Blind Source Separation Of Hard Clipped Audio Mixtures

By

Adahlia Sophie Lux

A thesis submitted to
the University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham
April 2023

ABSTRACT

Blind source separation of hard-clipped audio mixtures is a challenging problem. This thesis presents a novel approach to recovering sources from mixtures affected by hard clipping distortion. This approach relies on making geometric observations about the nature of clipped mixtures. An assumption of disjointness is key. A strict time-disjointness assumption is made in the first two methods. In the final method, we relax this assumption such that only a portion of the time points are time-disjoint. Evaluation experiments demonstrated an increase in source recovery performance for the proposed method when compared with sequential approaches when at least partial time-disjointness was present for speech and synthetic sources.

DEDICATION

Dedicated to my amazing mum. Я тебя люблю, мама.

ACKNOWLEDGMENTS

I would like to thank Ata Kabán for her diligent supervision and her fantastic advice when I started my PhD.

Thank you to the Joseph Chamberlain Scholarship and EPSRC for funding my studies.

Thank you to everyone else who has helped me throughout my studies, especially Rosie.

Trans rights

are

human rights

Contents

Page

1 Introduction 1

1.1 The cocktail party problem 1

1.2 Independent component analysis (ICA) 2

1.2.1 Preprocessing 4

1.3 Properties of audio source separation 4

1.3.1 Linearity 6

1.3.2 Noise 6

1.3.3 Determinedness 7

1.3.4 Time independence 7

1.3.5 Availability of information 8

1.3.6 Variance over time 8

1.3.7 Audio content 9

1.4 The spectral or time-frequency domain 9

1.4.1 Short Time Fourier Transform 9

1.4.2 Wavelet Transform 10

1.5 Clipping distortion 11

1.6 Thesis contributions 12

2 Literature Review 15

2.1 Independent Component Analysis (ICA) and Blind Source Separation (BSS) 15

2.1.1 ICA model recap 15

2.1.2 ICA by minimisation of mutual information 16

2.1.3 Contrast functions or non-Gaussianity functions 17

2.1.4 FastICA 20

2.1.5 Maximum likelihood ICA 22

2.1.6 ICA research 23

2.1.7 Performance of ICA 24

2.2 Signal reconstruction and declipping 25

2.2.1 Compressed sensing 25

2.2.2 Compressed sensing for declipping 27

2.2.3 Dictionary learning for sparse representations 28

2.2.4 Joint declipping and informed source separation 30

2.3 Alternative source separation approaches 30

2.3.1 Informed source separation 30

2.3.2 Binary masking for underdetermined separation 32

2.3.3 Non-negative matrix factorisation (NMF) for single mixture separation . . . 34

2.4 Criticisms of the literature 36

2.4.1 Nonrobustness of existing independent component analysis (ICA) methods for clipping distortion 36

2.4.2 No model for declipping and blind source separation 37

2.4.3 No model for inverse dynamic range compression (DRC) and source separation 39

3 One Clipped Mixture and One Unclipped Mixture of Two Sources 41

3.1 Introduction 41

3.2 Methodology summary 43

3.3 Experiment results summary 45

3.4 Paper One 46

4 Two Clipped Mixtures of Many Sources 53

4.1 Methodology summary 53

4.2 Experiment results summary 55

4.3 Paper Two 56

5 Two Clipped Mixtures of Many Sources with some Time-Disjointness 63

5.1 Introduction 63

5.2 Methodology 64

5.2.1 Overview of the proposed approach 64

5.2.2 Mixing matrix estimation 65

5.2.3 Declipping by ℓ_1 minimisation 66

5.2.4 Source estimation 68

5.2.5 Sequential method 68

5.3 Experiments 69

5.3.1 Hypotheses 70

5.3.2 Preparing the data 70

5.3.3 Implementation and frame-based processing 72

5.3.4 Performance measurement 72

5.4 Results 73

5.5 Conclusion 79

6 Discussions and Conclusions 80

6.1 Research summary 80

6.2 Results and conclusions 81

6.3 Future work 82

6.3.1 Time-frequency disjointness 82

6.3.2 Additional performance measures 82
6.3.3 Additional source separation algorithms 83
6.3.4 Computational learning theory 83
6.3.5 Problematic mixtures 84

References **86**

List of Figures

1.1 Example of a mixture of 2 uniform sources and the sources after preprocessing has taken place. 5

1.2 The connection between time domain and spectral domain, illustrated with a chirp signal. 11

1.3 An unclipped sine wave (left) and a clipped sine wave (right) 12

2.1 Overall Perceptual Score (OPS) versus clipping threshold (hard clipping) 37

2.2 OPS versus clipping threshold (soft *tanh* clipping) 38

3.1 Effects of clipping on one of two mixtures. 42

5.1 Two speech sources, 50% time disjoint (left); linear mixtures of these sources (middle); clipped mixtures, 20% clipped (right). 65

5.2 Sequential method yields better performance than joint method at 256 window size (2 sine sources, 256 window size, 0.5% time-disjointness) 74

5.3 Sine sources yield better performances than speech and Gaussian (2 sources, 64 window size, 5% time-disjointness) 76

5.4 FastICA yields better performs than our approach at 0% time-disjointness (2 sources, 256 window size, 0% time-disjointness) 77

5.5 Increasing the number of sources worsens performance (Sine sources, 256 window size, 2% time-disjointness) 78

Acronyms

DCT Discrete Cosine Transform. 28, 67, 70

DRC dynamic range compression. ii, 9, 39

ICA independent component analysis. i, ii, 2–4, 6–8, 12, 13, 15–17, 19, 20, 23–25, 28, 32, 36, 37, 39, 42, 43, 53, 64, 72, 74

NMF non-negative matrix factorisation. ii, 31, 34–36, 83

NTF non-negative tensor factorisation. 30, 37, 38

OPS Overall Perceptual Score. v, 36–38

STFT short time Fourier transform. 9–11, 32, 35

Chapter One

Introduction

This thesis is composed of a series of published and unpublished works that describe the original research undertaken by the author. These works have been published by the author under three different legal names.

This thesis is about blind source separation, which is best explained by *the cocktail party problem.*

1.1 The cocktail party problem

Imagine you are at a cocktail party, and there are lots of guests speaking at the same time. Despite the overlap of the guests' voices in time, our brains are capable of focusing on and identifying an individual given voice, or source, at a time. Our brains separate a desired source from the mixture - the entire cocktail party. This task, however, is challenging for a computer. In signal processing, this is known as the cocktail party problem; it is more generally known as *source separation*. This problem can be extended to any other type of source; these might be medical signals such as electrocardiograms; musical instruments; video; or astronomical signals.

To help understand blind source separation, we begin by describing the most common set of mathematical approaches to it, Independent component analysis (ICA).

1.2 Independent component analysis (ICA)

ICA describes a set of techniques and there are many ways to formulate the problem. Generally, when discussing ICA, we are referring to linear ICA. ICA is a latent variable model, which means the variables we are interested in cannot be directly observed (Hyvärinen and Oja, 2000b). The noiseless linear case can be expressed with the equation below: the observed mixtures $\mathbf{x}$ are modelled as linear mixtures of the unknown source signals $\mathbf{s}$, which are achieved by the unknown mixing matrix $\mathbf{A}$.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1.1}$$

The noisy linear case can be expressed by assuming there is an additive constant noise $\mathbf{n}$. For noisy ICA, we generally assume that the noise is i.i.d. (independent and identically distributed) Gaussian noise. In the real world, noise is often more complex than a simple additive constant, however, treating noise this way is a useful simplification for ICA applications, where we are interested in the sources rather than the noise itself.

