

Underdetermined Audio Source Separation using Laplacian Mixture Modelling

Nikolaos Mitianoudis

Abstract The problem of underdetermined audio source separation has been explored in the literature for many years. The instantaneous K -sensors, L -sources mixing scenario (where $K < L$) has been tackled by many different approaches, provided the sources remain quite distinct in the virtual positioning space spanned by the sensors. In this case, the source separation problem can be solved as a directional clustering problem along the source position angles in the mixture. The use of Laplacian Mixture Models in order to cluster and thus separate sparse sources in underdetermined mixtures will be explained in detail in this chapter. The novel Generalised Directional Laplacian Density will be derived in order to address the problem of modelling multidimensional angular data. The developed scheme demonstrates robust separation performance along with low processing time.

1 Introduction

Let a set of K sensors $\mathbf{x}(n) = [x_1(n), \dots, x_K(n)]^T$ observe a set of L sound sources $\mathbf{s}(n) = [s_1(n), \dots, s_L(n)]^T$. We will consider the case of instantaneous mixing, i.e. each sensor captures a scaled version of each signal with no delay in transmission. Moreover, the possible additive noise will be considered negligible. The above instantaneous mixing model can be expressed in mathematical terms, as follows:

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) \tag{1}$$

where \mathbf{A} represents the $K \times L$ *mixing matrix* and n the sample index. The blind source separation problem provides an estimate of the source signals $\mathbf{s}(n)$ given the sensor signals $\mathbf{x}(n)$. Usually, most separation approaches are *semi-blind*, which

Nikolaos Mitianoudis

Image Processing and Multimedia Lab, Electrical and Computer Engineering Department, Democritus University of Thrace, 67100 Xanthi, Greece. e-mail: nmitiano@ee.duth.gr

implies some knowledge of the source signal's general statistical structure. A number of algorithms have been proposed to solve the overdetermined and complete source separation problem ($K \geq L$) with great success. The additional assumption of statistical independence between the sources led to a group of source separation algorithms, summarised under the general term *Independent Component Analysis* (ICA). Starting from different interpretations of statistical independence, most algorithms perform source separation with great accuracy. An overview of current ICA and general blind source separation algorithms can be found in tutorial books on ICA by Hyvärinen et al. [31], Cichocki-Amari [14] and Common et al. [15].

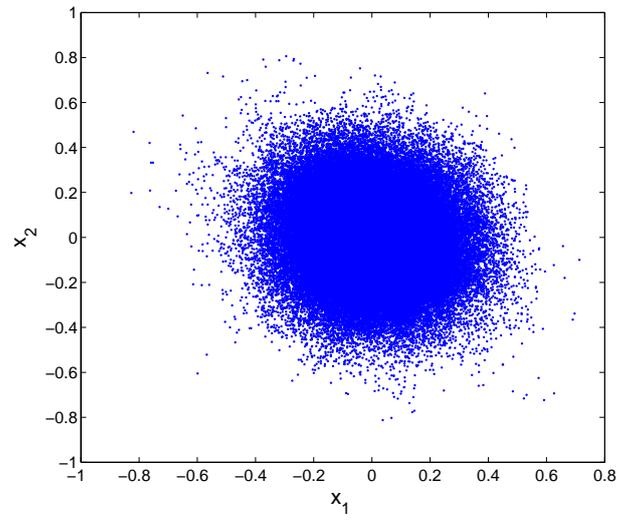
The underdetermined source separation problem ($K \leq L$) is more challenging, since in this case, the estimation of the mixing matrix \mathbf{A} is not sufficient for the estimation of the source signals $\mathbf{s}(n)$. This type of mixtures can be encountered in musical audio mixes. A number of solo instrument recordings are combined linearly in a stereo ($K = 2$) or a multichannel ($K = 5$ or $K = 7$) mixture, in order to form a musical score recording. Assuming Gaussian distributions for the sources and a known mixing matrix \mathbf{A} , one could estimate the sources using the pseudo-inverse of matrix \mathbf{A} in a Maximum Likelihood sense [38]. As most speech and audio signals tend to follow heavy-tailed “nonGaussian” distributions, the above linear operation is not sufficient to estimate the sources. Therefore, the underdetermined source separation problem can be divided into two sub-problems: i) estimating the mixing matrix \mathbf{A} and ii) estimating the source signals $\mathbf{s}(n)$.

The existence of a unique source estimate for the underdetermined source separation problem, even in the case that \mathbf{A} is known, is always under question, since it is an ill-conditioned problem that has an infinite number of solutions. Any linear system with less equations than unknown variables has an infinite number of solutions (source estimates) [34]. However, according to Eriksson and Koivunen [24], the linear generative model of (1) can have a *unique* and *identifiable* solution for the underdetermined case, provided i) there are no Gaussian sources present in the mixture, ii) the mixing matrix \mathbf{A} is of full row rank, i.e. $\text{rank}(\mathbf{A}) = K$ and iii) none of the source variables has a characteristic function featuring a component in the form $\exp(Q(u))$, where $Q(u)$ is a second-order polynomial or higher. This implies that this intractable problem may have a non-infinite number of solutions, under several constraints and probabilistic criteria for the sources.

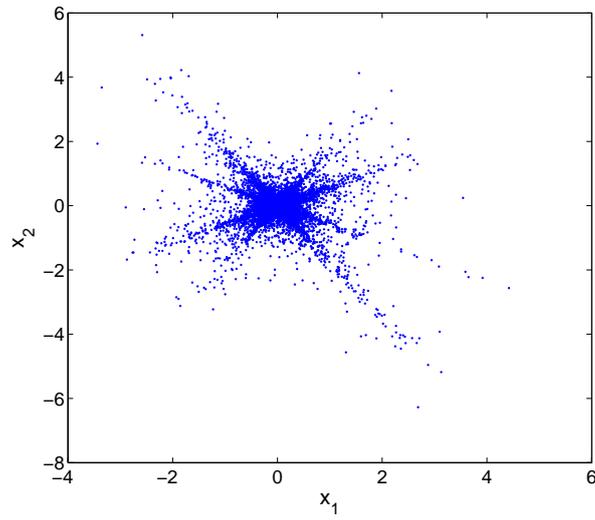
One probabilistic profile that satisfies the assumptions set above are *sparse* distributions. *Sparsity* is mainly used to describe signals that are mostly close to a mean value with the exception of several large values. Common models that can be used for approximating sparsity are minimum \mathcal{L}_0 or \mathcal{L}_1 norms [38], *Mixture of Gaussians* (MoG) [18, 46, 6] or *factorable Laplacian distributions* [28]. The separation quality for the underdetermined case seems to improve with sparsity, as usually the performance of source separation algorithms is closely connected with the “non-Gaussianity” of the source signals [12]. However, in many practical applications, the source data are not sparse. For example, some musical instrument signals tend to be less sparse than speech signals in the time-domain. Speech contains a lot of silent segments that guarantee sparsity (many zero samples), however, this might not be the case with many instrument signals. Therefore, the assumed sparse mod-

els are not accurate enough to describe the statistical properties of the signals in the time-domain. Many natural signals can have sparser representations in other transform domains, including the Fourier transform, the Wavelet transform and the Modified Discrete Cosine Transform (MDCT). Since these transformations are linear, it is equivalent to estimate the generative model and the sources in the transform domain. There are also alternative methods, where one can generate sparse representations for a specific dataset [17]. In the following analysis, the MDCT is employed to provide a sparser representation of audio signals.

The underdetermined source separation problem has been covered extensively in the literature. Lewicki [38] provided a complete Bayesian approach, assuming Laplacian source priors to estimate both the mixing matrix and the sources in the time domain. In [36], Lee et al. applied the previous algorithm to the source separation problem. Girolami [28] employed the factorable Laplacian distribution and variational EM to estimate the mixing matrix and the sources. More complete sparse source models, such as the *Student-t* distribution, were employed by Févotte et al. [25]. The parameters of the model, the mixing matrix and the source signals were estimated using either *Markov Chains Monte Carlo* (MCMC) simulations [25] or a *Variational Expectation Maximisation* (EM) algorithm [13], featuring robust performance, however, being computationally expensive. Clustering solutions were introduced by Hyvärinen [30] and Zibulevsky et al. [62], also featuring good results and lower computational complexity. In this case, the mixing matrix and the source signals are estimated by performing clustering in a sparser representation of the signals in the transform domain. Bofill-Zibulevsky [11] presented a shortest path algorithm based on \mathcal{L}_1 minimisation that could estimate the mixing matrix and the sources. O’Grady and Pearlmutter [48] proposed an algorithm to perform separation via Oriented Lines Separation (LOST) using clustering along lines (Hard-Lost) in a similar manner to Hyvärinen [30]. In addition, they proposed a soft-thresholding technique using an EM on a mixture of oriented lines to assign points to more than one source [47]. Davies and Mitianoudis [18] employed two-state Gaussian Mixture Models (GMM) to model the source densities in a sparse representation and also the additive noise. An EM-type algorithm was used to estimate the parameters of the two-state models and perform source coefficients clustering. The latter approach can be considered a joint Bayesian and clustering approach. A two-sensor more-sources setup, modelling also some delays between the sensors, was addressed using the DUET algorithm [61] that can separate the sources, by calculating amplitude differences (AD) and phase differences (PD) between the sensors. An online version of the algorithm was also proposed [51]. Recently, Arberet et al [5] proposed a method to count and locate sources in underdetermined mixtures. Their approach is based on the hypothesis that in localised neighbourhoods around some time-frequency points (t, f) (in the Short-Time Fourier Transform (STFT) representation) only one source essentially contributes to the mixture. Thus, they estimate the most dominant source (the Estimated Steering Vector) and a local confidence Measure which increases where a single component is only present. A clustering approach merges the above information and estimates the mixing matrix \mathbf{A} . In [58], Vincent et al used local Gaussian Modelling of minimal constrained variance of the



(a) Time domain



(b) MDCT domain

Fig. 1 Scatter plot of a two-sensor four-sources mixture in the time domain and in the sparse MDCT domain. The almost Gaussian-like structure of the time-domain representation is enhanced using the MDCT and the four sources can be clearly identified in the mixture.

local time-frequency neighbours assuming knowledge of the mixing matrix \mathbf{A} . The candidate sources' variances are estimated after minimising the Kullback-Leibler (KL) divergence between the empirical and expected mixture covariances, assuming that at most 3 sources contribute to each time-frequency neighbourhood and the sources are derived using Wiener filtering.

The instantaneous mixtures model is rather incomplete in the case of sources recorded in a real acoustic room environment. Assume the case of a sound source and a microphone in a room. Previous research has shown that the signal captured by the microphone can be well represented by a *convolution* of the source signal with a high-order FIR filter, modelling the room acoustics between the source and the sensor [42]. In the case of many sources and sensors, the signal at each sensor can be modelled by the following equation:

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{h}_{11} & \dots & \mathbf{h}_{1L} \\ \dots & \dots & \dots \\ \mathbf{h}_{K1} & \dots & \mathbf{h}_{KL} \end{bmatrix} * \mathbf{s}(n) \quad (2)$$

where $*$ denotes the linear convolution operator and \mathbf{h}_{ij} denotes an FIR filter modelling the room impulse response between the i -th microphone and the j -th source.

