beta|\mathbf{x}-\mathbf{v}|} \propto e^{-\beta|\mathbf{x}-\mathbf{v}|}, \quad (7)$$

where \mathbf{v} represents the centre of the Laplacian and β controls the boundary of the density. For our purposes, we use multivariate densities, where the centre of the Laplacian, \mathbf{v} , (which is normalised $\|\mathbf{v}\| = 1$) and the observation $\mathbf{x}(t)$ are vectors that represent lines that cross the origin. We require a metric that measures the distance between such lines; an

appropriate measure is achieved by calculating the difference between $\mathbf{x}(t)$ and the projection of $\mathbf{x}(t)$ onto \mathbf{v} :

$$q_{it} = \|\mathbf{x}(t) - (\mathbf{v}_i \cdot \mathbf{x}(t))\mathbf{v}_i\|, \quad (8)$$

where \cdot denotes the dot product of the Euclidean space. When the Laplacian centre and observation are coincident, q_{it} is at its minimum. We characterised each linear subspace by the following distribution [11]:

$$\mathcal{L}(q_{it}, \beta) = e^{-\beta q_{it}}, \quad (9)$$

and define a mixture of multivariate Laplacians as

$$P(\mathbf{x}(t)) = \sum_i^N \mathcal{L}(q_{it}, \beta) = \sum_i^N e^{-\beta q_{it}}, \quad (10)$$

where speech is assumed to be identically and independently distributed, and β is assumed to be the same for each distribution.

2.2. Line orientation estimation

Here, we describe the procedure used to estimate the parameters of the specified mixture of multivariate Laplacians [11]. Since there are N lines, each with a different orientation vector \mathbf{v}_i , the observations are segregated into sets associated with each line. Segregation is achieved by estimating the probability of an observation belonging to a line:

$$\tilde{q}_{it} = P(\mathbf{x}(t)|\mathbf{v}_i) = \frac{e^{-\beta q_{it}}}{\sum_{i'} e^{-\beta q_{i't}}}, \quad (11)$$

where \tilde{q}_{it} indicates the membership of the observation $\mathbf{x}(t)$ to the line \mathbf{v}_i . Calculating the probability of $\mathbf{x}(t)$

for all \mathbf{v}_i represents a partial or *soft assignment* of the observation to each line. The data set associated with each line can be calculated using the observations, \mathbf{X} , and their soft assignments \tilde{q}_{it} , for all i, t . Alternatively, a *hard assignment* may be used, which corresponds to winner-takes-all assignment, where each observation is assigned to just one line [13]. Furthermore, the algorithm obtained from a hard assignment is a k -means clustering algorithm. Typically, k -means performs vector quantisation, while EM performs density estimation—which fits better with our mixture of multivariate Laplacians model. However, both approaches give similar clusters. Although, in general, k -means will consistently find densities with less overlap than EM [14] and makes a strict sparseness assumption where only one source is expected to be active at any time.

For density mixture models, it is common that each density has a separate β specific to that density. Although, for our algorithm, which utilises a multivariate density model, where Laplacian densities are centred along lines that *cross the origin*, individual β parameters are not possible as they may grow at different rates, over-weighting points close to the origin, that do not belong to the line, potentially squeezing out lines. However, this is not a problem, as it is reasonable to assume that speech is identically and independently distributed, which makes β the same for each distribution.

The orientation of a linear subspace can be thought of as the direction of its greatest variance. One method that can be used to determine the variance of a data set, and its direction, is *principal component analysis* (PCA) [15]. PCA is a dimensionality reduction technique that represents a data set by the variance of the data in orthogonal directions. The principal component with the largest variance, λ_{\max} , which corresponds to the principal eigenvector, \mathbf{u}_{\max} , of the covariance matrix for the weighted observations

$$\Sigma_i = \mathbf{U}_i \Lambda_i \mathbf{U}_i^{-1} \quad (12)$$

identifies the centre of the line [11]

$$\mathbf{v}_i = \mathbf{u}_{\max}, \quad (13)$$

where the columns of the matrix \mathbf{U}_i contain the eigenvectors of Σ_i , and the diagonal matrix Λ_i contains its associated eigenvalues $\lambda_1, \dots, \lambda_M$. However, contrary to our mixture model (10), PCA employs a Gaussian noise model and therefore does not provide a true maximum likelihood estimate of the line, under the Laplacian assumption. Although, PCA may be considered to be the best unbiased linear estimator (BLUE), where the principal eigenvector approaches the maximum likelihood estimate of the line, under the Laplacian assumption, as the number of samples approaches infinity. A similar approach to cluster centre re-estimation using singular-value decomposition is presented in [16], while an alternative approach that fits a straight line to the data points in a linear subspace is presented in [17].

The density boundary parameter β controls the spread of the densities centred on each line. It is obvious from Figure 2 that such a spread may be represented by the variance of the linear subspace that is orthogonal to the line, that is, the

second largest eigenvalue of Λ_i . We estimate the value of β using a scheme that creates a set of second largest eigenvalues for all Σ_i , and update β to the reciprocal of the largest value in this set, $\beta \leftarrow 1/\max(\lambda_1, \dots, \lambda_N)$.

The procedure of soft assignment and line centre repositioning using PCA is repeated until \mathbf{v}_i converge, at which point $\hat{\mathbf{A}}$ is constructed by adjoining the estimated line orientations to form the columns of the matrix

$$\hat{\mathbf{A}} = [\mathbf{v}_1 | \dots | \mathbf{v}_N]. \quad (14)$$

Such a procedure resembles an Expectation-Maximisation (EM) algorithm [18]—or more correctly a pseudo-EM algorithm—which finds maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. The EM algorithm alternates between an expectation (E-)step, which calculates an expectation of the latent variables, and a maximisation (M-)step, which calculates the maximum likelihood estimates of the parameters by maximising the expected likelihood found on the E-step. The parameters found on the M-step are then used to begin another E-step, and the process is repeated.

In our case, the E-step calculates posterior probabilities assigning observations to lines and the M-step repositions the lines to match the points assigned to them. This pseudo-EM procedure comprises the line estimation stage of the LOST algorithm, and is illustrated in Figure 3.

Alternatively, the line estimation stage of the LOST algorithm can be thought of as a piecewise linear operation, where observations are soft assigned to lines, and PCA is performed for the data partially assigned to each line.

2.3. Sparse transformation

In order for the linear subspaces in the scatter plot to be well defined, an appropriate sparse transformation is required. For the LOST algorithm, we exploit the sparseness of speech in the short-time Fourier transform domain, which results in well-defined lines (see Figure 4). However, it is evident that some observations are perturbed by noise, broadening the lines. It is necessary that the lines are as well defined as possible, as the line estimation stage of the LOST algorithm is dependent on the quality of the sparse representation.

The broadening of the lines may be reduced by controlling the effects of the perturbing noise, which may be achieved by segregating the STFT coefficients into different classes based on some notion of noise level. Here, we propose such an approach, where we examine the levels of noise present in each frequency bin over all STFT frames. Since speech is sparse in the STFT domain, we can assume that frequency bins that have a distribution of coefficients that reflect a Gaussian are mostly noise, while frequency bins that exhibit a Laplacian distribution contribute mostly to the definition of the lines; the distinguishing feature between the two distributions is their *peakedness*. We measure the peakedness of the distribution of coefficients for each bin using kurtosis, $\text{kurt}(c_k) = \langle (c_k - \mu)^4 \rangle / \sigma^4 - 3$, where c_k is the distribution of coefficients for the k th frequency bin. Each