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{1.2}$$

ICA is an ill-posed problem because it has two unknowns $\mathbf{A}$ and $\mathbf{s}$. In order to make the problem solvable, we have to make some assumptions: the main assumption is the statistical independence of the sources $\mathbf{s}$. If we think back to the cocktail party problem (Section 1.1) then it is easy to see this is a reasonable assumption, as each person is stood in a different location and is speaking at a different loudness. A second assumption is required, that all but one of the source signals must have non-Gaussian distributions. This is a reasonable assumption: many signals, such as speech and musical instruments, are non-Gaussian in nature. The reason for this assumption comes from the central limit theorem (CLT): as we sum source signals to generate a mixture, their overall distribution will tend towards a Gaussian. For this reason we are unable to distinguish a Gaussian source from a Gaussian mixture (Hyvärinen and Oja, 2000b). This principle of non-Gaussianity is fundamental to many ICA algorithms.

Under these two assumptions, the ICA model is identifiable, which means that it is possible to discern the parameters of the model given sufficient measurements (Hyvärinen, Karhunen, and Oja, 2001, p. 153). In the determined case where there is the same number of sources **s** and mixtures **x**, it is usually assumed that the matrix **A** is invertible. Many algorithms deal with the determined case, such as FastICA, requiring the same number of mixtures **x** as sources **s**, making the mixing matrix **A** square. In the noiseless linear case, this assumption allows us to invert the estimated mixing matrix, $\mathbf{W} = \widehat{\mathbf{A}^{-1}}$, and simply recover the estimated sources $\widehat{\mathbf{s}}$ by the following equation:

$$\widehat{\mathbf{s}} = \mathbf{W}\mathbf{x} \tag{1.3}$$

The ICA model cannot recover the scaling or signs of the sources. Since both **A** and **s** are unknowns, scaling factors in either could easily be cancelled by division in the other. A further indeterminacy evident in Equation 1.1 is that the original permutation of the sources cannot be recovered either.

Although ICA can work in certain scenarios, it usually requires preprocessing of the data in order for it to achieve separation of the sources.

1.2.1 Preprocessing

ICA assumes that the sources are statistically independent, linearly mixed and with equal variance. This might not always be the case, so preprocessing the data can help by removing scaling factors and mitigating the effects of nonlinear mixing and noise.

Linear mixing can be viewed as a rotation, once scaling factors have been removed. This is useful because we can rotate the data in order to undo the linear mixing process. This can be seen in Figure 1.1: a scatter plot of two independent sources is shown in Figure 1.1a and a scatter plot of the mixtures, after preprocessing to remove scaling factors, is shown in Figure 1.1c. To generate the mixtures, we used MATLAB's *rand* function to generate two example sources, s_1 and s_2, 1000 points each drawn from a uniform distribution. then multiplied the resulting data by mixing matrix **A** (Equation 1.4) to yield mixtures x_1 and x_2.

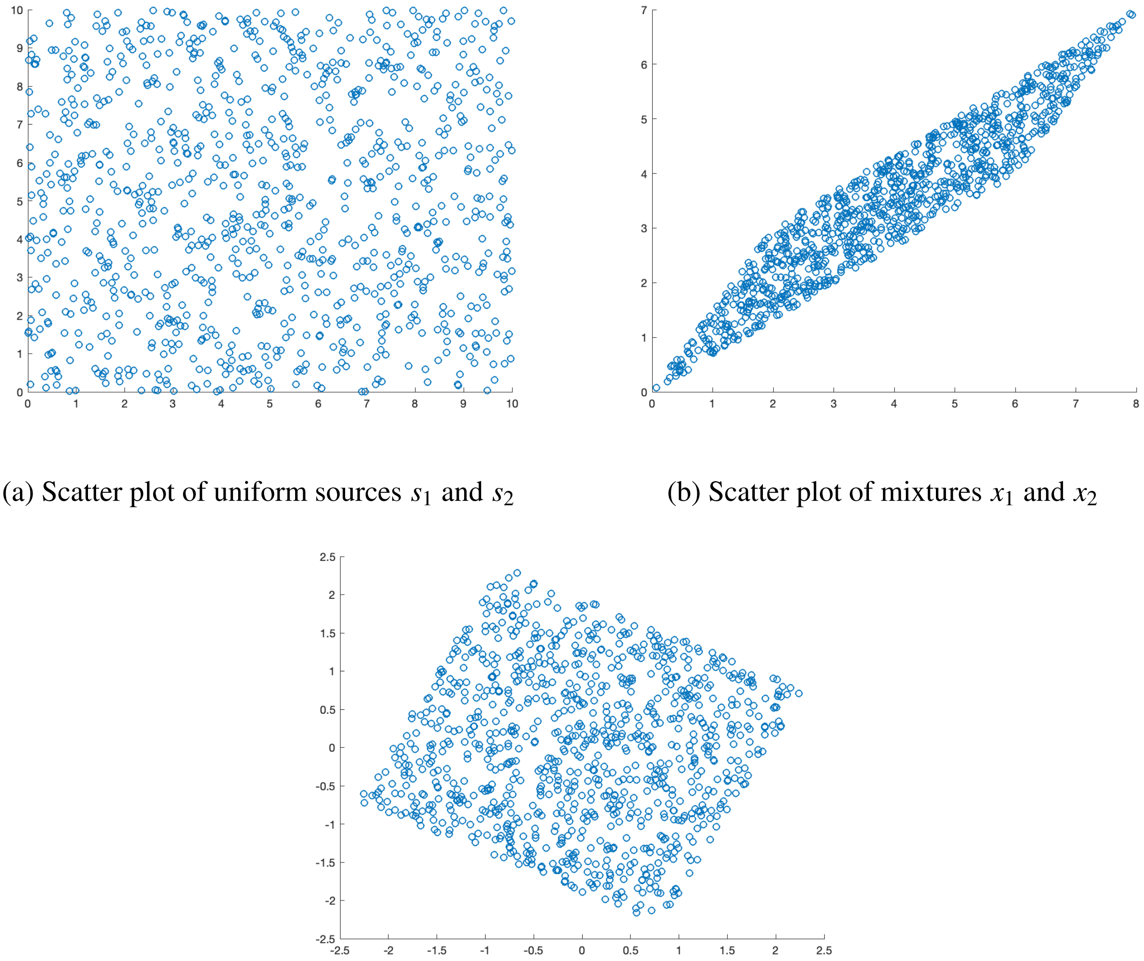
$$\mathbf{A} = \begin{bmatrix} 0.2 & 0.3 \\ 0.6 & 0.4 \end{bmatrix} \tag{1.4}$$

Two steps are usually involved in the preprocessing for linear ICA: the first, known as *centring*, refers to shifting the data to have zero mean in all dimensions. The second, known as *whitening*, refers to decorrelating and scaling the data to have identity covariance (Hyvärinen, Karhunen, and Oja, 2001, p.140). ICA is discussed in more detail in Section 2.1.