Many methods have been proposed to solve the square ($K = L$) convolutional ICA problem. Some of them suggested working directly in the time-domain [35, 57]. Working in the time domain has the disadvantage of being rather computationally expensive, due to calculating many convolutions and the size of the unmixing filters. Other approaches suggested moving to the STFT domain in order to transform the convolution into multiplication and apply ICA methods for instantaneous mixtures (i.e. the natural gradient) for each frequency bin [55]. However, there is an inherent *permutation problem* in all FD-ICA methods, which does not exist in time-domain methods. Mitianoudis and Davies [42] proposed a *time-frequency source model* for a ML-ICA approach, incorporating a *time-varying* parameter, aiming to impose *frequency coupling* between neighboring frequency bins. In addition, a *likelihood ratio* test was proposed to address the permutation problem. In [43], Mitianoudis and Davies described a mechanism to align permutations using subspace methods at each frequency bin. This idea was refined and was extended for underdetermined convolutive mixtures by Sawada et al [52, 4, 53]. Winter et al [60] estimate the mixing matrix based on hierarchical clustering, assuming sparsity of the source signals. Sources are then estimated using L1-norm minimization of complex numbers, using Laplacian source priors. Duong et al [23] model the contribution of each source to all mixture channels in the time-frequency domain as a zero-mean Gaussian random variable (r.v.) whose covariance encodes the spatial characteristics of the source. They derive a family of iterative EM algorithms to estimate the parameters of each model and propose suitable procedures adapted from previous convolutive approaches to align the order of the estimated sources across all frequency bins.

A more general source separation case can be introduced by using the following non-linear mixing setup:

$$\mathbf{x} = f(\mathbf{s}) \quad (3)$$

where $f(\cdot)$ is a general non-linear function, which provides a mapping $f: \mathcal{R}^L \rightarrow \mathcal{R}^L$. The solution for this problem forms a new class of source separation algorithms, termed *non-linear BSS*. The non-linear problem has a fundamental characteristic that solutions always exist; however, they are highly non-unique [31]. The general nonlinear BSS problem can be addressed using Kohonen Self-Organising maps [49]. In [33], Jutten and Karhunen state that one can reduce these great indeterminacies by constraining the mapping $f(\cdot)$ to a certain set of transformations. *Smooth non-linear mappings*, i.e. mappings that preserve independence of the components, such as a rotation matrix, can be unmixed using multi-layer perceptron (MLP) networks. In the *Post non-linear* (PNL) model, the nonlinear mapping has the following structure:

$$x_i(n) = f_i\left(\sum_{j=1}^L a_{ij}s_j(n)\right), \quad i = 1, \dots, K \quad (4)$$

where the nonlinear functions $f_i(\cdot)$ are assumed to be invertible. Such models may appear in array processing, satellite and microwave communications. A separation method for PNL models generally consists of two stages [31]: a) a nonlinear stage, where the functions f_i are inverted, b) a linear stage, where the linearised mixture is separated using an ordinary linear instantaneous ICA algorithm. In addition, there are other special nonlinear mixing cases, that can be linearised using another nonlinear mapping function $g(\cdot)$ (see [33]). Recently, Duarte et al [22] introduced a blind compensation scheme of the nonlinear distortion introduced in PNL mixtures, by using a semi-blind cost function to estimate the parameters of a known inverting function. Nevertheless, further exploration of nonlinear mixtures separation goes beyond the scope of this chapter.

In this chapter, instantaneous underdetermined source separation is examined in the form of a directional clustering problem. Clustering is performed with the application of density mixture models, which are trained on the directional data using the EM algorithm. We examine three cases of candidate densities based on the Laplacian distribution, which is well-suited to model sparse data. The directionality of the source separation data led to the introduction of a wrapped-Laplacian density and finally a generalised directional Laplacian density, a closed-form expression that can model multidimensional directional data.

2 Underdetermined Source Separation as a directional clustering problem

Let us assume a two-sensor instantaneous mixing approach. In Fig. 1(a), one can see the scatter plot of the two sensor signals, in the case of two sensors and four sources. The four sources are 7 sec of speech, accordion, piano and violin signals. In the time-domain representation, no directions of the input sources are visible

in the mixture. Consequently, the separation problem seems very difficult to solve. To get a sparser representation of the data, the MDCT or the *Short-Time Fourier Transform* (STFT) can be applied on the observed signals. The MDCT is a linear, real transform that has excellent sparsifying properties for most audio and speech signals. The harmonic content of most speech and musical instrument signals can be represented by harmonically related sinusoids with great accuracy (excluding transient and percussive parts in audio and unvoiced segments in speech). Consequently, using a transformation that projects the audio data on sinusoidal bases will most probably result into a more compact and sparse representation of the original data. The MDCT is more preferable than the STFT, since it is real and retains all the required sinusoidal signal structure. The need for sparser representations in underdetermined source separation and audio analysis in general is discussed more rigorously in [29, 50, 17, 37, 10]. When the sources are sparse, smaller coefficients are more probable, whereas all the signal's energy is concentrated in few large values. Therefore, the density of the data in the mixture space shows a tendency to cluster along the directions of the mixing matrix columns [62]. Observing the scatter plot in Fig. 1(b), it is clear that the angular difference between the two sensors can be used to identify and separate the sources in the mixture. That is to say, the two-dimensional (2D) problem can be transformed to a one-dimensional (1D) problem, as the main important parameter is the angle θ_n of each point.

$$\theta_n = \text{atan} \frac{x_2(n)}{x_1(n)} \quad (5)$$

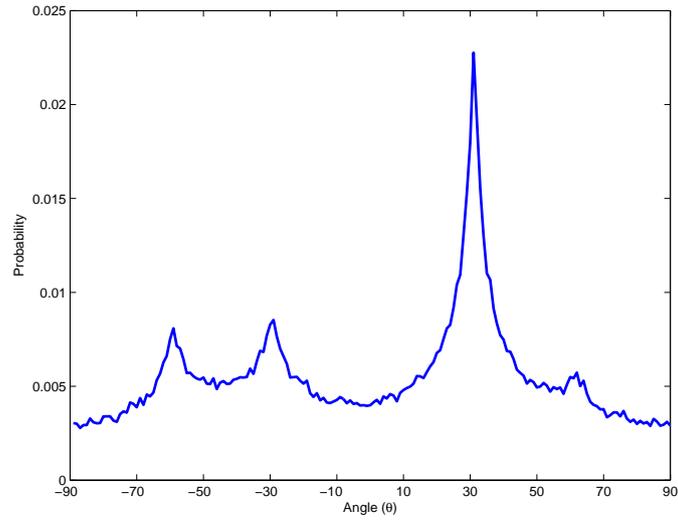
Using the directional differences between the two sensors is equivalent to mapping all the observed data points on the unit-circle. Extending this to a general multi-channel scenario, one can map K -dimensional points $\mathbf{x}(n)$ to the K -D unit sphere, by dividing with the vector's norm $\|\mathbf{x}\|$. In Fig. 2(a), we plot the histogram of the observed data angle θ_n in the previous example.¹ The strong ‘‘superGaussian’’ characteristics of the individual components in the MDCT domain are preserved in the angle representation θ_n . Then, the vectors $\mathbf{x}_{norm}(n)$ contain only directional information in a polar reference system.

$$\mathbf{x}_{norm}(n) = \frac{\mathbf{x}(n)}{\|\mathbf{x}(n)\|} \quad (6)$$

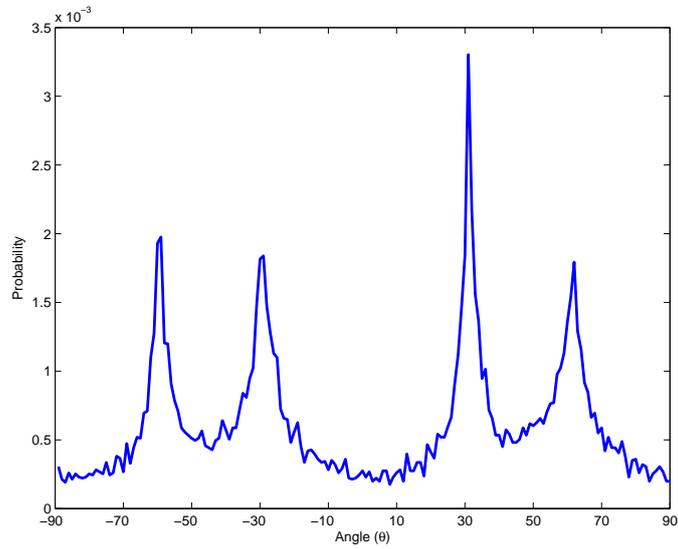
We can also define the magnitude r_n of each point $\mathbf{x}(n)$, as follows:

$$r_n = \|\mathbf{x}(n)\| = \sqrt{x_1(n)^2 + \dots + x_K(n)^2} \quad (7)$$

¹ A π -periodicity is valid for the observed phenomenon, since data in $(\pi/2, 3\pi/2)$ are symmetrical to the ones in $(-\pi/2, \pi/2)$ (See Fig. 1(b)). Hence, the use of the atan function instead of the extended atan2 function is justified. For the rest of the analysis, we will assume that θ_n takes values between $(0, \pi)$ rather than $(-\pi/2, \pi/2)$. This implies that data in the 4th quadrant $(-\pi/2, 0)$ are mapped with odd symmetry to the 2nd quadrant $(0, \pi/2)$. This is performed in order to facilitate the derivations of the Generalised Directional Laplacian Distribution and does not alter anything in the actual data.



(a) Original histogram



(b) Modified histogram

Fig. 2 Histograms of angle θ_n in the four sources example of Fig. 1. The four sources are identifiable in the original histogram (a), however, keeping only the most “superGaussian” components (b), we can facilitate the separation process, as the directions of arrival are more clearly identifiable.

We can observe that points that are close to the origin have a more Gaussian structure and thus do not contribute to the desired “superGaussian” profile. Consequently, we can use a “reduced” representation of the original data in order to estimate the columns of the mixing matrix more accurately. In Fig. 2(b), we can see a histogram of those points n , whose magnitude r_n is above a threshold, e.g. $r_n > 0.1$. Comparing with the original histogram of Fig. 2(a), the four components are more clearly identifiable in this reduced representation, which can facilitate the estimation of the columns of the mixing matrix, i.e. the directions of arrival for each source. In this representation, we will present three models based on the Laplacian density, that can be applied to cluster and separate the sources.