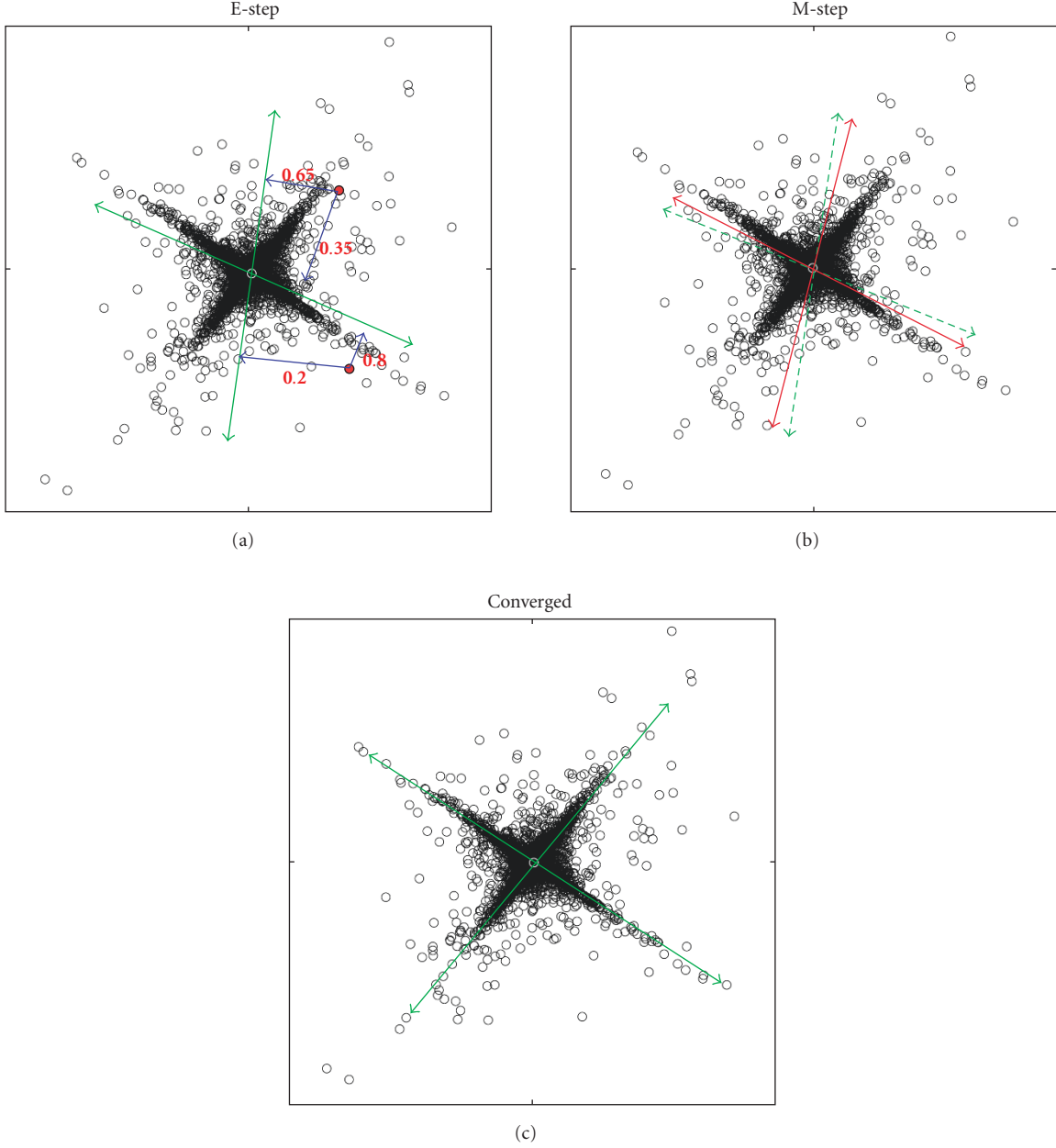


FIGURE 3: Illustration of the LOST algorithm’s line estimation procedure. The E-step calculates posterior probabilities partially assigning data points to line orientation estimates, and the M-step repositions the line orientation estimates to the points assigned to them. After convergence, the estimated line orientations coincide with the linear subspace directions in the scatter plot.

bin is subsequently scaled by its kurtosis, $\text{kurt}(c_k)$. Weighting the frequency bins that have a Laplacian distribution of values greater than those that have a Gaussian pushes those observations away from the origin while pulling the noisy observations toward the origin, resulting in better defined lines and improved line estimates.

The effect of kurtosis scaling the STFT coefficients is illustrated in Figure 4. It can be seen that the kurtosis weighted STFT domain produces the best defined lines, which is especially evident for the two mixtures of two sources scatter plot. The effectiveness of kurtosis scaling is discussed in Section 3.3.

2.4. Automatic detection of the number of sources

For most BSS algorithms, the number of sources, N , present in the mixtures is a parameter that must be manually specified by the user. One of the advantages of the LOST algorithm is that the number of sources can be detected from the mixtures automatically. The principal eigenvalues corresponding to the columns of $\hat{\mathbf{A}}$ indicate the variance of each discovered line. If N is specified to a value greater than the number of actual sources, then a line (or a number of lines) may be represented by many vectors. Consequently, the energy associated with the variance of the line is split among

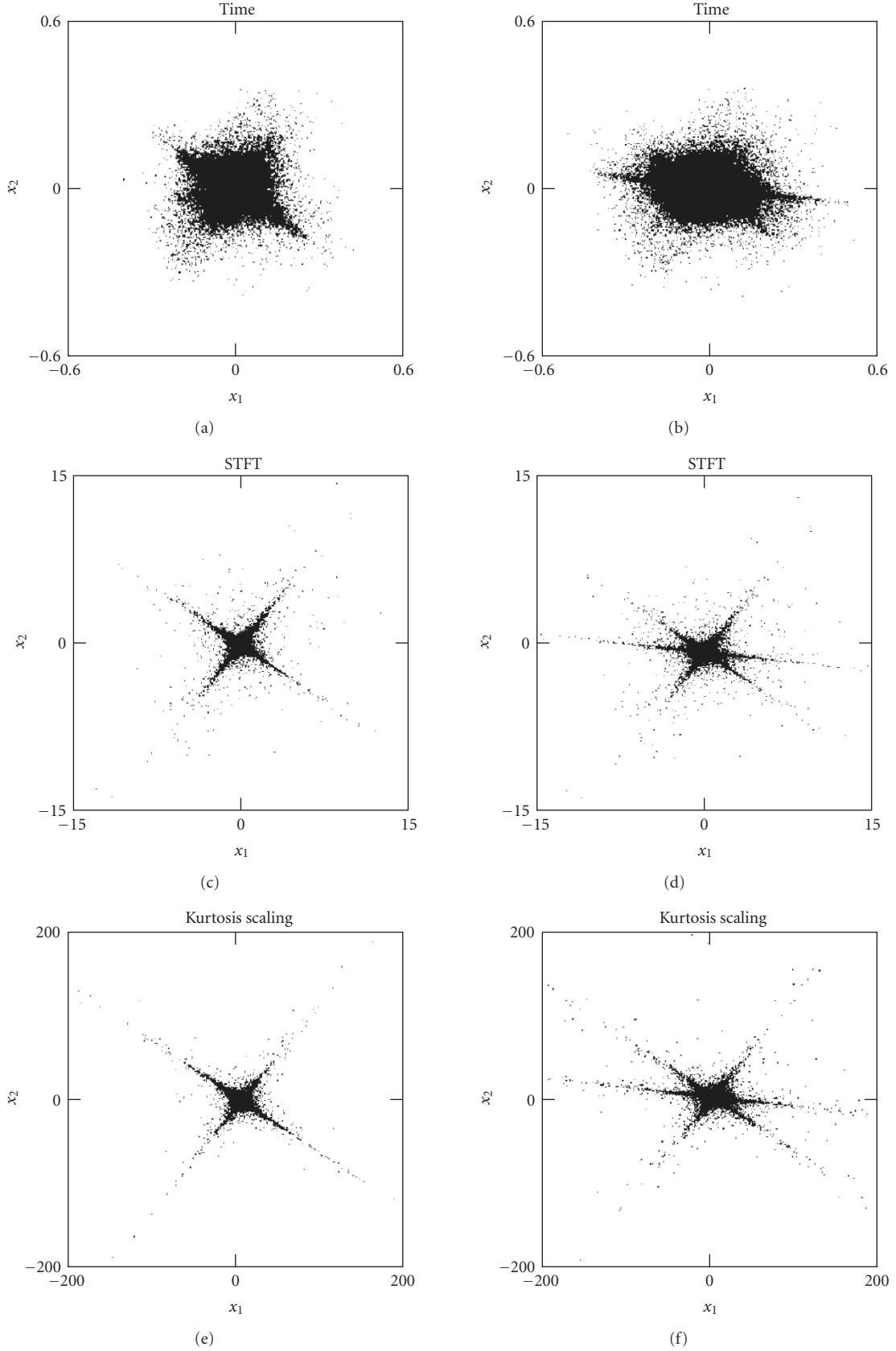


FIGURE 4: Scatter plots for two mixtures of two sources and two mixtures of three sources in the time domain ((a)-(b)), real coefficients of the 512-point STFT domain ((c)-(d)) and kurtosis weighted STFT domain ((e)-(f)). It can be seen that the kurtosis scaled STFT domain produces the best defined lines, which is especially evident for the two mixtures of two sources scatter plot. The figures axes are measured in arbitrary units of mixture coefficients.

the vectors, resulting in the vectors having small principal eigenvalues.

We employ a heuristic that exploits the indicative properties of the principal eigenvalues to identify the number of sources. An upper limit on the number of sources, N_{\max} , is specified in advance and a corresponding number of line orientation vectors are initialised. As the algorithm iterates, extraneous line orientation vectors are pruned as their principal eigenvalues fall beneath a predefined threshold, τ . In this way, the algorithm detects the number of sources in the mixtures. The accuracy of this scheme is discussed in Section 3.4.