1.3 Properties of audio source separation

For the purpose of this thesis we will be focusing on audio source separation, however the techniques in this thesis could be applied to other source types.

Source separation is a broad problem. There are several parameters with which one can



(a) Scatter plot of uniform sources s_1 and s_2

(b) Scatter plot of mixtures x_1 and x_2

(c) Scatter plot of centered and whitened mixtures x'_1 and x'_2

Figure 1.1: Example of a mixture of 2 uniform sources and the sources after preprocessing has taken place.

specify a source separation problem and these parameters can be combined, leading to many permutations. The main parameters are listed below and discussed briefly to assess their relevance and provide context to the problems approached in this thesis.

1.3.1 Linearity

Linearity is an important parameter of mixtures. In the real world, mixtures of audio can be modelled as linear systems. This is also true in music production, although the linear audio mixtures produced are further processed after mixing, introducing a nonlinearity. Jutten and Karhunen described these mixtures as post-nonlinear (PNL) mixtures (Jutten and Karhunen, 2003). This is a simpler problem than general nonlinear mixtures: provided the nonlinearity is invertible, the problem can be solved by linear ICA. Moreover, general nonlinear ICA problems are not identifiable without further constraints and regularisation. For this reason, linear ICA approaches will be most relevant to the aims of this research.

1.3.2 Noise

In signal processing, noise is unavoidable. Both analogue and digital systems suffer from noise, which can be introduced at capture, processing, transmission and storage for a variety of reasons. Some ICA methods handle noise by considering it to be merely another source, while others include noise separately in the model. Hyvärinen and Oja remarked that the noiseless model was challenging enough to solve and that, for many applications, appeared to be sufficient (Hyvärinen and Oja, 2000b).

1.3.3 Determinedness

A source separation problem is determined if there are the same number of source signals as there are mixtures. Many ICA algorithms, such as FastICA (Hyvärinen and Oja, 2000b) and EMICA (Welling and Weber, 2001), deal with the determined case. In the determined case the mixing matrix is square, which simplifies the problem.

Overdetermined problems, with more mixtures than sources, are not relevant to this research: dimensionality reduction techniques such as principal component analysis (PCA) can be used to convert these into determined problems (Winter, Sawada, and Makino, 2005).

A more challenging, and perhaps realistic, problem is the underdetermined case, where one has access to fewer mixtures than source signals. Unlike determined and overdetermined problems, we cannot simply use the inverse or pseudo-inverse of mixing matrix $\mathbf{A}$ (Plumbley et al., 2010). Stereo audio, or two mixtures, is very common, however, music productions usually contain many more than two source signals. More challenging still is single mixture or single channel source separation. The single mixture problem, and underdetermined source separation in general, can be modelled in the spectral domain Section 1.4.

1.3.4 Time independence

While ICA algorithms are built on the notion that the source signals are independent, this is not a realistic assumption (Hyvärinen, 2013). In musical recordings, especially, there are structures to be found in time and frequency (Vincent et al., 2014): instruments are correlated due to playing the same rhythms and melodies in time. Most ICA algorithms treat the time points as identically distributed (i.i.d.) and work in practice. It is worth considering, however, that the time dependence of the sources can provide additional insights about the nature of the sources.

1.3.5 Availability of information

The blind source separation problem, which ICA was developed to solve, assumes access to the mixtures and no further information. In reality, no source separation algorithm is truly blind due to the assumptions that need to be made about the nature of the sources (Parvaix and Girin, 2011), the mixing process and the number of sources present. In general, informed or guided source separation refers to approaches that account for further side information about the mixtures, such as a musical score or instrument activation times. Since additional information beyond the mixtures themselves is unavailable in many scenarios, this research will focus on blind source separation methods. For more challenging problems, however, such as time-varying distortions, informed source separation methods may prove necessary in order to achieve good results.

1.3.6 Variance over time

A mixture is instantaneous if it contains no echoes or time delays. Convolutive mixtures, which can contain multiple time delayed instances of the sources, are a more challenging problem. Convolution is commonly used in music production to create the impression of space, mimicking the multiple reflections of sound waves heard in a physical location. Convolutive source separation algorithms often deal with time-invariant mixtures rather than time-varying mixtures (Kounades-Bastian et al., 2015), the latter giving the impression of sources moving from one loudspeaker to another. Convolutive mixtures are less relevant in solving the problems considered by this research and as such are not a priority. Primarily, instantaneous mixtures will be considered, though future work could extend this research to convolutive mixtures.

1.3.7 Audio content

Finally, the audio content should be considered. There are methods that are designed to work exclusively for speech or music, for example. Many informed source separation methods require a musical score or other musical cues, which makes them suitable only for musical data (Ewert et al., 2014). Since clipping distortion and dynamic range compression (DRC) are widespread phenomena, it would be appropriate to focus on general methods that will work for any type of audio. For more challenging types of distortion, however, approaches that make assumptions based on the audio content being music or speech would be useful.

1.4 The spectral or time-frequency domain

A plot of the time points of a signal is known as a waveform, however, it is often useful to transform a signal into the spectral, or time-frequency, domain. Signals may overlap in the time domain, but this may not be the case in the spectral domain. Sources comprise different frequencies, and separating these frequencies can help with identifying the sources. These are some of the reasons the spectral domain is useful for source separation.

1.4.1 Short Time Fourier Transform

Fixed-basis transforms such as the short time Fourier transform (STFT) are easily invertible and provide unique representations of a signal (Nesbit et al., 2010): as such, they are commonly used in audio. The equation for the discrete version of STFT is given in Equation 1.5. In the equation, n is the time point index; t is the window or frame number; f is the frequency; N is the window size; h is the hop size, or the number of time points by which the window moves each time.

$$\textbf{STFT}(t,f) \text{ of } y_n \equiv Y(t,f) = \sum_{n=1}^{N} w_n y_{n+th} \exp^{-j2\pi fn} \tag{1.5}$$

The STFT works on overlapping segments of a signal at a time by shifting a window w across the signal y. There are several types of windowing function one could use for w. One of the most common choices is the Hann window:

$$w_n = 0.5\left(1 - cos\left(\frac{2n\pi}{N+1}\right)\right), n = 1...N \tag{1.6}$$