3 Identification using Laplacian Mixtures Models

A model, that is commonly used in the literature to model sparse data, is the *Laplacian* density function. The definition for the Laplacian probability density function (pdf) is given by the following expression:

$$\mathcal{L}(\theta, k, m) = ke^{-2k|\theta - m|} \quad (8)$$

where m defines the mean and $k > 0$ controls the “width” (approximate standard deviation) of the distribution. In a similar fashion to Mixtures of Gaussians (MoG), one can employ *Laplacian Mixture Models* (LMM) in order to model a mixture of “heavy-tailed signals”. A *Laplacian Mixture Model* of K Laplacians can thus be defined, as follows:

$$p(\theta) = \sum_{i=1}^K \alpha_i \mathcal{L}(\theta, k_i, m_i) = \sum_{i=1}^K \alpha_i k_i e^{-2k_i|\theta - m_i|} \quad (9)$$

where α_i , m_i , k_i represent the weight, mean and width of each Laplacian respectively and all weights should sum up to one, i.e. $\sum_{i=1}^K \alpha_i = 1$. The EM algorithm is employed to train the parameters of the mixture model. A complete derivation of an EM algorithm was presented by Dempster et al. [19] and has been employed to fit a MoG on a training data set [9]. Assuming N training samples for an 1D r.v. θ_n and Laplacian Mixture densities (9), the log-likelihood of these training samples takes the following form:

$$I(\alpha_i, k_i, m_i) = \sum_{n=1}^N \log \sum_{i=1}^K \alpha_i \mathcal{L}(\theta_n, k_i, m_i) \quad (10)$$

Introducing unobserved data items that can identify the components that “generated” each data item, we can simplify the log-likelihood of (10) for Laplacian Mixtures, as follows:

$$J(\alpha_i, k_i, m_i) = \sum_{n=1}^N \sum_{i=1}^K (\log \alpha_i + \log k_i - 2k_i |\theta_n - m_i|) p(i|\theta_n) \quad (11)$$

where $p(i|\theta_n)$ represents the probability of sample θ_n belonging to the i^{th} Laplacian of the LMM. In a similar fashion to MoGs, the updates for $p(i|\theta_n)$ and α_i can be given by the following equations:

$$p(i|\theta_n) = \frac{\alpha_i \mathcal{L}(\theta_n, m_i, k_i)}{\sum_{i=1}^K \alpha_i \mathcal{L}(\theta_n, m_i, k_i)} \quad (12)$$

$$\alpha_i^+ \leftarrow \frac{1}{N} \sum_{n=1}^N p(i|\theta_n) \quad (13)$$

The updates for m_i and k_i are estimated by setting $\partial J(\alpha_i, k_i, m_i) / \partial m_i = 0$ and $\partial J(\alpha_i, k_i, m_i) / \partial k_i = 0$ respectively. Following some derivation (see [44]), we get the following update rules:

$$m_i^+ \leftarrow \frac{\sum_{n=1}^N \frac{\theta_n}{|\theta_n - m_i|} p(i|\theta_n)}{\sum_{n=1}^N \frac{1}{|\theta_n - m_i|} p(i|\theta_n)} \quad (14)$$

$$k_i^+ \leftarrow \frac{\sum_{n=1}^N p(i|\theta_n)}{2 \sum_{n=1}^N |\theta_n - m_i| p(i|\theta_n)} \quad (15)$$

The four update rules are iterated until convergence. Enhancing the sparsity in the angle representation θ_n will increase EM's convergence speed and will provide more accurate estimates for the sources' angles. Therefore, we train the LMM with a subset of those data points n that satisfy $r_n > B$, where B is a threshold.

Once the LMM is trained, the centre of each Laplacian m_i should represent a column of the mixing matrix \mathbf{A} in the form of $[\cos(m_i) \sin(m_i)]^T$. Each wrapped Laplacian should model the statistics of each source in the transform domain and can be used to perform underdetermined source separation.

The main issue with LMMs is that they attempt to model a circular r.v. (angles) using a pdf that has infinite support. The Laplacian density, as described in (8), is valid $\forall \theta \in (-\infty, +\infty)$. However, the range of θ_n is not only bounded to the $(0, \pi)$ interval but the two boundaries are actually connected. Assume that you have a concentration of points close to π . The EM algorithm will attempt to fit a Laplacian around this cluster, however, assuming a linear support on θ . As a result, the algorithm can not attribute points that belong to the same cluster, but are close to 0, due to the assumed linear support. Therefore, the algorithm can not model densities with m_i close to 0 or π with great accuracy. To alleviate the problem, the estimated centres m_i can be rotated, so that the affected boundary (0 or π) is mapped to the middle of the centres m_i that feature the greatest distance (see [44]). This can offer a heuristic but not complete solution to the problem.

4 Identification using Mixtures of Wrapped Laplacian Models

To address this problem in a more elegant manner, we examine the use of an approximate wrapped-Laplacian distribution to model the π periodicity that exists in $\text{atan}(\cdot)$. The observed angles θ_n of the input data can be modelled, as a Laplacian wrapped around the interval $(0, \pi)$.

Definition 1. A wrapped-Laplacian can be described by the following additive model

$$\mathcal{L}_w(\theta, k, m) = \frac{1}{2T-1} \sum_{t=-T}^T k e^{-2k|\theta-m-\pi t|} = \frac{1}{2T-1} \sum_{t=-T}^T \mathcal{L}(\theta - \pi t, k, m) \quad (16)$$

where $T \in \mathbf{Z}^+$ denotes the number of ordinary Laplacians with mean m and width k that participate in the wrapped version.

The above expression models the wrapped Laplacian by an ordinary Laplacian and its periodic repetitions by π (see Fig. 3). This is an extension of the wrapped Gaussian distribution proposed by Smaragdis and Boufounos [56] for the Laplacian case. The addition of the wrapping of the distribution aims at mirroring the wrapping of the observed angles at $\pm\pi$. In theory, the model should have $T \rightarrow \infty$ components, however, it seems that a small range of values for T can successfully approximate the full wrapped probability density function in practice. In a similar fashion to LMMs,

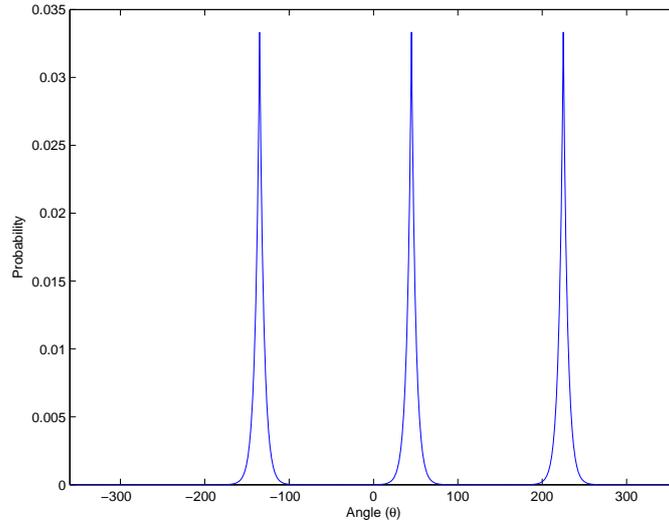


Fig. 3 An example of the Wrapped Laplacian for $T = [-1, 0, 1]$ $k = 0.01$ and $m = \pi/4$.

one can introduce *Mixture of wrapped Laplacians* (MoWL) in order to model a

mixture of angular or circular sparse signals. A *Mixture of wrapped Laplacians* can thus be defined, as follows:

$$p(\theta) = \sum_{i=1}^K \alpha_i \mathcal{L}_w(\theta, k_i, m_i) = \sum_{i=1}^K \alpha_i \frac{1}{2T-1} \sum_{t=-T}^T k_i e^{-2k_i |\theta - m_i - \pi t|} \quad (17)$$

where α_i , m_i , k_i represent the weight, mean and width of each Laplacian respectively and all weights should sum up to one, i.e. $\sum_{i=1}^K \alpha_i = 1$. We can derive the EM algorithm, based on the previous analysis. Assuming N training samples for θ_n and a Mixture of wrapped Laplacians densities (17), the log-likelihood of these training samples θ_n takes the following form:

$$I(\alpha_i, k_i, m_i) = \sum_{n=1}^N \log \sum_{i=1}^K \alpha_i \mathcal{L}_w(\theta_n, k_i, m_i) \quad (18)$$

One can introduce the probability $p(i|\theta_n)$ of sample θ_n belonging to the i^{th} wrapped Laplacian of the MoWL and the probability $p(t|i, \theta_n)$ of sample θ_n belonging to the t^{th} individual Laplacian of the i^{th} wrapped Laplacian $\mathcal{L}_w(k_i, m_i)$. The updates for $p(t|i, \theta_n)$, $p(i|\theta_n)$ and α_i can be then given by the following equations:

$$p(t|i, \theta_n) = \frac{\mathcal{L}(\theta_n - \pi t, m_i, k_i)}{\sum_{t=-T}^T \mathcal{L}(\theta_n - \pi t, m_i, k_i)} \quad (19)$$

$$p(i|\theta_n) = \frac{\alpha_i \mathcal{L}_w(\theta_n, m_i, k_i)}{\sum_{i=1}^K \alpha_i \mathcal{L}_w(\theta_n, m_i, k_i)} \quad (20)$$

$$\alpha_i \leftarrow \frac{1}{N} \sum_{n=1}^N p(i|\theta_n) \quad (21)$$

$$m_i \leftarrow \frac{\sum_{n=1}^N \sum_{t=-T}^T \frac{\theta_n - \pi t}{|\theta_n - \pi t - m_i|} p(t|i, \theta_n) p(i|\theta_n)}{\sum_{n=1}^K \sum_{t=-T}^T \frac{1}{|\theta_n - \pi t - m_i|} p(t|i, \theta_n) p(i|\theta_n)} \quad (22)$$

$$k_i \leftarrow \frac{\sum_{n=1}^N p(i|\theta_n)}{2 \sum_{n=1}^N \sum_{t=-T}^T |\theta_n - \pi t - m_i| p(t|i, \theta_n) p(i|\theta_n)} \quad (23)$$

Once the MoWL is trained, the centre of each wrapped Laplacian m_i should represent a column of the mixing matrix \mathbf{A} in the form of $[\cos(m_i) \sin(m_i)]^T$. Each wrapped Laplacian should model the statistics of each source in the transform domain and can be used to perform underdetermined source separation. This approach addresses the problem of modelling directional data in a more elegant manner, however, the cost of training two EM algorithms makes this approach less attractive.

5 A complete solution using the Generalised Directional Laplacian Distribution

The previous two efforts do not offer a closed form solution to the problem and they can not be easily expanded to more than two sensors. The proposed multidimensional Directional Laplacian model offers a closed form solution to the modelling of directional sparse data and can also address the general $K \times L$ underdetermined source separation problem, which is rarely tackled in the literature. There exist distributions that are periodic by definition and can therefore offer closed-form models for circular or directional data.