2.5. Source unmixing

The dimensionality of the estimated mixing matrix, $\hat{\mathbf{A}}$, determines the procedure used to estimate the sources, $\hat{\mathbf{s}}$. Therefore, so as to be applicable to separation problems that exhibit an arbitrary number of sources and mixtures, the LOST algorithm employs three different source-unmixing methods. For the *even-determined* case, where $M = N$, $\hat{\mathbf{A}}$ is square and the data points can be assigned to line orientations using $\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t)$. When there are more observations than sources, that is, the *over-determined* case ($M > N$), data points can be assigned to sources by finding the least squares solution. When $M < N$, the *under-determined* case, the inverse of \mathbf{A} is ill-posed since $\hat{\mathbf{A}}\hat{\mathbf{s}}(t) = \mathbf{x}(t)$ has more unknowns in \mathbf{s} than knowns in \mathbf{x} , therefore \mathbf{S} needs to be estimated by some other means. One technique is so-called *hard assignment* of coefficients [19–23]. Another is partial assignment, where each coefficient is decomposed into more than one source. For sparse sources, this is generally done by minimisation of the L_1 -norm, which can be seen as a maximum likelihood reconstruction under the assumption that the coefficients are drawn from a Laplacian distribution—this being the method used by the LOST algorithm. For complex data L_1 -norm minimisation can be solved using *second-order conic programming* (SOCP). Alternatively, L_1 -norm minimisation can be implemented by a *linear programming* where the real and imaginary parts are treated separately, thus doubling the number of coefficients. Furthermore, it has been shown that this approach gives solutions that are comparable to, or even better than SOCP, with the added advantage of lower computational cost for low-dimensional problems [24]. Minimisation of the L_1 -norm may also be used for the over-determined case, although the resultant separation performance is essentially the same as for least squares.

2.6. The LOST algorithm summary

The following is a summary of the LOST algorithm, describing both line orientation estimation and source unmixing.

2.6.1. Line estimation

(1) Create a scatter plot of \mathbf{X} in a sparse domain, transform the observations, x_1, \dots, x_M , using an STFT, and perform

kurtosis scaling of the coefficients; the transformed observations are subsequently plotted against each other.

(2) Randomly initialise the N line orientation vectors \mathbf{v}_i , where $\|\mathbf{v}_i\| = 1$ throughout, and initialise β to a sufficiently large value. For the automatic detection of sources, initialise $N = N_{\max}$ line orientation vectors.

(3) Partially assign each observation, $\mathbf{x}(t)$, to each line orientation vector, \mathbf{v}_i , using a soft data assignment:

$$q_{it} = \|\mathbf{x}(t) - (\mathbf{v}_i \cdot \mathbf{x}(t))\mathbf{v}_i\|^2, \quad (15)$$

$$\tilde{q}_{it} = \frac{e^{-\beta q_{it}}}{\sum_{i'} e^{-\beta q_{i't}}},$$

where β controls the boundary between the regions attributed to each line, and \tilde{q}_{it} are the computed weightings of the observation at time t for each line i .

(4) Calculate the covariance matrix of the weighted observations assigned to each line. The covariance matrix expression and assignment weightings are combined as follows:

$$\Sigma_i = \frac{\sum_t \tilde{q}_{it} (\mathbf{x}(t) - \mu) (\mathbf{x}(t) - \mu)^T}{\sum_t \tilde{q}_{it}}, \quad (16)$$

where μ is a vector of the mean values of the rows of \mathbf{X} , which is typically zero for speech, and Σ_i is the covariance of weighted observations associated with line i .

(5) Update the line orientation estimates to the principal eigenvector of each covariance matrix. The eigenvector decomposition of Σ_i is

$$\Sigma_i = \mathbf{U}_i \Lambda_i \mathbf{U}_i^{-1}, \quad (17)$$

where the columns of the matrix \mathbf{U}_i contain the eigenvectors of Σ_i , and the diagonal matrix Λ_i contains its associated eigenvalues $\lambda_1, \dots, \lambda_M$. The new line orientation vector estimate is the principal eigenvector of Σ_i :

$$\mathbf{v}_i \leftarrow \mathbf{u}_{\max}, \quad (18)$$

where \mathbf{u}_{\max} is the principal eigenvector, that is, the eigenvector with the largest eigenvalue, λ_{\max} . For automatic detection of sources, compare all λ_{\max} to the predefined threshold τ , and remove at most one orientation vector which is beneath this threshold.

(6) Update β using the variances that are orthogonal to the direction of the lines, select the second largest eigenvalue from each diagonal matrix Λ_i , and update to the reciprocal of the largest eigenvalue from this set:

$$\beta \leftarrow \frac{1}{\max(\lambda_1, \dots, \lambda_N)}, \quad (19)$$

where λ_i is the second largest eigenvalue of Σ_i . Return to step 3 and repeat until \mathbf{v}_i converge.

(7) After convergence, adjoin the line orientations estimates to form $\hat{\mathbf{A}}$:

$$\hat{\mathbf{A}} = [\mathbf{v}_1 | \dots | \mathbf{v}_N]. \quad (20)$$

Contrasting approaches for mixing matrix estimation include: kernel methods [25], clustering using topographic maps [26], feature extraction using the Hough transformation [23], joint unitary diagonalisation [7], entropy maximisation [6], and independence maximisation [27]. All of which are discussed in [28].

2.6.2. Source unmixing

- (1) Perform LOST line estimation procedure to calculate $\hat{\mathbf{A}}$.
- (2) (a) Even-determined case, source estimates are calculated using linear transformation:

$$\hat{\mathbf{s}}(t) = \mathbf{W}\mathbf{x}(t), \quad t = 1, \dots, T, \quad (21)$$

where $\mathbf{W} = \hat{\mathbf{A}}^{-1}$.

- (b) Over-determined case, source estimates are calculated by finding the least squares solution:

$$\text{minimise } \|\hat{\mathbf{A}}\hat{\mathbf{s}}(t) - \mathbf{x}(t)\|_2, \quad t = 1, \dots, T. \quad (22)$$

- (c) Under-determined case, source estimates are calculated using L_1 -norm minimisation for each observation in the sparse STFT domain, $\mathbf{x}(\omega)$, such that

$$\underset{\hat{\mathbf{s}}(\omega) \in \mathbb{R}^N}{\text{argmin}} \|\hat{\mathbf{s}}(\omega)\|_1 \text{ subject to } \hat{\mathbf{A}}\hat{\mathbf{s}}(\omega) = \mathbf{x}(\omega). \quad (23)$$

(The solution can be found efficiently using linear programming [29]. We introduce vectors \mathbf{s}^+ and \mathbf{s}^- , each with the same dimensionality as $\hat{\mathbf{s}}(t)$, and use the linear constraints $\mathbf{s}^+, \mathbf{s}^- \geq 0$ and $\hat{\mathbf{A}}\mathbf{s}^+ - \hat{\mathbf{A}}\mathbf{s}^- = \mathbf{x}(t)$. The minimisation of $\|\hat{\mathbf{s}}\|_1 = \sum_i |\hat{s}_i|$ becomes the linear objective of minimising $\sum_i (s_i^+ + s_i^-)$. After solving this system, the desired coefficients are $\hat{\mathbf{s}}(t) = \mathbf{s}^+ - \mathbf{s}^-$. When using complex data, as in the case of a STFT representation, we treat the real and imaginary parts separately, thus doubling the number of coefficients.) Subsequent to which, an inverse transformation is performed, $\hat{\mathbf{s}}(\omega) \mapsto \hat{\mathbf{s}}(t)$.

- (3) The final result is an $N \times T$ matrix $\hat{\mathbf{S}}$ that contains the source estimates, $\hat{s}_1, \dots, \hat{s}_N$, in each row.

3. EXPERIMENTS

To demonstrate the performance of the LOST algorithm, we investigate its separation performance when applied to speech mixtures: We use speech sources that are extracted from a commercial audio CD of poems read by their authors [30]; each source is a ten second segment of a poem, which has been down-sampled to 8 kHz; details of the extraction procedure and the poems used are presented in the appendix.

Throughout this section we use the notation $MmNs$ to denote the mixtures, where M and N indicate the number of mixtures and sources respectively, for example, $4m6s$

indicates an instantaneous mixture that has 4 observations of 6 sources. For all experiments, we evaluate the separation performance of the LOST algorithm when applied to the following mixture set: $2m2s$, $2m3s$, $3m2s$, $3m3s$, $3m4s$, $4m3s$, $4m4s$, $4m5s$, and $4m6s$; which includes even-determined, over-determined and under-determined mixtures.

3.1. Performance measurement

For the purposes of ease of comparison with existing separation methods, we evaluate the separation performance of the LOST algorithm using the measures provided by the BSS_EVAL toolbox [31]. The performance measures are based on the principle that a given source estimate, \hat{s} , is composed of the original source and different classes of additive noise:

$$\hat{s}(t) = s(t) + \epsilon_i(t) + \epsilon_n(t) + \epsilon_a(t), \quad (24)$$

where $\epsilon_i(t)$ is noise due to interference from other sources, $\epsilon_n(t)$ is perturbing noise (such as Gaussian noise) and $\epsilon_a(t)$ is the noise due to artifacts (such as musical noise). The noise introduced by each class is estimated by the toolbox and used in the following global performance measures:

- (1) *Source-to-Artifact Ratio* (SAR): Measures the level of artifacts in the source estimate,

$$\text{SAR} = \frac{\|s + \epsilon_i + \epsilon_n\|^2}{\|\epsilon_a\|^2}. \quad (25)$$

- (2) *Source-to-Interferences Ratio* (SIR): Measures the level of interference from the other sources in the source estimate,

$$\text{SIR} = \frac{\|s\|^2}{\|\epsilon_i\|^2}. \quad (26)$$

- (3) *Source-to-Distortion Ratio* (SDR): Provides an overall separation performance criterion,

$$\text{SDR} = \frac{\|s\|^2}{\|\epsilon_i + \epsilon_n + \epsilon_a\|^2}. \quad (27)$$

All performance measures are expressed in dB, where higher performance values indicate better quality estimates.