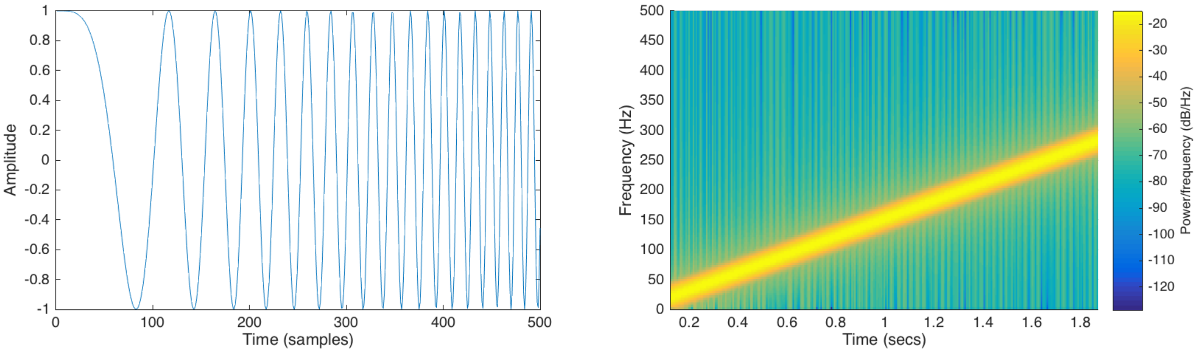
The STFT has a complex-valued output. Converting to polar coordinates gives us a *magnitude* coefficient and *phase* angle for the frequency f at each window determined by t. Note that the time resolution of the time frames t is necessarily coarser than the resolution of the individual time points n. Since frequency is measured over time, some time resolution is necessarily lost in this process. A spectrogram can be created by ignoring phase and plotting the squared magnitude coefficients of the STFT. A chirp signal, which is a signal that increases in frequency over time, was generated in MATLAB. The first 500 samples of signal, which has a sampling rate of 1000 Hz, starts at zero and crosses 150 Hz when t = 1 s, are plotted in Figure 1.2a. The spectrogram of this chirp signal is shown in Figure 1.2b.

1.4.2 Wavelet Transform

Another related tool for time-frequency analysis is the wavelet transform.

Like the discrete STFT, it gives an insight into the properties of a signal changing in time and frequency. Both transforms are useful for time-frequency analysis of signals, and both have applications in various fields such as signal processing and feature extraction.

The key differences are in resolution and localisation. The wavelet transform offers a more



(a) Waveform of a linear chirp signal.

(b) Spectrogram of a linear chirp signal.

Figure 1.2: The connection between time domain and spectral domain, illustrated with a chirp signal.

flexible approach, using variable resolutions, unlike the STFT. As such, the wavelet transform is better suited for analysis signals that are non-stationary. Whereas the STFT localises in time-frequency through windowing, the wavelet transform achieves this through the shifting and scaling of a mother wavelet function. Wavelets are small, localised oscillatory functions, and a mother wavelet is a fundamental component in the signal processing technique of wavelet analysis.

There is no direct equation for the discrete version of the wavelet transform, it is instead typically implemented through filter banks and subsampling.

1.5 Clipping distortion

Source separation becomes even more challenging when the mixtures we observe have been affected by clipping distortion. Clipping distortion occurs when a signal surpasses a threshold and its amplitude is truncated. There are two types of clipping distortion: hard clipping where the peaks of the waveform are literally clipped off (Figure 1.3) and soft clipping where the signal is compressed using a non linear transform. Crain and Van Tasell's study showed that peak clipping increases the speech recognition threshold (SRT) (Crain and Van Tasell, 1994). This effectively means that peak clipping makes quieter voices harder to recognise.

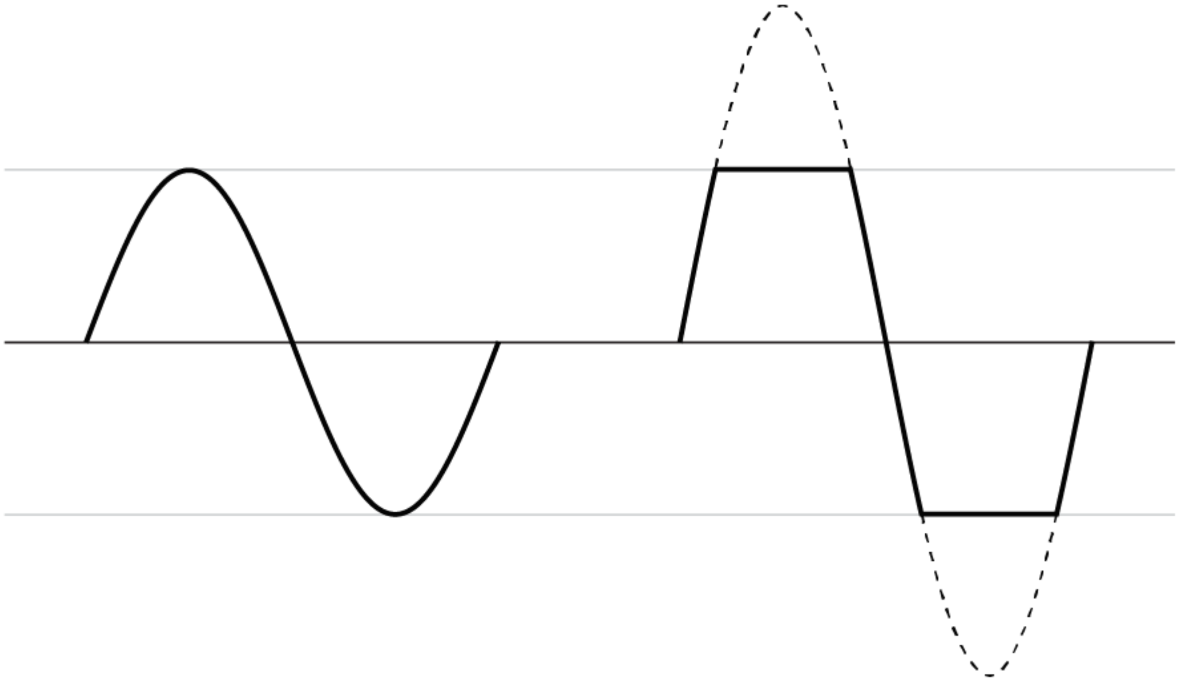


Figure 1.3: An unclipped sine wave (left) and a clipped sine wave (right)

Clipping distortion alters the existing spectral content of the mixtures by introducing new harmonic content and other non-linearities, which can make the original sources harder to identify. This further complicates the problem of source separation. An investigation, by the current author, into existing ICA techniques made before this thesis discovered that they are not robust to clipping distortion (Turl, 2015).

1.6 Thesis contributions

To make this problem solvable, at least approximately, we assume time-disjointness of the sources. We look at both the determined case, where there is the same number of sources as there are mixtures, and the underdetermined case, where there are more sources than mixtures.

To begin to make source separation of clipped mixtures solvable, at least approximately, we make the assumption that the sources are disjoint in time. This means that a maximum of one source signal has a non-zero amplitude value at any given time point. When we plot a scattergram of the time points of a set of linear mixtures of sources, the sources can be discerned from the plot as straight lines. This fact was exploited previously for separating more sources than mixtures (Plumbley et al., 2010). This is the underdetermined case. As we shall see, this particular representation of the sources is key to the novel geometric method described within this thesis. In this approach, there is no need to assume the sources are statistically independent, which is something required in ICA.

The work in Chapter Three covers the most simplified version of our proposed approach: the determined case, where there is the same number of sources and mixtures, and where only one of the two observed mixtures is clipped. The work is included as an inline published paper in Section 3.4. The methodology is summarised in Section 3.2. Empirical experiments and results are summarised in Section 3.3.

In Chapter Four, our approach is extended to consider the case where both mixtures are clipped. We assume two linear mixtures of the sources are observed and that both are clipped. Additionally, we look at the underdetermined case, where we have more sources than mixtures. The work is included as an inline published paper in Section 4.3. The methodology is summarised in Section 4.1. Empirical experiments and results are in Section 4.3.