The *von Mises distribution* (also known as the circular normal distribution) is a continuous probability distribution on the unit circle [32, 27]. It may be considered the circular equivalent of the normal distribution and is defined by:

$$p(\theta) = \frac{e^{k \cos(\theta - m)}}{2\pi I_0(k)}, \forall \theta \in [0, 2\pi) \quad (24)$$

where $I_0(k)$ is the modified Bessel function of the first kind of order 0, m is the mean and $k > 0$ describes the “width” of the distribution. A generalisation of the previous density is the p -dimensional (p -D) von Mises-Fisher distribution [21, 40]. A p -D unit random vector \mathbf{x} ($\|\mathbf{x}\| = 1$) follows a *von Mises-Fisher* distribution, if its probability density function is described by:

$$p(\mathbf{x}) \propto e^{k\mathbf{m}^T \mathbf{x}}, \forall \|\mathbf{x}\| \in \mathcal{S}^{p-1} \quad (25)$$

where $\|\mathbf{m}\| = 1$ defines the centre, $k \geq 0$ and \mathcal{S}^{p-1} is the p -D unit hypersphere. Since the random vector \mathbf{x} resides on the surface of a p -D unit-sphere, \mathbf{x} essentially describes directional data. In the case of $p = 2$, \mathbf{x} models data that exist on the unit circle and thus can be described only by an angle. In this case, the von Mises-Fisher distribution is reduced to the von-Mises distribution of (24). The von Mises-Fisher distribution has been extensively studied and many methods have been proposed to fit the distribution or its mixtures to normally distributed circular data [8, 32, 40, 21].

5.1 A Generalised Directional Laplacian model

Assume a r.v. θ modelling directional data with π -periodicity. The periodicity of the density function can be amended to reflect a “fully circular” phenomenon (2π), however, for the rest of the paper we will assume that $\theta \in [0, \pi)$, since it is required by the source separation application. From the definition of the von-Mises distribution in (24), one can create a Laplacian structure simply by introducing a $|\cdot|$ operator in the superscript of the exponential. This action introduces a large concentration around the mean, which is needed to describe a sparse or Laplacian density. Values far away from the mean are smoothed out by the exponential. Additionally,

we have to perform some minor amendments to the phase shift and also invert the distribution in order to impose the desired shape on the derived density.

Definition 2. The following probability density function models directional Laplacian data over $[0, \pi]$ and is termed *Directional Laplacian Density* (DLD):

$$p(\theta) = c(k)e^{-k|\sin(\theta-m)|}, \forall \theta \in [0, \pi] \quad (26)$$

where $m \in [0, \pi]$ defines the mean, $k > 0$ defines the width (“approximate variance”) of the distribution, $c(k) = \frac{1}{\pi I_0(k)}$ and $I_0(k) = \frac{1}{\pi} \int_0^\pi e^{-k \sin \theta} d\theta$.

The normalisation coefficient $c(k) = 1/\pi I_0(k)$ is derived from the fundamental normalisation property of probability density functions [41]. Examples of (26) and more details on the special 1D DLD case can be found in [41]. The next

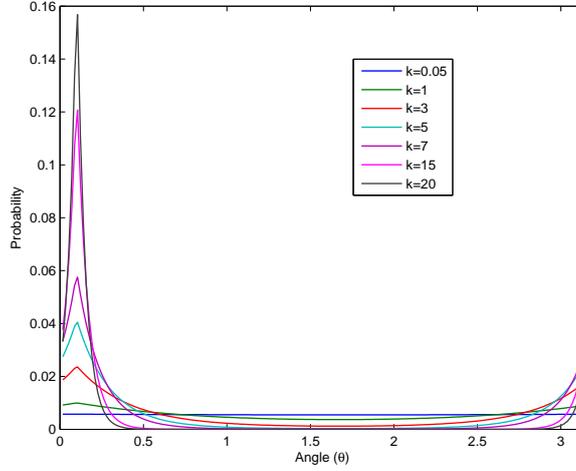


Fig. 4 The proposed Directional Laplacian Density (DLD) for various values of k .

step is to derive a generalised definition for the Directional Laplacian model. To generalise the concept of 1D DLD in the p -D space, we will be inspired by the p -D von Mises-Fisher distribution [21, 40]. The von Mises-Fisher distribution is described by $p(\mathbf{x}) \propto e^{k\mathbf{m}^T \mathbf{x}}$ (see (25)). Since $\|\mathbf{x}\| = \|\mathbf{m}\| = 1$, the inner product $\mathbf{m}^T \mathbf{x} = \cos \psi$, where ψ is the angle between the two vectors \mathbf{x} and \mathbf{m} (see Fig. 5). Following a similar methodology to the 1D-DLD, we need to formulate the term $-k|\sin \psi|$ in the superscript of the exponential. We can then derive $|\sin \psi| = \sqrt{1 - \cos^2 \psi} = \sqrt{1 - (\mathbf{m}^T \mathbf{x})^2}$. Thus, the superscript of the generalised DLD can be given by $-k\sqrt{1 - (\mathbf{m}^T \mathbf{x})^2}$.

Definition 3. The following probability density function models p -D directional Laplacian data and is termed *Generalised Directional Laplacian Distribution* (DLD):

$$p(\mathbf{x}) = c_p(k) e^{-k\sqrt{1-(\mathbf{m}^T \mathbf{x})^2}}, \forall \|\mathbf{x}\| \in \mathcal{S}^{p-1} \quad (27)$$

where \mathbf{m} defines the mean, $k \geq 0$ defines the width (“approximate variance”) of the distribution, $c_p(k) = \frac{\Gamma(\frac{p-1}{2})}{\pi^{\frac{p+1}{2}} I_{p-2}(k)}$, $I_p(k) = \frac{1}{\pi} \int_0^\pi e^{-k \sin \theta} \sin^p \theta d\theta$ and $\Gamma(\cdot)$ represents the Gamma function².

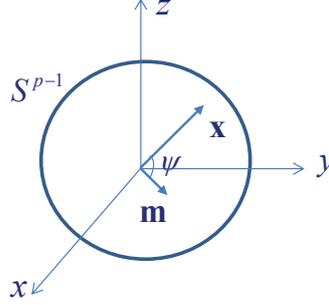


Fig. 5 Generalising the Directional Laplacian density in \mathcal{S}^{p-1} .

The normalisation coefficient $c_p(k)$ is calculated in Appendix 1. In the case of $p = 2$, the generalised DLD is reduced to the one dimensional DLD of (26), verifying the validity of the above model. The generalised DLD density models “directional” data on the half-unit p -D sphere, however, it can be extended to the unit p -D sphere, depending on the specifications of the application. In Fig. 6, an example of the generalised DLD is depicted for $p = 3$ and $k = 5$. The centre \mathbf{m} is calculated using spherical coordinates $\mathbf{m} = [\cos \theta_1 \cos \theta_2; \cos \theta_1 \sin \theta_2; \sin \theta_1]$ for $\theta_1 = 0.2$ and $\theta_2 = 2$.

5.2 Generalised Directional Laplacian Density samples generation

To generate 1D Directional Laplacian data, we employed the inversion of the cumulative distribution method [20]. Inversion methods are based on the observation that continuous cumulative distribution functions (cdf) range uniformly over the interval $(0, 1)$. Since the proposed density is bounded between $[0, \pi)$, we can evaluate the cdf of the Directional Laplacian density with uniform sampling at $[0, \pi)$ and approximate the inverse mapping using spline interpolation. Thus, uniform random data in the interval $(0, 1)$ can be transformed to 1D Directional Laplacian random samples, using the described inverse mapping procedure.

² Note that for n positive integer, we have that $\Gamma(n) = (n-1)!$

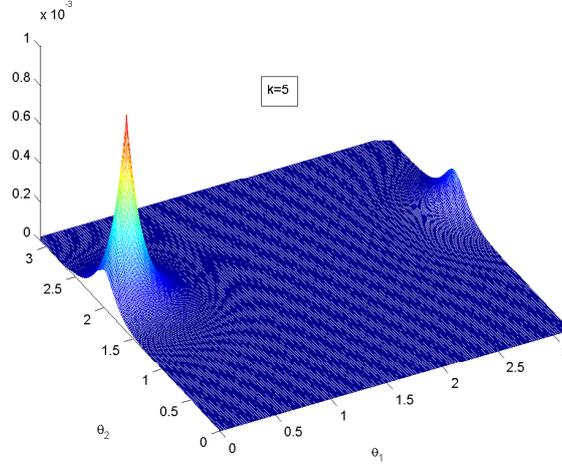


Fig. 6 The proposed Generalised Directional Laplacian Distribution for $k = 5$ and $p = 3$.

To simulate 2-D Directional Laplacian random data ($p = 3$), we sampled the 2-D density function for specific \mathbf{m} , k . The bounded value space ($\theta_1, \theta_2 \in [0, \pi)$) is quantised into small rectangular blocks, where the density is assumed to be uniform. Consequently, we generate a number of uniform random samples for each block. The number of samples generated from each block is different and defined by the overall DL density. The required 3-D unit-norm random vectors are produced using spherical coordinates with unit distance and angles θ_1, θ_2 from the random 2-D Directional data. The above procedure can be extended for the generation of p -D directional data.

5.3 Maximum Likelihood Estimation of parameters \mathbf{m} , k

Assume a population of p -D angular data $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n, \dots, \mathbf{x}_N\}$ that follow a p -D Directional Laplacian Distribution. To estimate the model parameters using Maximum Likelihood Estimation (MLE), one can form the log-likelihood and estimate the parameters \mathbf{m} , k that maximise it. For the Generalised DLD density, the log-likelihood function can be expressed, as follows:

$$J(\mathbf{X}, \mathbf{m}, k) = N \log \frac{\Gamma(\frac{p-1}{2})}{\pi^{\frac{p+1}{2}} I_{p-2}(k)} - k \sum_{n=1}^N \sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2} \quad (28)$$

Alternate optimisation is performed to estimate \mathbf{m} and k . The gradients of J along \mathbf{m} and k are calculated in Appendix 2. The update for \mathbf{m} is given by gradient ascent on the log-likelihood via:

$$\mathbf{m}^+ \leftarrow \mathbf{m} + \eta \sum_{n=1}^N \frac{\mathbf{m}^T \mathbf{x}_n}{\sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2}} \mathbf{x}_n \quad (29)$$

$$\mathbf{m}^+ \leftarrow \mathbf{m}^+ / \|\mathbf{m}^+\| \quad (30)$$

where η defines the gradient step size. Since the gradient step does not guarantee that the new update for \mathbf{m} will remain on the surface of \mathcal{S}^{p-1} , we normalise the new update to unit norm. To estimate k , a numerical solution to the equation $\partial J(\mathbf{X}, \mathbf{m}, k) / \partial k = 0$ is estimated. From the analysis in Appendix 2, we have that

$$\frac{I_{p-1}(k)}{I_{p-2}(k)} = \frac{1}{N} \sum_{n=1}^N \sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2} \quad (31)$$

To calculate k analytically from the ratio $I_{p-1}(k)/I_{p-2}(k)$ is not straightforward. However, after numerical evaluation, it can be demonstrated that the ratio $I_{p-1}(k)/I_{p-2}(k)$ is a smooth monotonic 1 – 1 function of k . In Fig. 7, the ratio $I_{p-1}(k)/I_{p-2}(k)$ is estimated for uniformly sampled values of $k \in [0.01, 30]$ and $p = 2, 3, 4, 5, 6$. Since this ratio is not dependent on data, one can create a look-up table for a variety of k values and use interpolation to estimate k from an arbitrary value of $I_{p-1}(k)/I_{p-2}(k)$. This look-up table solution is more efficient compared to possible iterative estimation approaches of k and generally accelerates the model's training.