The order of the elements within the rows of the estimated \mathbf{A} cannot be determined correctly, which may result in incorrect labelling of the returned source estimates, that is, *permutation ambiguity* associated with BSS. Therefore, prior to performance evaluation, we relabel the estimates by calculating the signal-to-noise ratio (SNR) of each source estimate with all the original sources, and assign each estimate the label of source that achieves the largest SNR.

3.2. Transform sparseness

We achieve a sparse representation of the mixtures by exploiting the sparseness of speech in the Short-Time Fourier

Transform domain. In order to find the optimal transform parameters for the data, we perform separation over a wide parameter space and evaluate the estimates. Specifically, we perform an STFT on each mixture where each frame is windowed using a Hamming function over a range of FFT sizes, $\{128, 256, 512, 1024, 2048, 4096\}$, and FFT frame advances, $\{1/16, 1/8, 1/4, 1/2, 1\}$ (expressed in fractions of FFT size). We perform this procedure for each of the previously specified mixtures and repeat for 40 Monte Carlo runs, resulting in a total of 10800 ($6 \times 5 \times 9 \times 40$) LOST algorithm experiments—automatic source detection is not used. Furthermore, the sources used in each mixture are randomly selected from our set of source signals (see the appendix), and are mixed using a randomly generated mixing matrix. The procedure for each experiment is as follows:

- (1) N source signals are randomly selected from the set of sources presented in the appendix, and are mixed using a randomly generated \mathbf{A} , which has normalised columns, resulting in a matrix of observations, $\mathbf{X} = \mathbf{A}\mathbf{S}$.
- (2) The LOST algorithm (Section 2.6) is applied to \mathbf{X} , and the source estimates, $\hat{\mathbf{S}}$, are constructed.
- (3) The estimates and the original sources are used to evaluate the SIR, SAR, and SDR performance of the LOST algorithm.

3.2.1. Results

The results from all experiments are collated and separation performance is calculated as follows: The performance values of the source estimates for each experiment are averaged, which are themselves averaged over 40 Monte Carlo runs. The worst, median and best performances results, and the transform parameters that achieved these results are tabulated in Table 1; average values for β and iterations are also tabulated. As indicated in Figure 2 the sparseness of the coefficients in the transform domain will have an important effect on how well defined the line orientations will be, which ultimately controls the separation performance of the LOST algorithm. The results show that a frame size of 4096 produces the worst separation performance for all three measures, which indicates that speech sampled at 8 kHz is not sufficiently sparse in this domain. Median performance is achieved for a frame size of 128 or 256, while the best performance is achieved for 512 and 1024. It is evident that the average β values obtained for the best performance values are smaller than all others, indicating that the line orientations are well defined when using the associated STFT parameters. Furthermore, the best performance experiments typically converge the fastest. Therefore, the sparseness of the transform domain effects not only the separation performance but convergence speed also.

To analyse the performance of the LOST algorithm for STFT parameters that achieve good separation, we select a subset of the experiments that have a frame size of 512 or 1024 (which results in a total of 400 experiments for

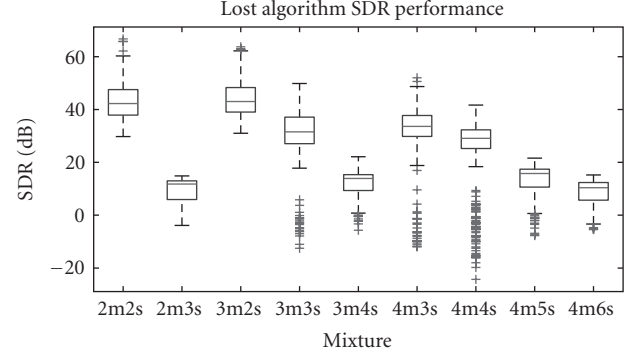


FIGURE 5: SDR results for the LOST algorithm: Box plots are used to illustrate the performance results for each mixture, with each box representing the median and the interquartile range of the results. For SDR, which represents overall separation performance, separation performance decreases as M & N increase, which decreases further as N increases greater than M .

each mixture) and represent the results using box plots: Each box presents information about the median and the statistical dispersion of the results. The top and bottom of each box represents the upper and lower quartiles, while the length between them is the interquartile range; the whiskers represent the extent of the rest of the data, and outliers are represented by $+$. Box plots for SDR, SIR, and SAR are presented in Figure 5, Figure 6, and Figure 7 respectively.

The performance values for SDR indicate that over-determined mixtures produce the best results, while under-determined mixtures produce the worst, which is to be expected for under-determined mixtures, as there are more unknowns in \mathbf{s} than knowns in \mathbf{x} . The general trend in SDR performance is that as M and N increase together, separation performance decreases, which decreases further as N increases greater than M : Comparing 2m2s and 4m4s for example, both are even-determined mixtures, however the median SDR achieved for 4m4s is lower than 2m2s, indicating that an increase in M and N together degrades performance. Furthermore, as indicated by the median SDR for 4m4s, 4m5s & 4m6s, when M is fixed and N increases greater than M , SDR performance degrades further. However, when N is fixed and M increases, as is the case for 2m3s, 3m3s & 4m3s, SDR performance increases.

For SAR performance, the large distances between the median values for the even-determined and under-determined performance results illustrate the high level of artifacts present in the under-determined mixture source estimates. Listening to these estimates reveals the presence of portions of the other sources in the estimates. Such artifacts are not audible for the even-determined or over-determined source estimates, and are produced by L_1 -norm minimisation when more than M sources are active at the same time. This contrasts with SIR, where the difference between even-determined and under-determined performance is not so great.

It is worth noting that over all performance measures, increasing the number of observations for an even-determined mixture, does not greatly improve separation

TABLE 1: The relationship between transform parameters and the separation performance of the LOST algorithm; average separation performance over 40 Monte Carlo runs for each experiment.

Measure	Rand. Mix.	Worst performance					Median performance					Best performance				
		FFT param.		β	Iter.	Avg. res. (dB)	FFT param.		β	Iter.	Avg. res. (dB)	FFT param.		β	Iter.	Avg. res. (dB)
		Frame	Adv.				Frame	Adv.				Frame	Adv.			
SDR	2m2s	128	32	45.07	10.98	35.49	128	8	36.78	9.35	39.0	512	128	27.27	7.57	42.55
	2m3s	256	32	61.5	20.96	2.25	128	8	12.18	23.98	8.50	1024	256	10.14	18.2	11.15
	3m2s	128	128	75.93	10.48	35.20	256	16	28.11	7.14	41.94	1024	64	13.18	9.32	46.88
	3m3s	4096	1024	9.6	23.05	22.52	128	128	92.82	16.07	28.2	1024	128	5.86	15.8	32.97
	3m4s	4096	4096	34.22	27.49	5.86	128	8	67.14	30.53	10.94	1024	512	7.01	21.93	13.75
	4m3s	4096	2048	12.56	20.57	25.66	256	16	22.59	15.1	30.59	512	32	8.04	14.83	34.95
	4m4s	4096	2048	8.01	28.93	18.16	128	16	73.33	20.9	23.60	512	32	6.47	22.71	28.91
	4m5s	128	8	42.28	33.14	6.21	128	16	63.71	36.05	11.60	512	128	5.45	32.36	15.97
	4m6s	4096	4096	30.47	41.88	1.29	128	8	57.06	48.49	8.1	512	64	4.85	47.46	11.38
SIR	2m2s	128	32	45.07	10.98	35.50	128	8	36.78	9.35	39.16	1024	1024	27.38	7.54	42.72
	2m3s	256	32	61.5	20.96	9.47	128	8	12.18	23.98	13.38	1024	1024	14.14	17.84	16.51
	3m2s	128	128	75.93	10.48	35.22	256	16	28.11	7.14	42.7	1024	64	13.18	9.32	47.20
	3m3s	4096	1024	9.6	23.05	22.52	128	128	92.82	16.07	28.3	1024	128	5.86	15.8	32.98
	3m4s	4096	4096	34.22	27.49	11.79	256	16	17.64	26.19	15.60	1024	512	7.01	21.93	18.94
	4m3s	4096	2048	12.56	20.57	25.66	256	16	22.59	15.1	30.59	512	32	8.04	14.83	34.95
	4m4s	4096	2048	8.01	28.93	18.16	128	16	73.33	20.9	23.61	512	32	6.47	22.71	28.92
	4m5s	128	8	42.28	33.14	11.13	256	32	15.22	29.6	15.18	512	128	5.45	32.36	19.97
	4m6s	4096	4096	30.47	41.88	7.86	256	16	16.12	38.88	12.22	512	64	4.85	47.46	15.94
SAR	2m2s	512	32	57.4	10	66.88	128	8	36.78	9.35	69.3	128	8	36.78	9.35	70.15
	2m3s	512	64	108.55	23.87	5.63	512	128	31.87	20.31	13.60	512	128	31.87	20.31	20.26
	3m2s	128	128	75.93	10.48	69.31	256	16	28.11	7.14	71.80	256	16	28.11	7.14	72.44
	3m3s	4096	1024	9.6	23.05	64.35	128	32	81.74	16.34	65.98	512	64	9.27	17.59	66.78
	3m4s	4096	4096	34.22	27.49	9.83	256	16	17.64	26.19	16.27	512	128	10.71	24.81	20.25
	4m3s	1024	128	5.86	23.6	67.7	128	8	75.01	20.07	68.68	512	32	8.04	14.83	69.49
	4m4s	1024	128	3.92	29.5	62.56	128	16	73.33	20.9	64.3	1024	512	4.7	26	65.45
	4m5s	128	8	42.28	33.14	11.97	128	16	63.71	36.05	19.14	1024	128	2.83	43.05	23.74
	4m6s	4096	4096	30.47	41.88	4.43	128	8	57.06	48.49	13.78	1024	128	2.41	59.37	16.88