In Chapter Five we relax our assumption of strict time-disjointness from the preceding chapters. Our final method recovers and scales an estimated mixing matrix under the assumption that only a small portion of the sources are disjoint in time. We combine this with knowledge of the geometry of clipped mixtures to constrain a convex optimisation problem, which we formulate and solve to recover the sources. Section 5.2 details our methodology. Experiments are covered in

Section 5.3. Results are in Section 5.4 and the conclusion in Section 5.5.

Chapter Two

Literature Review

2.1 Independent Component Analysis (ICA) and Blind Source Separation (BSS)

2.1.1 ICA model recap

First, we shall briefly recap the ICA model introduced in Section 1.2. Let us remember the ICA model equation:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2.1}$$

Here we observe **x**, which is an m-dimensional vector, **s** is an n-dimensional latent random vector whose components, also known as sources in source separation, are mutually independent, and **A** is a constant m x n matrix to be found. Usually we assume that $m = n$: this is the so-called determined case, where we have the same number of sources and mixtures.

Note that we also assume non-Gaussianity of the source components. We do this in order to make the model identifiable. Moreover, the central limit theorem (CLT) tells us that the sum of two random variables will be closer to Gaussian than the original variables; this sum will be the most non-Gaussian when it is equal to one of the independent sources.

We can also formulate the model as:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{2.2}$$

Here, we can think of $\mathbf{W}$ as a weight matrix that we aim to learn or estimate. If $\mathbf{W}$ is a square matrix, then its best value is $\mathbf{A}^{-1}$, but of course $\mathbf{A}$ is unknown.

2.1.2 ICA by minimisation of mutual information

One approach to ICA is to minimise the mutual information between the source estimates (Hyvärinen, 1999). Mutual information is a concept in information theory that measures the degree of independence of random variables (Hyvärinen and Oja, 2000b), which is useful since independence is a key assumption in most ICA approaches. Recall, two random variables are statistically independent if they convey no information about each other. More precisely, their joint distribution equals to the product of the individual marginal distributions. First, it will be useful to define differential entropy, which we can use to define mutual information. The differential entropy H of a random vector $\mathbf{y}$ with density $f(.)$ is defined below:

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \tag{2.3}$$

This can be normalised to get negentropy:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{2.4}$$

In Equation 2.4, $\mathbf{y}_{gauss}$ is a random vector with the same covariance matrix as $\mathbf{y}$. Negentropy can thus be considered to be a measure of non-Gaussianity. One can also define the mutual information I between the n (scalar) random variables y_i, $i = 1...n$ using negentropy as follows.

Denote the components of the vector $\mathbf{y} = (y_1, \ldots, y_n)$. Then, expressing mutual information using negentropy and constraining the variables to be uncorrelated yields the following (Hyvärinen, 1999):

$$I(y_1, y_2, ..., y_n) = J(\mathbf{y}) - \sum_i J(y_i) \tag{2.5}$$

As defined in (Hyvärinen, 1999), ICA of a random vector $\mathbf{x}$ can be defined as an invertible transformation, as seen in Equation 2.2, where the matrix $\mathbf{W}$ is determined such that mutual information of the transformed components s_i is minimised. Since information is not affected by multiplicative scalars, we cannot recover the scaling of the latent components s_i.

Negentropy is invariant for invertible linear transforms (Hyvärinen, Karhunen, and Oja, 2001, p. 113). For this reason, finding an invertible transformation $\mathbf{W}$ that minimises mutual information between the source estimates is equivalent to finding the directions in which negentropy is maximised.

2.1.3 Contrast functions or non-Gaussianity functions

In order to use the mutual information definition of ICA, we need to use approximations of negentropy, since it is hard to directly compute it (Hyvärinen, Karhunen, and Oja, 2001, Section 5.5). Some approximations that are commonly used in ICA take the following form:

$$J(y_i) \approx c\left[\mathbb{E}\{G(y_i)\} - \mathbb{E}\{G(v)\}\right]^2 \tag{2.6}$$

In this equation, $J(y_i)$ is the negentropy of a random variable y_i, which has zero mean and unit variance, c is an irrelevant constant, v is a Gaussian variable with zero mean and unit variance and G is a non-quadratic function.

Let us now take random variable y_i to be one mixture in our ICA model, that is $y_i = \mathbf{w}^T\mathbf{x}$ and we get the following:

$$J_G(\mathbf{w}) = \left[\mathbb{E}\{G(\mathbf{w}^T\mathbf{x})\} - \mathbb{E}\{G(v)\}\right]^2 \tag{2.7}$$

At this point, we can use the one-unit contrast function in Equation 2.7 as part of a so-called *deflation* approach, that estimates each component one-by-one (Hyvärinen, 1999). Since we are not interested in estimating each component one-by-one for our proposed method, let us instead look at how to extend this to the entire weight matrix $\mathbf{W}$.

We recall from Equation 2.5 that minimising mutual information is equivalent to maximising the negentropy of the components. This makes it trivial to extend this contrast function in Equation 2.7 to the entire weight matrix $\mathbf{W}$. When we maximise the sum of n one-unit contrast functions, we get the following:

$$\text{Maximise} \quad \sum_{i=1}^{n} J_G(\mathbf{w}_i) \quad \text{wrt} \quad \mathbf{w}_i, \quad i = 1, \ldots, n \tag{2.8}$$

$$\text{subject to} \quad \mathbb{E}\left\{\mathbf{w}_k^T\mathbf{x} \cdot \mathbf{w}_j^T\mathbf{x}\right\} = \delta_{jk}, i, k \in \{1, \ldots, n\}. \tag{2.9}$$

where δ_{jk} is the Kronecker delta, i.e. it takes the value 1 if $j = k$ and 0 otherwise. Equations 2.8-2.9 describe mathematically the FastICA approach to ICA, as a specification of an optimisation problem. We should note, when solving this optimisation problem in practice, the expectation w.r.t. the distribution of $\mathbf{x}$ is replaced with empirical average over the time points available (both in Equation 2.9 and in computing $J_G(\mathbf{w}_i)$). Solving this constrained optimisation problem gives us the entire matrix $\mathbf{W}$, since at the maximum, every vector $\mathbf{w}_i$ gives us one of the rows of the matrix $\mathbf{W}$ and we can then use the ICA transformation in Equation 2.2. Now the question is what to employ as a contrast function. One such function we could use is kurtosis.

Kurtosis, or the fourth-order cumulant, is a traditional measure of the 'peakedness' of a distribution in statistics and can be used to measure non-Gaussianity (Hyvärinen, Karhunen, and Oja, 2001). Excess kurtosis is the difference between a kurtosis measurement and the kurtosis of a Gaussian distribution. Excess kurtosis is, therefore, zero for Gaussian distributions. A distribution with negative excess kurtosis is *sub-Gaussian* or *platykurtic*; a distribution with positive excess kurtosis is *super-Gaussian* or *leptokurtic*. The excess kurtosis of a random variable y is shown in the following equation (2.10):

$$\text{kurt}(y) = \mathbb{E}\left\{y^4\right\} - 3(\mathbb{E}\left\{y^2\right\})^2 \tag{2.10}$$

The absolute value of excess kurtosis has been widely used as a contrast function for ICA, however, its usage has some drawbacks. The main problem with kurtosis is its sensitivity to outliers (Huber, 1985; Hyvärinen and Oja, 2000b) and, as such, kurtosis should not be considered a robust measure of non-Gaussianity (Hyvärinen and Oja, 2000b).