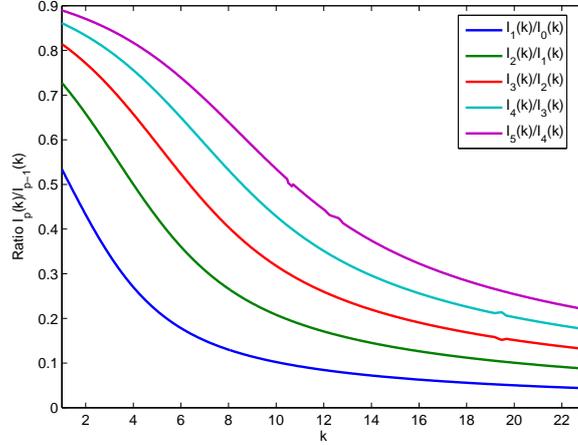


Fig. 7 The ratio $I_p(k)/I_{p-1}(k)$ is a monotonic 1 – 1 function of k .

5.4 Mixtures of Generalised Directional Laplacians

One can employ *Mixtures of Generalised Directional Laplacians* (MDLD) in order to model multiple concentrations of directional generalised “heavy-tailed signals”.

Definition 4. *Mixtures of Generalised Directional Laplacian Distributions* are defined by the following pdf:

$$p(\mathbf{x}) = \sum_{i=1}^K a_i c_p(k_i) e^{-k_i \sqrt{1 - (\mathbf{m}_i^T \mathbf{x})^2}}, \quad \forall \|\mathbf{x}\| \in \mathcal{S}^{p-1} \quad (32)$$

where a_i denotes the weight of each distribution in the mixture, K the number of DLDs used in the mixture and \mathbf{m}_i , k_i denote the mean and the “width” (approximate variance) of each distribution.

The mixtures of DLD can be trained using the EM algorithm. Following the previous analysis in [9], one can yield the following simplified likelihood function:

$$\mathcal{L}(a_i, \mathbf{m}_i, k_i) = \sum_{n=1}^N \sum_{i=1}^K \left(\log \frac{a_i \Gamma(\frac{p-1}{2})}{\pi^{\frac{p+1}{2}} I_{p-2}(k)} - k \sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2} \right) p(i|\mathbf{x}_n) \quad (33)$$

where $p(i|\mathbf{x}_n)$ represents the probability of sample \mathbf{x}_n belonging to the i^{th} Directional Laplacian of the mixture. In a similar fashion to other mixture model estimation, the updates for $p(i|\mathbf{x}_n)$ and α_i can be given by the following equations:

$$p(i|\mathbf{x}_n) \leftarrow \frac{a_i c_p(k_i) e^{-k_i \sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2}}}{\sum_{i=1}^K a_i c_p(k_i) e^{-k_i \sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2}}} \quad (34)$$

$$a_i \leftarrow \frac{1}{N} \sum_{n=1}^N p(i|\mathbf{x}_n) \quad (35)$$

Based on the derivatives calculated in Appendix 2, it is straightforward to derive the following updates for \mathbf{m}_i and k_i , as follows:

$$\mathbf{m}_i^+ \leftarrow \mathbf{m}_i + \eta \sum_{n=1}^N k_i \frac{\mathbf{m}_i^T \mathbf{x}_n}{\sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2}} \mathbf{x}_n p(i|\mathbf{x}_n) \quad (36)$$

$$\mathbf{m}_i^+ \leftarrow \mathbf{m}_i^+ / \|\mathbf{m}_i^+\| \quad (37)$$

To estimate k_i , in a similar fashion to the previous MLE, the optimisation yields:

$$\frac{I_{p-1}(k_i)}{I_{p-2}(k_i)} = \frac{\sum_{n=1}^N \sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2} p(i|\mathbf{x}_n)}{\sum_{n=1}^N p(i|\mathbf{x}_n)} \quad (38)$$

The training of this mixture model is also dependent on the initialisation of its parameters, especially the means \mathbf{m}_i [45]. In Appendix 3, the standard K-Means al-

gorithm is reformulated in order to tackle p -D directional data. The proposed p -D *Directional K-Means* is used to initialise the means \mathbf{m}_i of the DLDs in the generalised DLD mixture EM training. A *Directional K-Means* already exists in the literature [7], however, the proposed p -D *Directional K-Means* in Appendix 3 employs a distance function more relevant to sparse directional data.

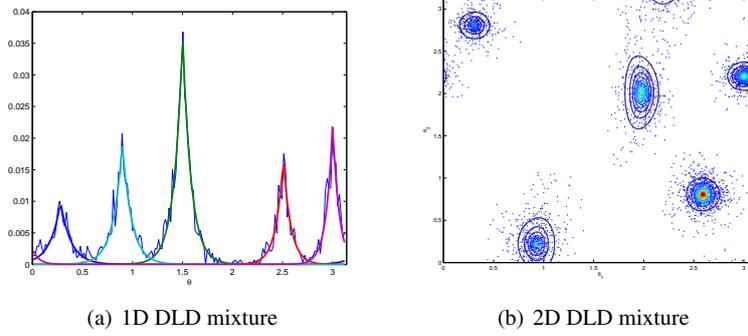


Fig. 8 Examples of fitting a Generalised MDLD model on 2000 randomly generated 1D (left) and 2D (right) Directional Laplacian data.

6 Source Separation using hard or soft thresholding

Once the Mixture Model is trained, optimal detection theory and the estimated individual Laplacians can be employed to provide estimates of the sources. The centre of each Laplacian m_i should represent a column of the mixing matrix \mathbf{A} in the form of $[\cos(m_i) \sin(m_i)]^T$. Each Laplacian should model the statistics of each source in the transform domain. Thus, using either a *hard* or a *soft decision threshold*, we can perform underdetermined source separation. The same strategy can hold for either of the three proposed Laplacian Mixture Models.

6.1 Hard thresholding

The hard thresholding (“Winner takes all”) strategy attributes each point of the scatter plot of Fig. 1(b) to only one of the sources. This is performed by setting a hard threshold at the intersections between the trained Laplacians. Consequently, the source separation problem becomes an *optimal decision* problem. The decision thresholds θ_{ij}^{opt} between the i -th and the j -th neighbouring Laplacians depend on the type of mixture model (LMM, MoWL or MDLD). Threshold formulas for the

LMM and MoWL can be found in [44, 45] respectively. Using these thresholds, the algorithm can attribute the points with $m_{ij}^{opt} < \theta_n < m_{jk}^{opt}$ to source j , where i, j, k are neighbouring Laplacians (sources). Fig. 9(a) depicts the fitted Laplacian Mixture Model, in a two sensors - four audio sources (voice, piano, accordion and violin) example and the hard thresholds imposed using the above equation. The points that belong to each of the four clusters, shown in Fig. 9(a), are attributed and are used to reconstruct each source.

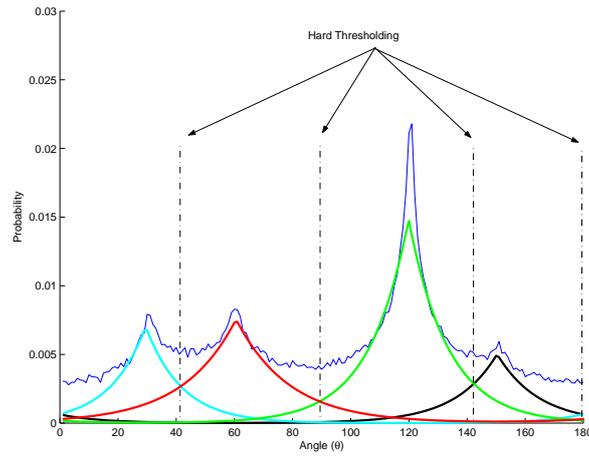
In the case of the 1D-MDLL, it is possible to derive the thresholds where two neighbouring DLDs intersect and therefore apply a hard thresholding strategy to cluster the audio data. In the case of an p -D MDLL, it is not straightforward to derive the intersecting hyperplanes between two neighbouring DLDs, therefore, in this case we resort to the soft-thresholding technique.

6.2 Soft thresholding

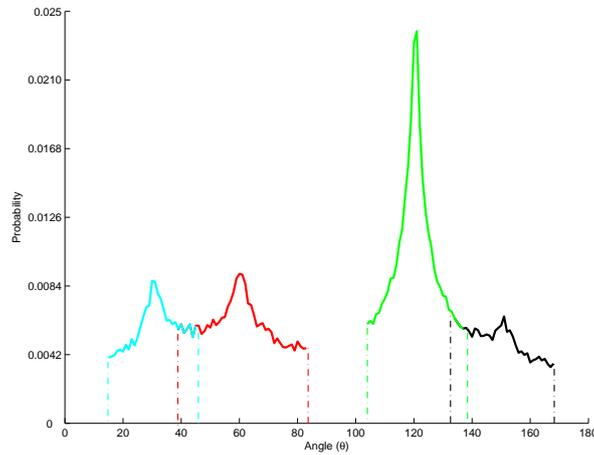
Observing the histograms of Fig. 2, we can attribute points that are distant from the centre of the 2D representation to each source with great confidence. In contrast, there exist points that can not be attributed to either source with great confidence. These points may belong to more than one source. One can then relax the hard threshold strategy, by allowing points belonging to more than one source simultaneously. A “soft-thresholding” strategy can attribute points that constitute a chosen ratio q (i.e. 0.7 – 0.9) of the density of each Laplacian (any of the three models) to the corresponding source (see Fig. 9). Hence, the i^{th} source can be associated with those points θ_n , for which $p(\theta_n) \geq (1 - q)\alpha_i k_i$, where $p(\theta_n)$ is given by the corresponding density model. A large value for q allows more points to belong to more than one Laplacian. A small value for q imposes stricter criteria for the points to belong to a Laplacian and essentially becomes a hard thresholding approach. This scheme can be effective, only if the estimated Laplacians are concentrated around each \mathbf{m}_i . In the opposite case, there will be components that will dominate the pdf and therefore be attributed with more points than it should and therefore they would contain contamination from other sources. In Fig. 9(b), we can see the four sources in the previous 1D example, as classified by the soft thresholding strategy. The different colours represent different clusters, i.e. different sources. We can see that several points are attributed to both the first and the second sources and both the third and fourth sources by the soft classification scheme.