performance. For example, we can see from inspection of the results for the mixtures 3m3s & 4m3s that the additional observation provides a small increase in performance, the same is also true for 2m2s & 3m2s. Such an incremental improvement may defy preconceptions, but is typical of BSS algorithms. A plot of the estimates for 4m6s produced by the LOST algorithm is presented in Figure 8.

It is evident that there are many low-performance outliers in the box plots, this is due to the random mixing matrices used to generate our mixtures. Such randomly generated mixtures may produce scatter plots that contain lines that are too close for the LOST algorithm to separate effectively, that is, \mathbf{A} is an ill-conditioned matrix. The presence of outliers may be ameliorated by discarding random matrices that have a poor condition number. However, in the interests of rigorously testing the algorithm, the authors have chosen not to implement such a scheme for these experiments.

Overall, the LOST algorithm provides very good results for the blind source separation of even-determined and

over-determined mixtures, and successfully achieves separation of under-determined mixtures with good separation performance.

3.3. Robustness to noise

We perform an empirical investigation on the separation performance of the LOST algorithm when Gaussian noise is added to \mathbf{S} . The noise added to each source is measured using the signal-to-noise ratio and is expressed in dB. We perform experiments where Gaussian noise of the following intensities is added to the each source: 20 dB, 15 dB, 10 dB, 5 dB, and 2 dB. As a means of comparison, we also perform an experiment where no noise (∞ dB) is added to the sources. We run the LOST algorithm without automatic source detection using an FFT frame size of 512 and frame advance of 128. In contrast to the experimental procedure presented in Section 3.2, each mixture is generated using a fixed mixing matrix and fixed set of sources, which is necessary as we are only interested in robustness to noise

TABLE 2: Average separation performance for the LOST algorithm on noisy mixtures, with and without kurtosis scaling.

Measure	Fixed mix.	FFT param.		With kurtosis scaling						Without kurtosis scaling					
		Frame	Adv.	Avg. res. (dB) for added noise (SNR)						Avg. res. (dB) for added noise (SNR)					
				None	20 dB	15 dB	10 dB	5 dB	2 dB	None	20 dB	15 dB	10 dB	5 dB	2 dB
SDR	2m2s	512	128	55.95	59.17	49.11	50.20	49.13	33.76	41.88	41.81	42.59	42.57	46.51	31.35
	2m3s			11.29	10.93	10.37	9.18	7.13	4.82	11.15	10.79	10.24	9.7	7.8	4.86
	3m2s			35.3	33.14	31.5	28.39	25.81	23.10	27.3	26.52	25.82	24.10	20.94	16.33
	3m3s			30.22	29.11	27.74	24.72	20.58	14.37	30.25	29.67	28.46	25.6	19.29	11.24
	3m4s			13.80	13.71	13.10	11.68	9.32	5.72	9.43	13.37	12.98	11.90	9.80	6.31
	4m3s			34.87	33.7	30.12	31.73	29.60	25.12	34.67	33.52	32.5	27.93	21.63	13.8
	4m4s			29.85	28.62	27.32	24.82	20.71	15.26	27.31	26.27	24.97	22.12	16.63	9.62
	4m5s			19.79	17.50	16.34	14.34	11.19	8.31	17.87	17.5	14.88	13.7	10.34	7.21
	4m6s			12.17	11.95	11.22	9.54	7.25	4.64	13.32	12.92	11.45	9.81	7.22	0.15
SIR	2m2s	512	128	55.98	59.29	49.11	50.21	49.14	33.76	41.88	41.81	42.59	42.58	46.54	31.35
	2m3s			16.64	16.7	15.29	13.65	10.77	7.43	16.49	15.96	15.22	13.62	10.74	7.30
	3m2s			35.3	33.14	31.5	28.39	25.81	23.10	27.3	26.52	25.82	24.10	20.94	16.33
	3m3s			30.22	29.11	27.74	24.72	20.58	14.37	30.25	29.67	28.46	25.6	19.29	11.24
	3m4s			19.12	18.77	17.84	15.87	12.51	7.33	13.12	18.29	17.60	16.9	13.18	8.67
	4m3s			34.87	33.7	30.12	31.73	29.60	25.12	34.67	33.52	32.5	27.93	21.63	13.8
	4m4s			29.85	28.62	27.32	24.82	20.71	15.26	27.31	26.27	24.97	22.12	16.63	9.62
	4m5s			26.51	23.70	22.28	19.72	15.21	10.82	24.5	22.92	19.85	17.48	13.63	9.3
	4m6s			17.85	17.49	16.53	14.40	11.28	7.52	19.27	18.69	16.75	14.65	11.3	1.44
SAR	2m2s	512	128	77.47	77.29	77.29	77.32	78.18	79.28	77.48	77.26	77.27	77.32	78.17	79.29
	2m3s			12.93	12.67	12.23	11.34	10.8	9.33	12.93	12.65	12.19	11.27	10.5	9.47
	3m2s			74.24	74.7	74.5	74.67	74.52	76.22	74.7	73.97	73.96	74.53	74.35	75.92
	3m3s			74.52	74.50	74.56	74.72	75.4	75.39	74.50	74.47	74.57	74.72	74.95	74.88
	3m4s			16.2	16.8	15.67	14.78	13.79	15.82	16.24	15.52	15.36	14.76	14.4	15.7
	4m3s			76.40	76.62	76.90	74.69	75.12	75.74	76.29	76.35	76.34	76.60	77.5	77.81
	4m4s			66.87	66.88	66.88	66.83	66.93	68.57	66.78	66.77	66.76	66.63	66.42	67.45
	4m5s			21.79	22.20	19.60	16.1	13.62	12.48	19.44	18.75	16.97	15.46	13.73	13.63
	4m6s			13.93	13.76	13.13	11.84	10.8	8.85	14.80	14.48	13.29	11.95	10.27	15.66

and not general separation performance. Additionally, we evaluate the performance of the LOST algorithm with and without kurtosis scaling of the STFT coefficients.

3.3.1. Results

The results from all experiments are collated and averaged as before, and separation performance for each experiment is presented in Table 2. It is evident that the SIR performance results degrade for all mixtures as the level of noise increases, this reflects the perturbation of the line orientations by the random noise, which influences the level of interference from other sources that will be present in the source estimates.

The SAR performance remains relatively constant for the even-determined and over-determined mixtures over all noise levels, while the results for the under-determined results gradually degrade as noise increases. This degradation in performance demonstrates that L_1 -norm minimisation is generally unstable for perturbation of \mathbf{A} . Furthermore, the results show that SAR is largely unaffected by the kurtosis scaling of the transform coefficients, which demonstrates that kurtosis scaling has no effect on the presence of artifacts.

TABLE 3: Typical run times for the LOST algorithm on 10 second mixtures, using a frame size of 512 and frame advance of 128.

Mixture	Time	Mixture	Time
2m2s	7 s	4m3s	20 s
2m3s	30 s	4m4s	18 s
3m2s	8 s	4m5s	45 s
3m3s	14 s	4m6s	60 s
3m4s	40 s		

Overall performance, as indicated by SDR, demonstrates that the LOST algorithm achieves good separation results over all noise levels. Furthermore, kurtosis scaling improves separation performance for all mixtures at all noise levels, however it is particularly effective for even-determined and over-determined mixtures. The tabulated results demonstrate that the LOST algorithm is an effective algorithm for blind source separation of over-determined, even-determined and under-determined mixtures, even in the presence of noise.