Negentropy approximations

We should select a contrast function that is computationally simple to compute and robust to outliers (Hyvärinen, 1999) that approximate negentropy. Two commonly used for ICA (Hyvärinen, 2001) are as follows:

$$G_1(y) = \frac{1}{a_1} \log \cosh a_1 y \tag{2.11}$$

$$G_2(y) = -\exp(-y^2/2) \tag{2.12}$$

where $a_1 \in [1, 2]$ is a constant. The corresponding derivatives, which will be needed in the algorithm in Section 2.1.4, are the following:

$$g_1(y) = \tanh(a_1 y) \tag{2.13}$$

$$g_2(y) = y \exp(-y^2/2) \tag{2.14}$$

Having built up the required concepts, let us discuss the most popular ICA algorithm (Oja and Yuan, 2006), FastICA.

2.1.4 FastICA

FastICA is based on the maximisation of a contrast function, which contains an approximation of a non-Gaussianity measure from the preceding Section 2.1.3. Algorithm 1, found in (Hyvärinen, Karhunen, and Oja, 2001, p. 196), describes a variant of FastICA that uses symmetric orthogonalisation. Orthogonalising the basis vectors $\mathbf{w}_1$, ..., $\mathbf{w}_m$, which are rows of the matrix $\mathbf{W}$, helps to prevent any of them from discovering the same source component and it is justified when whitening was carried out in the preprocessing step, since then we know that the matrix $\mathbf{W}$ that we are looking for should be a rotation matrix. The algorithm carries out the constrained optimisation from Equations 2.8-2.9 numerically using fixed-point iterations, and one can use either of the functions g from Equations 2.13-2.14. g' is the derivative of g.

The FastICA algorithm was shown to have good converge properties in theory (under suitable assumptions) (Oja and Yuan, 2006).

Algorithm 1 FastICA with symmetric orthogonalisation

```
procedure FastICA
    $\mathbf{x} \leftarrow \mathbf{x} - \mathbb{E}\{\mathbf{x}\}$        ▷ centre the mixtures x to have zero mean
    Whiten the mixtures to give z.
    $m \leftarrow$ number of source signals to estimate.
    for $i := 1$ to $m$ do        ▷ for each of the m sources
        $\mathbf{w}_i \leftarrow$ random unit norm vector.
    end for
    $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W}$        ▷ orthogonalise the matrix W
    while not converged do
        for $i := 1$ to $m$ do
            $\mathbf{w}_i \leftarrow \mathbb{E}\{\mathbf{z}g(\mathbf{w}_i^T\mathbf{z})\} - \mathbb{E}\{g'(\mathbf{w}_i^T\mathbf{z})\}\mathbf{w}$        ▷ update rule, g is the chosen contrast function
        end for
        $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W}$        ▷ orthogonalise the matrix W
    end while
end procedure
```

In practice, however, FastICA is not guaranteed to converge due to a variety of factors, including poor initialisation, numerical issues and characteristics of the data; or it may get stuck in a bad local optimum or saddle-point. In this case, we need to run the algorithm again or consider choosing a different contrast function.

2.1.5 Maximum likelihood ICA

Bell and Sejnowski devised an algorithm based on the principle of maximisation of mutual information between the mixtures and the source estimates (Bell and Sejnowski, 1995), which MacKay showed was equivalent to maximum likelihood estimation (MacKay, 2002). The algorithm, as summarised by MacKay, is shown in Algorithm 2. It maximises the following log likelihood L using gradient ascent:

$$\begin{aligned} L &= \sum_{t=1}^{T} \log p(\mathbf{x}_t) \\ &= \sum_{t=1}^{T} \log p(\mathbf{A}\mathbf{s}_t) \\ &= \sum_{t=1}^{T} \log det(\mathbf{W}) + \log p(\mathbf{s}_t) \\ &= T \log det(\mathbf{W}) + \sum_{t=1}^{T} \sum_{i=1}^{n} f_i(\mathbf{w}_i \mathbf{x}_t) \end{aligned} \tag{2.15}$$

In Equation 2.15, t is time index, i is component index, $\mathbf{W} = \mathbf{A}^{-1}$, and $f_i(\cdot)$ is the probability density function of the i-th source. The last equality is due to the statistical independence assumption of the sources. Ideally, the nonlinearity $\phi(\cdot)$ in the algorithm should have the form f_i'/f_i, as we take small steps in the direction of the negative gradient of the log likelihood. This of course requires knowledge of f_i, which is typically not available in practice, however generic nonlinearities such as the $tanh(\cdot)$ function often work well. Whitening is not necessary for maximum likelihood

ICA, though it can be helpful.

Algorithm 2 Maximum likelihood ICA

```
procedure MLICA
    $\widehat{\mathbf{s}} \leftarrow \mathbf{Wx}$                    ▷ initialise estimated sources $\widehat{\mathbf{s}}$
    while not converged do
        for $i := 1$ to $m$ do                    ▷ for each of the $m$ sources
            $z_i \leftarrow \phi(\widehat{s_i})$                    ▷ apply nonlinearity $\phi$ to each row of $\widehat{\mathbf{s}}$ to get $\mathbf{z}$
        end for
        $\Delta\mathbf{W} \leftarrow [\mathbf{W}^T]^{-1} + \mathbf{z}\mathbf{x}^T$                    ▷ calculate the gradient $\Delta\mathbf{W}$
        $\widehat{\mathbf{s}} \leftarrow \mathbf{Wx}$                    ▷ recalculate estimated sources $\widehat{\mathbf{s}}$
        $\mathbf{W} = \mathbf{W} + \epsilon\Delta\mathbf{W}$                    ▷ update $\mathbf{W}$ by small amount determined by $\epsilon$
    end while
end procedure
```

Like FastICA, Maximum likelihood ICA can also converge to bad local optima or saddle-point for the same reasons. Moreover, it has been observed to converge slower than FastICA in practice (Oja and Yuan, 2006). Note that there is also a version of FastICA that uses maximum likelihood with Newton optimisation updates (Hyvärinen, 1999). In fact, several different ICA approaches were shown to be essentially equivalent (Lee et al., 2000).

2.1.6 ICA research

ICA remains an active and interesting area of research. Although originally developed for the signal processing problem of source separation, ICA has proved useful for a wide range of applications, such as text mining (Bingham, Kabán, and Girolami, 2001), artefact separation for astronomical images (Funaro, Oja, and Valpola, 2003) and analysing financial time series (Oja, Kiviluoto, and Malaroiu, 2000). There has also been much work to use ICA to improve other techniques, such as decision trees (Pajunen and Girolami, 2000) and speech recognition (Zhao et al., 2012). Despite advances in 2014, source separation remains a challenging problem, especially in the underdetermined case where there are more sources than mixtures (Richard, 2014).