6.3 Source Reconstruction

Having attributed the points $\mathbf{x}(n)$ to the L sources, using either the “hard” or the “soft” thresholding technique, the next step is to reconstruct the sources. Let $S_i \subseteq N$ represent the point indices that have been attributed to the i^{th} source and \mathbf{m}_i the



(a) Hard Thresholding



(b) Soft Thresholding

Fig. 9 A two-sensors four-sources scenario, separated using LMM. In (a), the four trained Laplacians are depicted along with the actual density function and the imposed hard thresholds. Applying soft thresholds, the classification shown in (b) is achieved, which allows some overlapping between adjacent sources.

corresponding mean vector, i.e. the corresponding column of the mixing matrix. We initialise $u_i(n) = 0, \forall n = 1, \dots, N$ and $i = 1, \dots, L$. The source reconstruction is performed by substituting:

$$u_i(S_i) = \mathbf{m}_i^T \mathbf{x}(S_i) \quad \forall i = 1, \dots, L \quad (39)$$

In the case that we need to capture the multichannel image of the separated source, the result of the separation is a multichannel output that is initialised to $\mathbf{u}_i(n) = \mathbf{0}, \forall n = 1, \dots, N$. The source image reconstruction is performed by:

$$\mathbf{u}_i(S_i) = \mathbf{x}(S_i) \quad \forall i = 1, \dots, L \quad (40)$$

7 Experiments

In this section, we verify the validity of the above derived MLE algorithms and demonstrate the density’s relevance and performance in underdetermined audio source separation. We can see that the proposed MDLD model improves the LMM and MoWL modelling efforts in terms of stability, speed and performance and offers a fast alternative to state-of-the-art algorithms with reasonable separation performance.

We will use Hyvärinen’s clustering approach [30], the MoWL algorithm [45] and the “GaussSep” algorithm [58] for comparison. We preferred not to benchmark the LMM model, because the other two models (MoWL and MDLD) tackle data’s directionality more efficiently. After fitting the MDLD with the proposed EM algorithm, separation will be performed using hard or soft thresholding, as described earlier. In order to quantify the performance of the algorithms, we estimate the *Signal-to-Distortion Ratio* (SDR), the *Signal-to-Interference Ratio* (SIR) and the *Signal-to-Artifact Ratio* from the BSS_EVAL Toolbox v.3 [26]. The input signals for the MDLD, MoWL and Hyvärinen’s approaches are sparsified using the MDCT, as developed by Daudet and Sandler [16]. The frame length for the MDCT analysis is set to 32 msec for the speech signals and 128 msec for the music signals sampled at 16 KHz, and to 46.4 msec for the music signals at 44.1 KHz. We initialise the parameters of the MoWL and MDLD as follows: $\alpha_i = 1/K$ and $c_i = 0.001$, $T = [-1, 0, 1]$ (for MoWL only) and $k_i = 15$ (for the DLD only). The centres m_i were initialised in either case using the Directional *K-means* step, as described in Appendix 3. We used the “GaussSep” algorithm, as publicly available by the authors³. For the estimation of the mixing matrix, we used Arberet et al’s [5] DEMIX algorithm⁴, as suggested in [58]. The number of sources in the mixture was also provided to the DEMIX algorithm, as it was provided to all other algorithms. The

³ MATLAB code for the “GaussSep” algorithm is available from <http://www.irisa.fr/metiss/members/evincent/software>.

⁴ MATLAB code for the “DEMIX” algorithm is available from <http://infoscience.epfl.ch/record/165878/files/>.

“GaussSep” algorithm operates in the STFT domain, where we used the same frame length with the other approaches and a time-frequency neighbourhood size of 5 for speech sources and 15 for music sources.

Table 1 The proposed MDLD approach is compared for source estimation performance ($K = 2$) in terms of SDR (dB), SIR (dB) and SAR(dB) with GaussSep, WMoL and Hyvärinen’s approach. The measurements are averaged for all sources of each experiment.

	SDR (dB)				SIR (dB)				SAR (dB)			
	MDLD	GaussSep	MoWL	Hyva	MDLD	GaussSep	MoWL	Hyva	MDLD	GaussSep	MoWL	Hyva
Latino1	6.38	5.51	5.72	0.89	18.63	8.96	18.59	9.61	6.93	9.20	6.26	3.63
Latino2	3.21	4.71	2.10	0.89	11.50	8.87	11.28	9.61	4.95	9.20	3.85	3.63
Groove	0.22	0.39	-0.43	-0.08	9.48	3.62	9.60	8.88	2.12	7.37	1.00	1.83
Dev2Male3	3.04	6.22	2.11	-3.10	13.69	12.14	13.30	4.73	4.10	8.04	3.33	2.72
Dev2Female3	4.68	5.70	3.86	-1.85	15.28	11.45	16.58	5.02	5.41	7.51	4.61	3.13
Dev2WDrums	9.59	16.57	10.16	0.63	19.77	23.83	19.98	7.57	10.55	17.68	10.54	5.54
Dev1WDrums	4.96	16.54	3.81	6.86	13.88	20.94	12.38	16.75	6.37	19.30	5.20	7.73
<i>Average</i>	4.58	7.96	3.91	0.6	14.61	12.83	13.82	8.88	5.78	11.19	4.97	4.03

7.1 Two-microphone examples

We tested the algorithms with the *Groove*, *Latino1* and *Latino2* datasets, available by BASS-dB [59], and sampled at 44.1 KHz. The “Groove” dataset features four widely spaced sources: bass (far left), distorted guitar (center left), clean guitar (center right) and drums (far right). The two “Latino” datasets features four widely spaced sources: bass (far left), drums (center left), keyboards (center right) and distorted guitar (far right). We also used a variety of test signals from the Signal Separation Evaluation Campaigns SiSEC2008 [1] and SiSEC2010 [2]. We employed two audio instantaneous mixtures from the “dev1” and “dev2” data sets (“Dev2WDrums” and “Dev1WDrums” sets - 3 instruments at 16KHz) and two speech instantaneous mixtures from the “dev2” data set (“Dev2Male3” and “Dev2Female3” sets - 4 closely located sources at 16 KHz). We used the development (dev) datasets instead of the test data sets, in order to have all the source audio files for proper benchmarking.

In Table 1, we can see the results for the four methods in terms of SDR, SIR and SAR. For simplicity, we averaged the results for all sources at each experiment. The reader of the paper can visit the following url⁵ and listen to the described separation results. The proposed MDLD approach seems to outperform our previous separation effort MoWL and Hyvärinen’s algorithm in terms of all the performance indexes.

⁵ <http://utopia.duth.gr/~nmitiano/mdld.htm>

The proposed MDLD approach is not susceptible to bordering effects, since it is circular by definition and avoids shortcomings of our previous offerings. Compared to a state-of-the-art method, such as “GaussSep”, our method is better in terms of the SIR index but is falling behind in terms of the SDR and SAR indexes. The SIR index reflects the capability of an algorithm to remove interference from other sources in the mixture. The SAR index refers to the audible artifacts that remain in the separated signals, due to the overlapping of several points in the time-frequency space (even in the MDCT representation) in the underdetermined mixture that are incorrectly attributed to either source. In this sense, our algorithm seems to perform slightly better compared to “GaussSep” in terms of removing “crosstalk” from other sources, but there seem to be more audible artifacts after separation in our approach compared to “GaussSep”. This is due to the fact that the “GaussSep” segments the time-frequency representation in small localised neighbourhoods and performs local Gaussian Modelling so as to separate and filter sources from those areas that separation is more achievable. Instead, our approach simply clusters all time-frequency points according to the fitted DLD using hard thresholds (or soft-thresholds in the case $K > 2$).

Table 2 Running time comparison with GaussSep and MoWL approaches. The measurements are in seconds.

	MDLD	Gaussep	MoWL
Groove	2.39	224.21	20.46
Latino1	1.27	122.02	5.48
Latino2	1.28	129.09	3.59
Dev2Male3	2.31	72.64	19.67
Dev2Female3	2.33	75.92	16.09
Dev2WDrums	2.07	56.79	8.55
Dev1WDrums	1.55	54.06	11.88
<i>Average</i>	1.88	104.96	12.24
Dev3Female3	9.56	1021.31	-
Example(3×5)	4.04	1598.7	-
Example(4×8)	9.393	2359.1	-
<i>Average</i>	7.66	1659.70	-

Another important issue is to compare the processing time of the three best performing algorithms. All experiments were conducted on an Intel Core i5-460M (2.53 GHz) with 4GB DDR3 SDRAM running Windows Professional 64-bit and MATLAB R2012b. Our MATLAB implementations of the MDLD and MoWL algorithms were not optimised in terms of execution speed. In Table 2, the typical running time in seconds is summarised for each experiment and method. The first observation is that the MDLD approach is faster compared to the MoWL approach. As it was previously mentioned, employing a mixture of wrapped Laplacians to solve the “circularity” problem entails the running of two EM algorithms: one for the wrapped Laplacians and one for the mixture of wrapped Laplacians. This seems

to delay the convergence of the algorithm. Instead, MDLD requires the training of one EM algorithm for the mixture and it seems to converge faster compared to MoWL. The second observation is that there is an important difference between the processing time of the MDLD approach and the ‘‘GaussSep’’ algorithm. As previously mentioned, the ‘‘GaussSep’’ algorithm is more complicated in structure thus justifying its long running time. Nevertheless, the proposed MDLD approach offers a very fast underdetermined source separation alternative with high SIR performance that can be used in environments where processing time is important. The third observation is that the processing time for the ‘‘GaussSep’’ algorithm scales significantly with the duration of the signals and the number of sources, i.e. the ‘‘Groove’’, ‘‘Latino1’’, ‘‘Latino2’’ (44.1KHz - 4 sources) require more time than the Dev2Male3 and Dev2Female3 sets (16KHz - 4 sources) and the Dev2WDrums and Dev1WDrums sets (16KHz - 3 sources). Instead, MDLD’s running time seems to be closer to the average in most cases, maybe slightly deteriorating with the complexity of the source separation problem.

Table 3 The sources’ position angles that were used in the 3×5 and the 4×8 example.

		3 × 5							
		s_1	s_2	s_3	s_4	s_5			
θ_1		0°	-87°	-60°	0°	45°			
θ_2		85°	0°	-60°	0°	45°			
		4 × 8							
		s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8
θ_1		-75°	-30°	0°	50°	10°	80°	-45°	0°
θ_2		70°	30°	-20°	50°	-70°	0°	15°	-70°
θ_3		80°	20°	10°	-50°	0°	-10°	-25°	-35°

Table 4 The proposed MDLD approach is compared for source estimation performance ($K = 3, 4$) in terms of SDR (dB), SIR (dB) and SAR (dB) with the GaussSep approach. The measurements are averaged for all sources of each experiment.