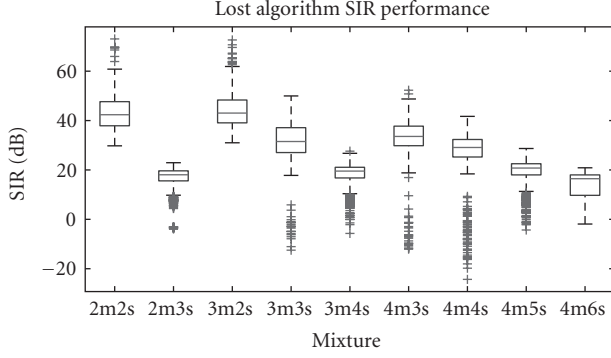


FIGURE 6: SIR results for the LOST algorithm: Box plots are used to illustrate the performance results for each mixture, with each box representing the median and the interquartile range of the results. The results indicate that the source estimates become more resilient to interference from other sources as M increases relative to N .

To illustrate the convergence of the LOST algorithm, convergence curves for both β and the norm of $\hat{\mathbf{A}}$ are presented for each mixture in Figure 9; the curves correspond to the experiments presented in Table 2 where kurtosis scaling is performed and no noise is added. It is evident that both β and $\hat{\mathbf{A}}$ converge to stable results after a small number of iterations, demonstrating the fast convergence properties of the LOST algorithm.

We implemented the LOST algorithm in C code, where version 1.00 was used in our experiments. Furthermore, all the experiments presented were run on a 3.06 GHz Intel Pentium-4 based computer with 768 MB of RAM running the Debian GNU/Linux operating system; typical run times for a frame size of 512 and frame advance of 128 are presented in Table 3.

3.4. Accuracy of automatic source detection scheme

Here, we investigate the accuracy of the automatic source detection scheme employed by the LOST algorithm. In our experiments we specify $\tau = 1.5$ and generate each mixture using a fixed mixing matrix and fixed set of sources, as in Section 3.3. Furthermore, we run the LOST algorithm using an FFT frame size of 512 and frame advance of 128. We repeat each experiment for 100 Monte Carlo runs and present the results in Table 4.

3.4.1. Results

The results from the experiments show that the scheme achieves 100% accuracy for our even-determined and over-determined mixtures over 100 Monte Carlo runs. For the under-determined case, mixtures 3m4s and 4m5s provide almost perfect accuracy; however, as the number of sources increases greater than the number of mixtures, accuracy deteriorates, as is the case for mixture 4m6s with 77% accuracy. Although, the accuracy of the results may be improved by adjusting τ .

TABLE 4: Accuracy of the LOST algorithm's source detection scheme; average principal eigenvalues with standard deviations are also presented. results are for 100 Monte Carlo runs of each experiment.

Mixture	Results	
	Avg. prin. eigenvalue	Identified N correctly
2m2s	2.82 ± 0	100%
2m3s	4.79 ± 0	100%
3m2s	3.27 ± 0	100%
3m3s	8.66 ± 0	100%
3m4s	4.88 ± 0.01	99%
4m3s	6.02 ± 0	100%
4m4s	5.40 ± 0	100%
4m5s	6.49 ± 0.13	98%
4m6s	6.31 ± 0.02	77%

The average principal eigenvalues over all runs reveal that there is a variance of results for the mixtures where 100% accuracy was not achieved. This variance is caused when the energy in the principal eigenvalues is split among extraneous line orientation vectors when over-estimation of the number of sources occurs. Furthermore, we have observed that the detection scheme only ever over estimates the number of sources in the mixtures.

3.5. LOST versus geoICA

One of the main advantages of the LOST algorithm is that it provides a solution for the under-determined case where $M \geq 2$. In order to demonstrate the usefulness of the LOST algorithm when applied to under-determined mixtures, we compare its performance to the geoICA algorithm [12], which also provides a solution for the under-determined case where $M \geq 2$ (Matlab implementations for geoICA and GCE are available at <http://www.biologie.uni-regensburg.de/Biophysik/Theis/research/geoICA.zip>). We test both algorithms using the previously specified mixtures; where \mathbf{A} is randomly generated and the sources are randomly selected as in Section 3.2. Furthermore, each experiment is repeated for 40 Monte Carlo runs. For the LOST algorithm a FFT size of 512 and frame advance of 128 is used, geoICA does not specify a STFT. However, in order to place both algorithms on an even footing in terms of mixture sparseness, we perform geoICA using speech that is STFT transformed using the same parameters as those specified for the LOST algorithm. Furthermore, we use geoICA with its default number of iterations, which is 10 times the number of samples.

The geoICA algorithm specifies no method to separate the sources once $\hat{\mathbf{A}}$ is found (such as L_1 -norm minimisation), therefore we measure the performance of the algorithms using the *Generalised Crosstalk Error* (GCE) [12] between \mathbf{A} and $\hat{\mathbf{A}}$:

$$\text{GCE} = \min_{\mathbf{L} \in \Pi} \|\mathbf{A} - \hat{\mathbf{A}}\mathbf{L}\|, \quad (28)$$

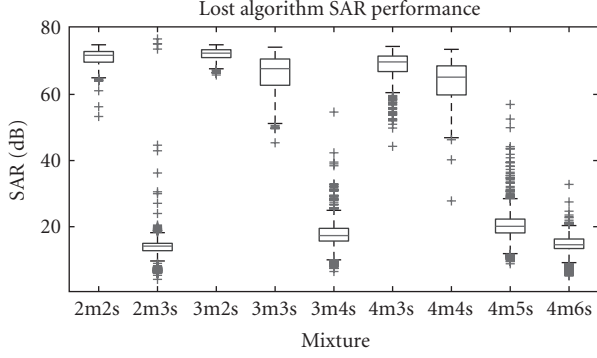


FIGURE 7: SAR results for the LOST algorithm: Box plots are used to illustrate the performance results for each mixture, with each box representing the median and the interquartile range of the results. For SAR, it is evident that there are large differences between the performances achieved for even-determined and under-determined mixtures, which is a consequence of the artifacts produced by L_1 -norm minimisation.

TABLE 5: Average GCE with standard deviations for LOST and geoICA over 40 Monte Carlo runs for each experiment; smaller values indicate better performance.

Mixture	Algorithm		
	LOST	geoICA	geoICA + STFT
2m2s	0.35 ± 0.83	0.45 ± 0.64	0.08 ± 0.2
2m3s	0.37 ± 0.97	0.94 ± 0.45	0.84 ± 0.53
3m2s	0.65 ± 1.45	0.29 ± 0.5	0.1 ± 0.33
3m3s	0.08 ± 0.1	1.22 ± 0.61	0.96 ± 0.68
3m4s	0.29 ± 0.54	1.76 ± 0.54	1.28 ± 0.7
4m3s	0.07 ± 0.02	1.52 ± 0.8	1.02 ± 0.84
4m4s	0.54 ± 0.88	2.04 ± 0.71	1.41 ± 0.73
4m5s	0.46 ± 0.67	2.87 ± 0.82	2.08 ± 0.72
4m6s	0.91 ± 1.01	3.78 ± 0.81	2.4 ± 0.81

where the minimum is taken over the group Π of all invertible matrices having only one non-zero entry per column. When \mathbf{A} and $\hat{\mathbf{A}}$ are equivalent, GCE vanishes, which indicates that GCE decreases as performance increases.

The results for each experiment are collated, and the average GCE performances, along with their standard deviations, are presented in Table 5. It is evident from the results that the LOST algorithm achieves superior performance over geoICA when applied to the separation of speech mixtures. While geoICA performs well for 2m2s, 2m3s, and 3m2s; it performs badly for all other mixtures, even when the observations are transformed to the STFT domain. The general trend of the results show that geoICA does not perform well when $M > 2$, and while the LOST algorithm does exhibit decreased performance, the scale of degradation is not as great as that exhibited by geoICA. One reason for this may be that geoICA maps the observations to the unit half-sphere, which may cause *edge effects* when the sources lie near or on the equator, as the mapping may fail to consolidate the line’s two halves giving the illusion of two lines, for example, if we take a scatter plot of two mixtures that exhibits two orthogonal

lines that are exactly vertical and horizontal, a mapping to the unit half-sphere will result in one cluster for the vertical line and two for the horizontal line, due to perturbations around the line. Another reason may be the fact that geoICA is a simple clustering approach that does not specify any particular prior, unlike the LOST algorithm, which assumes a Laplacian prior.

4. DISCUSSION

One of the main benefits of our approach is that a solution for the under-determined case can be found. In contrast to other similar sparse methods [32], the LOST algorithm is not constrained to just two mixtures. Furthermore, by comparison with the geoICA algorithm, we have demonstrated that the LOST algorithm produces good results when $M > 2$. Recently, modifications have been proposed that extend the *DUET* [33] blind source separation algorithm to the case where $M \geq 2$ [34]. Although, unlike the LOST algorithm, user intervention is required to identify sources. However, further extensions that employ k -means clustering for source identification have been proposed [35].

The performance of the LOST algorithm is heavily influenced by how well defined the linear subspaces are in the transform domain. Therefore, the sparse domain transformation is an integral component of the algorithm, and appropriate selection of such is required to provide useful results. We use the STFT transform, which achieves good separation performance for speech mixtures when an FFT frame size of 512 or 1024 is used. Alternative transformations such as Gabor or wavelet could also be used.