2.1.7 Performance of ICA

There are several ways to measure ICA performance. A number of methods are discussed and evaluated in (Vincent, Gribonval, and Fevotte, 2006). A commonly used measure cited in (Vincent, Gribonval, and Fevotte, 2006) that directly compares estimated sources $\widehat{s_i}$ to the true sources s_i is shown below:

$$\mathbf{D} := \min_{\epsilon=\pm 1} \left\| \frac{\widehat{s_i}}{||\widehat{s_i}||} - \epsilon \frac{s_i}{||s_i||} \right\|_2^2 \tag{2.16}$$

The Euclidean norm measures the distance between the true sources s_i and their estimates $\widehat{s_i}$ in a straightforward manner. It benefits from being scale invariant and resistant to sign flips, which is useful since we cannot recover the scaling of the sources. This is the performance measure we use in the original contributions of this thesis, since we have access to both the true sources and their estimates. Since we are not interested in noise or application specific concerns, it is sufficient for our purposes.

Other methods developed include the Blind Audio Source Separation *BASS* set of measures, which is a ratio-based set of measures that compare the estimated sources to different types of noise present (Vincent, Gribonval, and Fevotte, 2006). Crucially, these do not require knowledge of the separation method used. They are implemented in a MATLAB toolbox called *BASS-dB* (Gribonval, Vincent, and Févotte, 2006). Due to time constraints when running the experiments, these performance measures were not used in our experiments.

It should be noted, however, that selecting an appropriate performance measure for ICA is usually highly dependent on the specific application (Mogi and Kasai, 2012) and often requires knowledge about the separation method. Many other methods make unrealistic assumptions about the nature of the sources.

A further study concerning the perceived quality of separations (Emiya et al., 2011) developed a set of objective performance measures that aimed to predict subjective responses. Their objective perceptual score (OPS) measure is commonly used, and there is a MATLAB toolkit implementing these measures called Perceptual Evaluation methods for Audio Source Separation (PEASS). This toolkit aims to predict subjective evaluation of source separation performance, however, which is beyond the scope of this thesis.

2.2 Signal reconstruction and declipping

The name given to the process of reconstructing a signal of which portions are known to be clipped is known as declipping. A broader term is audio inpainting, which describes any audio reconstruction process whereby one observes a set of audio data of which a portion is missing or severely degraded (Adler et al., 2012). A technique commonly used for signal reconstruction is compressed sensing.

2.2.1 Compressed sensing

Compressed sensing is a signal processing technique that offers an efficient solution to feature extraction and dimensionality reduction. The advantage of compressed sensing is that it allows efficient acquisition and reconstruction of a signal from a relatively small number of measurements (Donoho, 2006). It has been used for declipping, and is highly relevant to this thesis. Indeed, a paper by Defraene et al. motivated the idea to use a framework like compressed sensing for this thesis. This paper built upon the framework of compressed sensing to develop a novel declipping method. Given that compressed sensing requires fewer time point measurements than traditional techniques, it is well suited to a task such as declipping. Sparse representations are useful for modelling musical signals, too: by nature, percussive instruments create short impacts and are sparse in the time domain, whereas melodic instruments produce a set of harmonic frequencies and are sparse in the frequency domain (Plumbley et al., 2010).

The compressed sensing framework typically involves three components: a signal model, a measurement process and a reconstruction algorithm. The signal model describes the structure of the signal you want to recover. We assume that the original signal $\mathbf{s} \in \mathbb{R}^N$ (Original Signal) is sparse or compressible in some way, meaning that most of the components are zero, where N is the dimensionality of the signal $\mathbf{s}$. The measurement process involves capturing a small number of linear measurements $\mathbf{y}$ of the signal, which are made by applying a measurement matrix $\boldsymbol{\Psi}$ to the signal, producing $\mathbf{y} = \boldsymbol{\Psi} \cdot \mathbf{s}$ (Measurement). Finally, the reconstruction algorithm attempts to reconstruct the original signal, by finding a sparse signal that satisfies the measurements model $\hat{\mathbf{s}}$ in Equation 2.17. In practice, this can be obtained by solving the following convex optimisation problem in Equation 2.17, and there are efficient implementations, e.g. the ℓ_1-Magic toolbox by Candès (Candès, Romberg, and Tao, 2006).

$$\hat{\mathbf{s}} = \underset{\mathbf{x}}{\operatorname{argmin}} \, ||\mathbf{x}||_1 \text{ subject to } \boldsymbol{\Psi} \cdot \mathbf{x} = \mathbf{y} \tag{2.17}$$

There are, however, challenges with compressed sensing. There is no guarantee of a unique solution, as different solutions may satisfy the time point measurements available; this is because the optimisation problem in Equation 2.17 is convex but not strictly convex. A bigger problem is when the signal is not sufficiently sparse, since compressed sensing assumes that a signal can be represented in a given sparse basis. There is also the problem that compressed sensing is not especially robust to noise.

There are various approaches for compressed sensing. Greedy algorithms, for example, iteratively select components for the signal based on their relevance to the measurements (Blumensath et al., 2012). There are also optimisation-based methods (Qin, 2020), which formulate an optimisation problem to solve, and also Bayesian methods (Ji, Xue, and Carin, 2008), to name a few of the approaches.

2.2.2 Compressed sensing for declipping

Defraene et al. built upon the framework of compressed sensing to develop a novel declipping method (Defraene et al., 2013) that focuses on the audio frequency components that are perceptually important to the human auditory system. We are not interested in the perceptual side for this thesis, but let us briefly explain the approach for completeness.

As already mentioned above, compressed sensing requires that a signal $\mathbf{s}$ is sparse in some fixed basis $\mathbf{\Phi}$ and that we have not directly observed the signal; instead we observe it via some measurement matrix $\mathbf{\Psi}$. This yields the following equation, where $\mathbf{y}$ is our observed signal:

$$\mathbf{y} = \mathbf{\Psi s} = \mathbf{\Psi \Phi x} = \mathbf{Ax} \tag{2.18}$$

In Equation 2.18, $\mathbf{s} \in \mathbb{R}^N$ is the original signal, $\mathbf{x} \in \mathbb{R}^N$ is its sparse decomposition, $\mathbf{\Phi} \in \mathbb{C}^{N \times N}$ is the fixed basis, $\mathbf{\Psi} \in \mathbb{R}^{M \times N}$ is the measurement matrix, $\mathbf{y} \in \mathbb{R}^M$ is the measurement and $\mathbf{A} = \mathbf{\Psi \Phi}$.

The problem is then solved by the following constrained optimisation problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{z}} ||\mathbf{z}||_p \quad \text{subject to} \quad \mathbf{A}\mathbf{z} = \mathbf{y} \tag{2.19}$$

$$\hat{\mathbf{s}} = \mathbf{\Phi}\hat{\mathbf{x}} \tag{2.20}$$

where $||\mathbf{z}||_p = (\sum_{i=1}^{n} |z_i|^p)^{1/p}$. For compressed sensing declipping, decompositions used for basis $\mathbf{\Phi}$ include the Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT) since they both break a signal down into constituent sinusoids: the DCT has the additional virtue of only requiring real-valued calculations, and is the basis we use throughout this work to simplify calculations.

2.2.3 Dictionary learning for sparse representations

Let us introduce the concept of dictionary learning, which plays a crucial role in our discussion on single mixture separation in Section 2.3.3. Dictionary learning is a relevant signal processing technique in the context of signal sparsity and compressed sensing, where it can enhance the overall performance of the framework. To draw a parallel, dictionary learning relates to ICA in its aim to recover a signal **s** as in compressed sensing, while simultaneously learning an unknown matrix $\mathbf{\Phi}$. A fundamental difference, however, lies in the origin of the columns of $\mathbf{\Phi}$, which are adaptively learned from the data rather than being predefined (Schnass, 2015).