	SDR (dB)		SIR (dB)		SAR (dB)	
	MDLD	GaussSep	MDLD	GaussSep	MDLD	GaussSep
Dev3Female3	6.02	16.93	23.84	22.43	6.17	18.40
Example 3×5	3.91	9.94	17.92	15.21	4.17	11.68
Example 4×8	2.24	-18.63	16.4	-17.58	2.52	9.39

7.2 Underdetermined source separation examples with more than two mixtures

In this section, we employ the described generalised DLD approach to perform separation of $3 \times L$ and $4 \times L$ mixtures. The 2-mixtures setup, that dominates the literature, may also arise from the fact that most audio recordings and CD masters are available as stereo recordings (2 channels is equivalent to 2 mixtures), where we need to separate the instruments that are present. Nowadays, the music industry is moving towards multichannel formats, including the 5.1 and the 7.1 surround sound formats, which implies more than 2 channels will be available for processing. In this section, we will attempt to perform separation of the Dev3Female3 set from SiSEC2011 [3] and a 3×5 (3 mixtures - 5 sources) and a 4×8 (4 mixtures - 8 sources) scenario using the male and female voices from Dev3. Our MDLD approach will be compared to the ‘‘GaussSep’’ algorithm that is able to work with multi-channel data. We used the same frame length and time-frequency neighbourhood sizes for both algorithms as previously. The MDLD was initialised as described in the previous section. After fitting the model, we employed the soft-thresholding scheme, as it was described in [44]. Since it was not straightforward to calculate the intersection surfaces between the individual p -D DLDs, we employed a soft-thresholding scheme, as described earlier, using a value of $q = 0.8$.

For the 3×5 example, we centred the 5 speech sources around the angles shown in Table 3 which were mixed using the following mixing matrix:

$$\mathbf{A}_{3 \times L} = \begin{bmatrix} \cos \theta_2 \cos \theta_1 \\ \cos \theta_2 \sin \theta_1 \\ \sin \theta_2 \end{bmatrix} \quad (41)$$

For the 4×8 example, we centred eight audio sources around the angles shown in Table 3 which were mixed using the mixing matrix:

$$\mathbf{A}_{4 \times L} = \begin{bmatrix} \cos \theta_3 \cos \theta_2 \cos \theta_1 \\ \cos \theta_3 \cos \theta_2 \sin \theta_1 \\ \cos \theta_3 \sin \theta_2 \\ \sin \theta_3 \end{bmatrix} \quad (42)$$

The separation results for the three experiments in terms of SDR, SIR and SAR can be summarised in Table 4. The reader can listen to the audio results from the following url (See Footnote 5). In the case of $K = 3$ mixtures, both algorithms managed to perform separation in either case. Similarly to the $K = 2$ case, ‘‘GaussSep’’ featured higher SDR and SAR performances, whereas the proposed MDLD algorithm featured higher SIR performance. The image is completely different in the case of $K = 4$ mixtures, where MDLD manages to separate all 8 sources in contrast to ‘‘GaussSep’’ that fails to perform separation. This might be due to fact that the sparsest ML solution in the optimisation of [58] is restricted to vectors with $K \leq 3$ entries, i.e. 3 sources present at each point. In contrast, the proposed MDLD al-

gorithm is designed to operate for any arbitrary number of sensors K , without any constraint.

In Table 2, we can see the processing times for the two algorithms for the three experiments. The MDLD processing time has increased slightly but still remains relatively fast, requiring an average of 7.66 secs to perform separation. This implies that the computational complexity of the proposed MDLD algorithm does not scale considerably with the number of sources L and sensors K . In contrast, the ‘‘GaussSep’’ algorithm’s processing has increased considerably with K . The processing time seems to scale up dramatically with increasing K and number of estimated sources L . For $K = 3$, it required an average of 1310 sec and for $K = 4$, it required 2359 sec which is almost the double processing time for $K = 3$. Thus, it appears that the proposed MDLD algorithm is capable of offering a faster and more stable multichannel solution to the underdetermined source separation problem, featuring higher SIR rates, compared to a state-of-the-art approach.

The main aspiration for future work behind these experiments is to combine the speed and stability of the MDLD approach with the low-artifact separation quality, proposed by Vincent et al [58]. It might be possible to import this time-frequency localised source separation framework, where the source clusters can be modeled by mixtures of MDLDs. A more intelligent fuzzy clustering algorithm may combine the information from the MDLD priors to attribute points to multiple sources, overcoming the artifacts that arise from the partitioning of the time-frequency space.

8 Conclusions - Possible Extensions

In this chapter, the problem of underdetermined instantaneous source separation is addressed. Since the data can have sparse representations in a transform domain, it is rational to use mixtures of heavy-tailed distributions, such as the Laplacian distribution, to model each source’s distribution in the mixture environment. As the main concentrations of data appear on the directions spanned by the columns of the mixing matrix, the source separation problem is transformed to an angular clustering problem. In other words, data that need to be processed are directional, that the use of Laplacian distributions with infinite support is not efficient for sources near 0, or π directions. The first improvement was to wrap the ordinary Laplacian distribution and create a Wrapped Laplacian distribution. Training mixture of the Wrapped Laplacian Distribution is computationally expensive due to the concurrent estimation of two EM algorithms. The existence of closed-form directional Gaussian models inspired the introduction of a Laplacian directional model. Building on previous work on directional Gaussian models (i.e. the von-Mises and the vonMises-Fisher densities) to propose a novel generalised Directional Laplacian model for modelling multidimensional directional sparse data. Maximum Likelihood estimates of the densities’ parameters were proposed along with an EM-algorithm that handles the training of DLD mixtures. The proposed algorithms were tested and demonstrated good performance in modelling the directionality of the data. The proposed

algorithm can also provide a solution for the general multichannel underdetermined source separation problem ($K \geq 2$), offering fast and stable performance and high SIR compared to state-of-the-art methods [58].

Possible extensions is to adapt this technique for a convolutive mixture scenario, where using the Short-Time Fourier Transform, we transform the convolutive mixtures into multiple complex instantaneous mixtures. Source separation-clustering for each frequency bin can be performed using a modified version of the proposed algorithm for complex numbers and permutation alignment can be performed using Time-Frequency Envelopes or Direction-of-Arrival methods as proposed by Mitianoudis and Davies [42, 43] or Sawada et al [54]. The speed of the proposed MDLD algorithm can be a very positive feature for FD-BSS, since these methods need to solve many complex instantaneous source separation problems simultaneously.

Another possible direction is to adapt the proposed technique for underdetermined PNL mixtures. Once the mixtures have been linearised by the blind compensation method of Duarte et al [22], it is always possible to use the proposed technique to unmix the PNL mixtures in the linear stage. The speed of the proposed MDLD algorithm may expedite the blind estimation of the inverse non-linear function of PNL mixtures.

Appendix 1

Calculation of the normalisation parameter for the Generalised DLD

To estimate the normalisation coefficient $c_p(k)$ of (27), we need to solve the following equation:

$$\int_{\mathbf{x} \in \mathcal{S}^{p-1}} c_p(k) e^{-k\sqrt{1-(\mathbf{m}^T \mathbf{x})^2}} d\mathbf{x} = 1 \quad (43)$$

Following equation (B.8) and in a similar manner to the analysis in Appendix B.2 in [21], we can rewrite the above equation as follows:

$$c_p(k) \int_0^\pi d\theta_{p-1} \int_0^\pi e^{-k\sqrt{1-\cos^2\theta_1}} \sin^{p-2}\theta_1 d\theta_1 \prod_{j=3}^{p-1} \int_0^\pi \sin^{p-j}\theta_{j-1} d\theta_{j-1} = 1 \quad (44)$$

Following a similar methodology to Appendix B.2 in [21], the above yields:

$$c_p(k) \pi \int_0^\pi e^{-k\sin\theta_1} \sin^{p-2}\theta_1 d\theta_1 \frac{\pi^{\frac{p-3}{2}}}{\Gamma(\frac{p-1}{2})} = 1 \quad (45)$$

Using the definition of $I_p(k)$, we can write

$$c_p(k)I_{p-2}(k)\frac{\pi^{\frac{p+1}{2}}}{\Gamma(\frac{p-1}{2})} = 1 \Rightarrow c_p(k) = \frac{\Gamma(\frac{p-1}{2})}{\pi^{\frac{p+1}{2}}I_{p-2}(k)} \quad (46)$$

Appendix 2

Gradient updates for \mathbf{m} and k for the MDDL

The first order derivative of the log-likelihood in (28) for the estimation of \mathbf{m} are calculated below:

$$\begin{aligned} \frac{\partial J(\mathbf{X}, \mathbf{m}, k)}{\partial \mathbf{m}} &= -k \sum_{n=1}^N \frac{-2\mathbf{m}^T \mathbf{x}_n}{2\sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2}} \mathbf{x}_n \\ &= k \sum_{n=1}^N \frac{\mathbf{m}^T \mathbf{x}_n}{\sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2}} \mathbf{x}_n \end{aligned} \quad (47)$$

Before we estimate k from the log-likelihood (28), we derive the following property:

$$\frac{\partial}{\partial k} I_0(k) = -\frac{1}{\pi} \int_0^\pi e^{-k \sin \theta} \sin \theta d\theta = -I_1(k) \quad (48)$$

The above property can be generalised as follows:

$$\frac{\partial^p}{\partial k^p} I_0(k) = (-1)^p \frac{1}{\pi} \int_0^\pi \sin^p \theta e^{-k \sin \theta} d\theta = (-1)^p I_p(k) \quad (49)$$

The first order derivative of the log-likelihood in (28) for the estimation of k is then calculated below:

$$\frac{\partial J(\mathbf{X}, \mathbf{m}, k)}{\partial k} = N \frac{I_{p-1}(k)}{I_{p-2}(k)} - \sum_{n=1}^N \sqrt{1 - (\mathbf{m}^T \mathbf{x}_n)^2} \quad (50)$$

Appendix 3

A Directional K-Means algorithm

Assume that K is the number of clusters, \mathcal{C}_i , $i = 1, \dots, K$ are the clusters, \mathbf{m}_i are the cluster centres and $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n, \dots, \mathbf{x}_N\}$ is a p -D angular dataset lying on the half-unit p -D sphere. The original K-means [39] minimises the following non-directional error function:

$$Q = \sum_{n=1}^N \sum_{i=1}^K \|\mathbf{x}_n - \mathbf{m}_i\|^2 \quad (51)$$

where $\|\cdot\|$ represents the Euclidean distance. Instead of using the squared Euclidean distance for the p -D Directional K-Means, we introduce the following distance function:

$$D_l(\mathbf{x}_n, \mathbf{m}_i) = \sqrt{1 - (\mathbf{m}_i^T \mathbf{x}_n)^2} \quad (52)$$

The novel function D_l is similarly monotonic as the original distance but emphasizes more on the contribution of points closer to the cluster centre. In addition, D_l is periodic with period π . The p -D Directional K-Means can thus be described as follows:

1. Randomly initialise K cluster centres \mathbf{m}_i , where $\|\mathbf{m}_i\| = 1$
2. Calculate the distance of all points \mathbf{x}_n to the cluster centres \mathbf{m}_i , using D_l .
3. The points with minimum distance to the centres \mathbf{m}_i form the new clusters \mathcal{C}_i .
4. The clusters \mathcal{C}_i vote for their new centres \mathbf{m}_i^+ . To avoid averaging mistakes with directional data, vector averaging is employed to ensure the validity of the addition. The resulting average is normalised to the half-unit p -D sphere:

$$\mathbf{m}_i^+ = \frac{1}{k_i} \sum_{\mathbf{x}_n \in \mathcal{C}_i} \mathbf{x}_n \quad (53)$$

$$\mathbf{m}_i^+ \leftarrow \mathbf{m}_i^+ / \|\mathbf{m}_i^+\| \quad (54)$$

5. Repeat steps 2), 3), 4) until the means \mathbf{m}_i have converged.

References

1. SiSEC 2008: Signal Separation Evaluation Campaign. URL <http://sisec2008.wiki.irisa.fr/tiki-index.php>
2. SiSEC 2010: Signal Separation Evaluation Campaign. URL <http://sisec2010.wiki.irisa.fr/tiki-index.php>
3. SiSEC 2011: Signal Separation Evaluation Campaign. URL <http://sisec.wiki.irisa.fr/tiki-index.php>
4. Araki, S., Sawada, H., Mukai, R., Makino, S.: Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors. *Signal Processing* **87**, 1833–1847 (2007)
5. Arberet, S., Gribonval, R., Bimbot, F.: A robust method to count and locate audio sources in a multichannel underdetermined mixture. *IEEE Trans. on Signal Processing* **58**(1), 121–133 (2010)
6. Attias, H.: Independent factor analysis. *Neural Computation* **11**(4), 803–851 (1999)
7. Banerjee, A., Dhillon, I.S., Ghosh, J., Sra, S.: Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning Research* **6**, 1345 – 1382 (2005)
8. Bentley, J.: Modelling circular data using a mixture of von Mises and uniform distributions. Simon Fraser University, MSc thesis (2006)
9. Bilmes, J.: A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Mixture Models. Tech. rep., Department of Electrical Engineering and Computer Science, U.C. Berkeley, California (1998)
10. Blumensath, T., Davies, M.: Sparse and shift-invariant representations of music. *IEEE Transactions on Audio, Speech and Language Processing* **14**(1), 50–57 (2006)
11. Bofill, P., Zibulevsky, M.: Underdetermined blind source separation using sparse representations. *Signal Processing* **81**(11), 2353–2362 (2001)

12. Cardoso, J.F.: Blind signal separation: statistical principles. *Proceedings of the IEEE* **9**(10), 2009–2025 (1998)
13. Cemgil, A., Févotte, C., Godsill, S.: Variational and stochastic inference for bayesian source separation. *Digital Signal Processing* **17**, 891913 (2007)
14. Cichocki, A., Amari, S.: *Adaptive Blind Signal and Image Processing*. John Wiley and Sons (2002)
15. Comon, P., Jutten, C.: *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Academic Press (2010). 856 pages
16. Daudet, L., Sandler, M.: MDCT analysis of sinusoids: explicit results and applications to coding artifacts reduction. *IEEE Trans. on Speech and Audio Processing* **12**(3), 302 – 312 (2004)
17. Davies, M., Daudet, L.: Sparse audio representations using the mclt. *Signal Processing* **86**(3), 358–368 (2006)
18. Davies, M., Mitianoudis, N.: A simple mixture model for sparse overcomplete ICA. *IEE proceedings in Vision, Image and Signal Processing* **151**(1), 35–43 (2004)
19. Dempster, A.P., Laird, N., Rubin, D.: Maximum likelihood for incomplete data via the EM algorithm. *J. of the Royal Statistical Society, ser. B* **39**, 1–38 (1977)
20. Devroye, L.: *Non-Uniform Random Variate Generation*. New York: Springer-Verlag (1986)
21. Dhillon, I., Sra, S.: *Modeling Data using Directional Distributions*. Tech. rep., Technical Report TR-03-06, University of Texas at Austin, Austin, TX (2003)
22. Duarte, L., Suyama, R., Rivet, B., Attux, R., Romano, J., Jutten, C.: Blind compensation of nonlinear distortions: Application to source separation of post-nonlinear mixtures. *IEEE Trans on Signal Processing* **60**(11), 5832–5844 (2012)
23. Duong, N., Vincent, E., Gribonval, R.: Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Trans. Audio, Speech, and Language Processing* **18**(7), 1830–1840 (2010)
24. Eriksson, J., Koivunen, V.: Identifiability, separability, and uniqueness of linear ica models. *IEEE Signal Processing Letters* **11**(7), 601–604 (2004)
25. Févotte, C., Godsill, S.: A bayesian approach to blind separation of sparse sources. *IEEE Trans. on Audio, Speech and Language Processing* **14**(6), 2174–2188 (2006)
26. Févotte, C., Gribonval, R., Vincent, E.: *BSS EVAL Toolbox User Guide*. Tech. rep., IRISA Technical Report 1706, Rennes, France, April 2005, http://www.irisa.fr/metiss/bss_eval/
27. Fisher, N.: *Statistical Analysis of Circular Data*. Cambridge University Press (1993)
28. Girolami, M.: A variational method for learning sparse and overcomplete representations. *Neural Computation* **13**(11), 2517–2532 (2001)
29. Gribonval, R., Nielsen, M.: Sparse decomposition in unions of bases. *IEEE Trans. Information Theory* **49**(12), 3320–3325 (2003)
30. Hyvärinen, A.: Independent Component Analysis in the presence of Gaussian Noise by Maximizing Joint Likelihood. *Neurocomputing* **22**, 49–67 (1998)
31. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. John Wiley, New York (2001). URL <http://www.cis.hut.fi/projects/ica/book/>. 481+xxii pages
32. Jammalamadaka, S., Sengupta, A.: *Topics in Circular Statistics*. World Scientific (2001)
33. Jutten, C., Karhunen, J.: Advances in nonlinear blind source separation. In: *Proc. of 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pp. 245–256. Nara, Japan (2003)
34. Kreyszig, E.: *Advanced Engineering Mathematics*. Wiley (1998)
35. Lee, T.W., Bell, A.J., Lambert, R.: Blind separation of delayed and convolved sources. In: *Advances in Neural Information Processing Systems*, vol. 9, pp. 758–764. MIT Press (1997)
36. Lee, T.W., Lewicki, M., Girolami, M., Sejnowski, T.: Blind source separation of more sources than mixtures using overcomplete representations. *IEEE Signal Processing Letters* **4**(5) (1999)
37. Lewicki, M.: Efficient coding of natural sounds. *Nature Neuroscience* **5**(4), 356–363 (2002)
38. Lewicki, M., Sejnowski, T.: Learning Overcomplete Representations. *Neural Computation* **12**, 337–365 (2000)

39. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. Berkeley, California (1967)
40. Mardia, K., Kanti, V., Jupp, P.: *Directional Statistics*. Wiley (1999)
41. Mitianoudis, N.: A Directional Laplacian Density for Underdetermined Audio Source Separation. In: 20th International Conference on Artificial Neural Networks (ICANN). Thessaloniki, Greece (2010)
42. Mitianoudis, N., Davies, M.: Audio source separation of convolutive mixtures. *IEEE Trans. Audio and Speech Processing* **11**(5), 489–497 (2003)
43. Mitianoudis, N., Davies, M.: Permutation alignment for Frequency Domain ICA using subspace beamforming methods. In: Proc. Int. Workshop on Independent Component Analysis and Source Separation (ICA2004), pp. 127–132. Granada, Spain (2004)
44. Mitianoudis, N., Stathaki, T.: Batch and Online Underdetermined Source Separation using Laplacian Mixture Models. *IEEE Transactions on Audio, Speech and Language Processing* **15**(6), 1818–1832 (2007)
45. Mitianoudis, N., Stathaki, T.: Underdetermined Source Separation using Mixtures of Warped Laplacians. In: International Conference on Independent Component Analysis and Source Separation (ICA). London, UK (2007)
46. Moulines, E., Cardoso, J.F., Gassiat, E.: Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models. In: Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97), pp. 3617–3620. Munich, Germany (1997)
47. O'Grady, P., Pearlmutter, B.: Hard-LOST: Modified K-Means for oriented lines. In: Proceedings of the Irish Signals and Systems Conference, pp. 247–252. Ireland (2004)
48. O'Grady, P., Pearlmutter, B.: Soft-LOST: EM on a mixture of oriented lines. In: Proc. International Conference on Independent Component Analysis 2004, pp. 428–435. Granada, Spain (2004)
49. Pajunen, P., Hyvärinen, A., Karhunen, J.: Nonlinear blind source separation by self-organizing maps. In: Proc. Int. Conf. on Neural Information Processing, pp. 1207–1210. Hong Kong (1996)
50. Plumbley, M., Abdallah, S., Blumensath, T., Davies, M.: Sparse representations of polyphonic music. *Signal Processing* **86**(3), 417–431 (2006)
51. Rickard, S., Balan, R., Rosca, J.: Real-time time-frequency based blind source separation. In: Proc. ICA2001, pp. 651–656. San Diego, CA (2001)
52. Sawada, H., Araki, S., Makino, S.: A two-stage frequency-domain blind source separation method for underdetermined convolutive mixtures. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2007), pp. 139–142 (2007)
53. Sawada, H., Araki, S., Makino, S.: Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. *IEEE Trans. Audio, Speech, and Language Processing* **19**(3), 516–527 (2011)
54. Sawada, H., Mukai, R., Araki, S., Makino, S.: A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE Trans. Speech and Audio Processing* **12**(5), 75–87 (2004)
55. Smaragdis, P.: Blind separation of convolved mixtures in the frequency domain. *Neurocomputing* **22**, 21–34 (1998)
56. Smaragdis, P., Boufounos, P.: Position and trajectory learning for microphone arrays. *IEEE Trans. Audio, Speech, and Language Processing* **15**(1), 358–368 (2007)
57. Torrkola, K.: Blind separation of delayed and convolved sources. In: S. Haykin (ed.) *Unsupervised Adaptive Filtering*, Vol. I, pp. 321–375. Wiley (2000)
58. Vincent, E., Arberet, S., Gribonval, R.: Underdetermined instantaneous audio source separation via local gaussian modeling. In: 8th Int. Conf. on Independent Component Analysis and Signal Separation (ICA), pp. 775–782. Paraty, Brazil (2009)
59. Vincent, E., Gribonval, R., Fevotte, C., Nesbit, A., Plumbley, M., Davies, M., Daudet, L.: BASS-dB: the blind audio source separation evaluation database. URL <http://bass-db.gforge.inria.fr/BASS-dB/>

60. Winter, S., Kellermann, W., Sawada, H., Makino, S.: Map based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and l1-norm minimization. *EURASIP J. Adv. Signal Process.* **1** (2007)
61. Yilmaz, O., Rickard, S.: Blind separation of speech mixtures via time-frequency masking. *IEEE Trans. Signal Processing* **52**(7), 1830–1847 (2004)
62. Zibulevsky, M., Kisilev, P., Zeevi, Y., Pearlmutter, B.: Blind source separation via multinode sparse representation. *Advances in Neural Information Processing Systems* **14**, 1049–1056 (2002)