The algorithm we present is a batch operation algorithm, which operates on the entire set of observations. Conversely, an online approach that operates on an observation-by-observation basis is also possible. We have previously presented such an algorithm [13], where the PCA computations of the batch algorithm are replaced by the stochastic gradient algorithm, which converges to the direction of the largest variance of its input data. Moreover, the source unmixing stage is also computed in an online manner.

The scheme we use for line estimation involves updating the current line estimates to the principal eigenvector of the covariance matrix associated with each line. While this is a perfectly acceptable assumption for small values of M . For very large hyper spaces, where M is large, such a scheme may not produce an optimal estimate of the direction of linear subspace. The same is also true for the β update. Therefore, to more accurately estimate the direction and width of a linear subspace in a high dimensional space, a more sophisticated scheme using the provided eigenvalues may be required.

The scheme we implement to automatically detect the number of sources present in the mixtures requires a threshold value, τ , for our data $\tau = 1.5$ works well. However, when the LOST algorithm is applied to other data sets our choice for τ may not be optimal. In this event, a good guess for τ can be gleaned from the principal eigenvalues, which are presented in a data file when our C code implementation of the LOST algorithm is run.

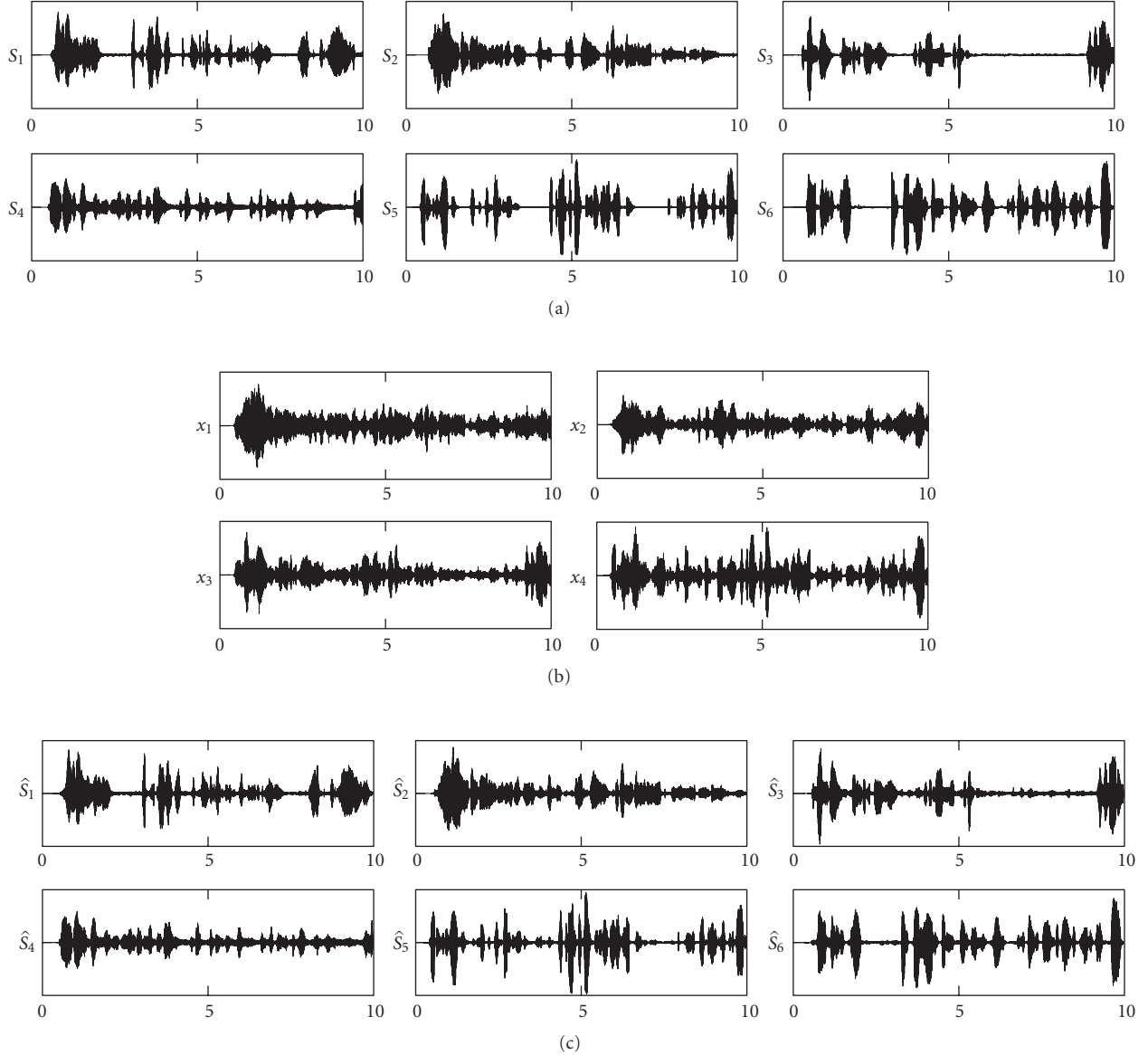


FIGURE 8: Source estimate plots for the LOST algorithm. The plots above show ten second clips of six acoustic sources, s_1, \dots, s_6 ; 4 mixtures, x_1, \dots, x_4 ; and 6 source estimates, $\hat{s}_1, \dots, \hat{s}_6$. Sound wave pressure is plotted against time in units of seconds.

The LOST algorithm is specific to the instantaneous mixing case. However, it has been demonstrated that scatter plot representations can also be used in the anechoic case, where source arrival delays between sensors are also considered (See [28] for a discussion of the anechoic & echoic generative model). A method for anechoic unmixing where the amplitude and delay parameters of the mixing process are segregated into two matrices is presented in [36]. The amplitude parameters are discovered using a line estimation procedure (kernel density estimation [25]) similar to the LOST algorithm, where a scatter plot is formed from the magnitudes of the complex-valued observations. The estimated delay matrix is formed by taking the real and imaginary coefficients assigned to each source in the previous operation, and iteratively rectifying the delay parameter

until the kernel function of the data is maximised. The procedure is repeated for the N sources and the resultant delay parameters form the estimated delay matrix. Following such an approach, it may be possible to extend the LOST algorithm to the anechoic case.

Unlike other BSS algorithms [12, 32], the source identification procedure of the LOST algorithm is not prone to edge effects (as previously discussed in Section 3.5), which enables the LOST algorithm to separate arbitrarily positioned sources.

Throughout our experiments, we have observed on occasion that the random initialisation of \mathbf{A} affects the performance of the line estimation procedure. Sensitivity to initial conditions is common among clustering algorithms, and in the case of the LOST algorithm, from our experience

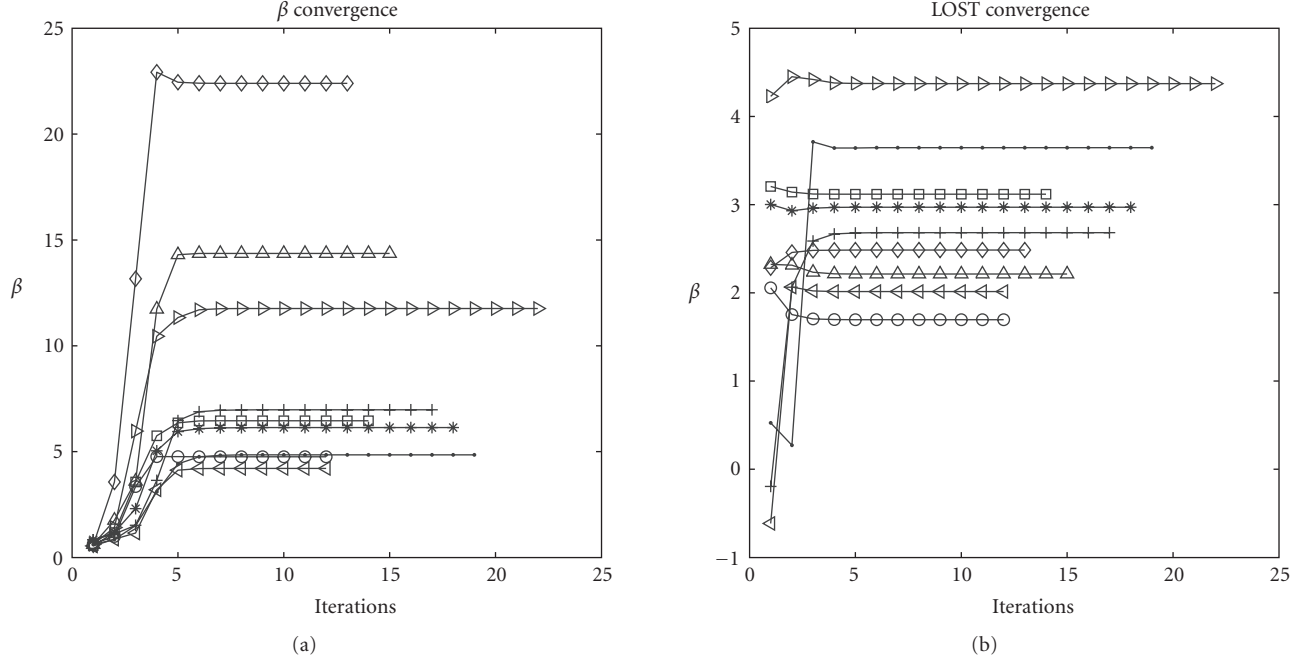


FIGURE 9: LOST algorithm convergence plots for the following experiments: 2m2s \circ , 2m3s \square , 3m2s \diamond , 3m3s Δ , 3m4s \triangleright , 4m3s \triangleleft , 4m4s $+$, 4m5s \star , 4m6s \bullet ; the convergence of the mixing matrix, $\hat{\mathbf{A}}$ is presented on the right, while convergence of the boundary value, β , is presented on the left. It is evident that both β and $\hat{\mathbf{A}}$ quickly converge to stable values.

such a scenario is indicated when $\beta < 1$. In this event, we suggest that \mathbf{A} is reinitialised and that the experiment is repeated.

Finally, occasionally we observe that the scheme we use for the adaption of β causes the parameter to grow without bounds. This typically happens when the transform parameters selected produce scatter plots that are not well defined. When this behaviour is observed, we recommend that β is fixed to some suitably large value. Alternatively, we have observed that increasing the dynamic range of the mixtures works on occasion.

5. CONCLUSION

In this paper, we presented an algorithm that identifies linear subspaces that cross the origin, we have illustrated how such a problem arises in the context of blind source separation of instantaneous mixtures, where mixture matrix columns correspond to linear subspaces in a scatter plot. This method, combined with a transformation into a sparse domain and an L_1 -norm optimisation, constitutes the LOST algorithm, which provides a solution for the blind source separation of instantaneous mixtures with an arbitrary number of mixtures and sources. Moreover, we implement a simple scheme that automatically detects the number of sources present in the mixtures, where extraneous line vectors are pruned when the energy of its principal eigenvalue is beneath a predefined threshold. We performed an extensive investigation on the general separation performance of the LOST algorithm using randomly generated mixtures, which yielded good results, and demonstrated the algorithm’s robustness in the presence of noise. Furthermore, we demonstrated that the LOST

algorithm performs well when compared to the geoICA algorithm.

LOST algorithm software

Our C code implementation of the LOST algorithm is released under the GNU General Public License and is freely available for download from the first author’s webpage: <http://ee.ucd.ie/~pogrady/>.

APPENDIX

SOURCES SIGNALS

The source signals are taken from a commercial audio CD of poems read by their authors [30]. The data is recorded as raw 44.1 kHz 16-bit stereo waveforms. Prior to further processing, ten-second clips are extracted, the two signal channels are averaged, and the data is down-sampled to 8 kHz. The scale of the audio data is arbitrary, leading to the arbitrary units on the auditory waveforms presented throughout the paper. The sources are extracted from the following poems:

- s_1 *Coole Park and Ballylee*, by William Butler Yeats.
- s_2 *The Lake Isle of Innisfree*, by William Butler Yeats.
- s_3 *Among Those Killed in the Dawn Raid Was a Man Aged a Hundred*, by Dylan Thomas.
- s_4 *Fern Hill*, by Dylan Thomas.
- s_5 *Ave Maria*, by Frank O’Hara.
- s_6 *Lana Turner Has Collapsed*, by Frank O’Hara.

ACKNOWLEDGMENTS

Supported by Higher Education Authority of Ireland (An tÚdarás Um Ard-Oideachas), and Science Foundation Ireland grant Numbers 00/PI.1/C067 & 05/YI2/I677.

REFERENCES

- [1] P. Comon, "Independent component analysis: a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [2] J. Karvanen and A. Cichoki, "Measuring sparseness of noisy signals," in *Proceedings of the 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA '03)*, pp. 125–130, Nara, Japan, April 2003.
- [3] S. Rickard and M. Fallon, "The Gini index of speech," in *Proceedings of the 38th Conference on Information Science and Systems (CISS '04)*, Princeton, NJ, USA, March 2004.
- [4] N. Hurley, S. Rickard, P. Curran, and K. Drakakis, "Maximizing sparsity of wavelet representations via parameterized lifting," in *Proceedings of the 15th International Conference on Digital Signal Processing (ICDSP '07)*, pp. 631–634, Cardiff, UK, July 2007.
- [5] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.
- [6] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.
- [7] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and É. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.
- [8] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.
- [9] J.-F. Cardoso, "Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '90)*, vol. 5, pp. 2655–2658, Albuquerque, NM, USA, April 1990.
- [10] M. Lewicki and T. J. Sejnowski, "Learning nonlinear overcomplete representations for efficient coding," in *Advances in Neural Information Processing Systems 10*, pp. 556–562, MIT Press, Denver, Colo, USA, 2001.
- [11] P. D. O'Grady and B. A. Pearlmutter, "Soft-LOST: EM on a mixture of oriented lines," in *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '04)*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 430–436, Granada, Spain, September 2004.
- [12] F. J. Theis, E. W. Lang, and C. G. Puntonet, "A geometric algorithm for overcomplete linear ICA," *Neurocomputing*, vol. 56, no. 1–4, pp. 381–398, 2004.
- [13] P. D. O'Grady and B. A. Pearlmutter, "Hard-LOST: modified k -means for oriented lines," in *Proceedings of the Irish Signals and Systems Conference*, pp. 247–252, June–July 2004, Belfast, UK.
- [14] M. Kearns, Y. Mansour, and A. Y. Ng, "An information-theoretic analysis of hard and soft assignment methods for clustering," in *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI '97)*, pp. 282–293, Providence, RI, USA, August 1997.
- [15] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.
- [16] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [17] M. Babaie-Zadeh, A. Mansour, C. Jutten, and F. Marvasti, "A geometric approach for separating several speech signals," in *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '04)*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 798–806, Granada, Spain, September 2004.
- [18] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1976.
- [19] S. T. Rickard and F. Dietrich, "DOA estimation of many W -disjoint orthogonal sources from two mixtures using DUET," in *Proceedings of the 10th IEEE Workshop on Statistical and Array Processing (SSAP '00)*, pp. 311–314, Pocono Manor, Pa, USA, August 2000.
- [20] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems 13*, pp. 793–799, MIT Press, Denver, Colo, USA, 2001.
- [21] L. Vielva, D. Erdogmus, and J. C. Principe, "Underdetermined blind source separation using a probabilistic source sparsity model," in *Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Signal Separation (ICA '00)*, pp. 675–679, Helsinki, Finland, June 2000.
- [22] L. Vielva, D. Erdogmus, C. Pantaleón, I. Santamaría, J. Pereda, and J. C. Principe, "Underdetermined blind source separation in a time-varying environment," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, vol. 3, pp. 3049–3052, Orlando, Fla, USA, May 2002.
- [23] J. K. Lin, D. G. Grier, and J. D. Cowan, "Feature extraction approach to blind source separation," in *Proceedings of the 7th IEEE Workshop on Neural Networks for Signal Processing (NNSP '97)*, pp. 398–405, Amelia Island, Fla, USA, September 1997.
- [24] S. Winter, W. Kellermann, H. Sawada, and S. Makino, "MAP-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and ℓ_1 -norm minimization," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 24717, 12 pages, 2007.
- [25] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.
- [26] M. M. van Hulle, "Clustering approach to square and non-square blind source separation," in *Proceedings of the 9th IEEE Workshop on Neural Networks for Signal Processing (NNSP '99)*, pp. 315–323, Madison, Wis, USA, August 1999.
- [27] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural models," in *Proceedings of AIP Conference on Neural Networks for Computing (AIP '86)*, pp. 206–211, Snowbird, Utah, USA, April 1986.
- [28] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard, "Survey of sparse and non-sparse methods in source separation," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 18–33, 2005.
- [29] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal of Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

- [30] E. Paschen and R. P. Mosby, Eds., *Poetry Speaks: Hear Great Poets Read Their Work from Tennyson to Plath*, Sourcebooks, Naperville, Ill, USA , 2001.
- [31] C. Févotte, R. Gribonval, and E. Vincent, “BSS-EVAL toolbox user guide,” Tech. Rep. 1706, IRISA, Rennes, France, 2005.
- [32] N. Mitianoudis and T. Stathaki, “Overcomplete source separation using Laplacian mixture models,” *IEEE Signal Processing Letters*, vol. 12, no. 4, pp. 277–280, 2005.
- [33] Ö. Yilmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking,” *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
- [34] T. Melia and S. Rickard, “Underdetermined blind source separation in echoic environments using DESPRIT,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 86484, 19 pages, 2007.
- [35] S. Araki, H. Sawada, R. Mukai, and S. Makino, “Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors,” *Signal Processing*, vol. 87, no. 8, pp. 1833–1847, 2007.
- [36] P. Bofill, “Underdetermined blind separation of delayed sound sources in the frequency domain,” Tech. Rep. UPC-DAC-2001-14, Universitat Politecnica de Catalunya, Barcelona, Spain, 2002.