The goal of sparse representation is to be able to express an n-dimensional signal **y** as a linear combination of a small number of elements, known as atoms or basis vectors, from a dictionary $\mathbf{\Phi}$ (Tošić and Frossard, 2011). Atoms of the dictionary are typically unit norm functions. Denote the dictionary as $\mathbf{\Phi}$ and its atoms as $\phi_k, k = 1, \ldots, N$, where is N is the size of the dictionary $\mathbf{\Phi}$. A dictionary is said to be overcomplete ($N > n$) when its atoms are linearly dependent and the dictionary covers the entire signal space, i.e. every signal can be represented as a linear combination of atoms in the dictionary, as shown in the following:

$$\mathbf{y} = \mathbf{\Phi s} = \sum_{k=1}^{N} s_k \phi_k \tag{2.21}$$

The problem with overcomplete dictionaries is that our sparse representation **s** is not unique. This is why the sparsity constraint is required. The requirement to find the exact representation is relaxed and approximations are sought instead. The objective is to find a sparse vector **s** containing a small number of non-zero coefficients. This optimisation can be formulated as follows:

$$\min_{\mathbf{s}} ||\mathbf{y} - \mathbf{\Phi s}||_2^2 + \lambda ||\mathbf{s}||_1 \tag{2.22}$$

Approaches to solving the optimisation in Equation 2.22 include convex relaxation methods such as basis pursuit denoising (Chen, Donoho, and Saunders, 2001) or least shrinkage and selection operator (LASSO) (Tibshirani, 1996).

Dictionary learning, however, faces several challenges. One inherent challenge is overfitting. When adapting the dictionary too closely to the training data, there is a risk that the learned basis functions will not generalise well to unseen data (Xie et al., 2014). Finding a balance between fitting the training data and ensuring robustness can be a challenging task. A further challenge is the computational complexity. Especially in high-dimensional data scenarios, such as image processing, dictionary learning can become computationally intensive. The sheer dimensionality of the problem can result in scalability issues, requiring efficient algorithms and techniques. There is also the challenge that many of the problems dictionary learning is used for are non-convex optimisations, making it challenging to find a global minimum. This necessitates the use of sophisticated optimisation techniques.

A key advantage, however, is adaptability. Dictionary learning excels at capturing the underlying structure of data, making it a valuable tool in tasks such as denoising and feature extraction. By adapting the dictionary to the data, it often outperforms traditional fixed dictionaries in representing the data effectively. Compared to using fixed dictionaries, which may be suboptimal for certain types of data, learned dictionaries adapt to the characteristics of the data (Tošić and Frossard, 2011).

2.2.4 Joint declipping and informed source separation

A paper by Bilen, Ozerov, and Pérez claims and appears to be the first to attempt joint declipping and source separation (Bilen, Ozerov, and Pérez, 2015b). The authors note that many signals one may wish to declip are composed of multiple sources and conversely that many source separation tasks may be clipped due to poorly calibrated audio equipment, proposing both tasks may be improved by performing them jointly. They use a non-negative tensor factorisation (NTF) for both the declipping and source separation tasks, but there appears to be no principled justification for doing so in the paper. A weakness of this approach is that the number of instruments and the times when each instrument is playing must be known, so this approach is not blind source separation. There is no temporal alignment of this information with the data, so the usefulness of this method appears to be limited. Moreover, they use the unrealistic assumption that each source is composed of exactly five components in order to determine the rank K for the NTF matrices. Parameter estimation is performed by minimising divergence of the variances of the estimated sources and the NTF model, but no justification for this approach over others is given.

2.3 Alternative source separation approaches

2.3.1 Informed source separation

Since blind source separation remains a challenging problem, there has been much work on so-called informed source separation. This involves obtaining prior information about the sources such as a musical score or manually adding timing annotations for the instruments. An overview of methods in this area was written by Ewert et al. (Ewert et al., 2014), who observed that manually adding annotations to the audio data was laborious and instead focused on musical score informed source separation. Early on, he notes that while blind source separation methods assume statistical independence of the sources, musical sources are often highly correlated in both frequency and time.

Ewert et al. use non-negative matrix factorisation (NMF) source separation to demonstrate the usefulness of this added knowledge. NMF splits the signal into a template matrix W, representing sets of frequency components for each instrument, and an activation matrix H, representing time activations for each instrument component. He observed that fixing parts of both matrices W and H was necessary to achieve a significant improvement in separation performance. The shape of the harmonics of a given instrument is known as its spectral envelope. The knowledge of the notes played by a given instrument along with its spectral envelope can be used to estimate the columns of W. The spectral envelope can be estimated from examples of similar instruments or digital emulations of them.

A significant problem for informed source separation is that, for many musical pieces, no score is available. Another major issue is timing. Quite often, human musical performances will vary in time and other factors. Moreover, scores often do not specify a specific tempo but instead a more general semantic descriptor such as ‘quickly’. Ewert et al. described methods of temporal alignment of the score and the audio data. He notes that chroma-based musical features, which correspond to the musical notes playing, are useful for this task. Harmonic instruments exhibit structures in the time domain: once identified, it is trivial to map these structures directly to the score. The problem of vibrato, where a given frequency oscillates over time, can be ameliorated by leaving a gap above and below the identified frequency components based on the score. Other problems remain, though, such as deviation from the score by repeating a chorus, rearranging and adding additional notes.

2.3.2 Binary masking for underdetermined separation

The underdetermined approach has been solved using binary masking. This involves using sparse representations of the mixtures and the sources. Sparse representations been found to be highly useful in audio signal processing (Plumbley et al., 2010) and can be used to assist various problems, including blind source separation. Moreover, time-frequency representations of a signal, which are commonly sparse, are more stable than time domain representations (Grais and Erdogan, 2013). Musical and speech signals can be made sparse in orthogonal bases such as the wavelet transform or nonorthogonal transforms such as the STFT (Nesbit et al., 2010). Musical recordings typically have structures in a time-frequency representation For example, one would expect percussive instruments to be sparse in the time direction, and harmonic instruments to be sparse in the frequency direction (Ewert et al., 2014).

Sparse component analysis describes source separation techniques that assume the sources can be made sparse in some basis (Plumbley et al., 2010). Let us briefly explain this approach.

It is assumed that the source components s_n have a sparse representation using atoms ϕ_q from a full rank basis matrix $\mathbf{\Phi} \in \mathbb{R}^{Q \times T}$. It is also assumed that $Q = T$. The coefficients of the sparse representation are represented each by z_{nq}.

$$s_n = \sum_{q=1}^{N} z_{nq} \phi_q \tag{2.23}$$

In matrix form, this can be written as:

$$\mathbf{s} = \mathbf{z}\mathbf{\Phi} \tag{2.24}$$

Denote $\mathbf{u} = \mathbf{x}\mathbf{\Phi}^{-1}$ as the representation of $\mathbf{x}$ in the basis $\mathbf{\Phi}$. This is combined with the ICA model as in the following:

$$\mathbf{u} = \mathbf{A}\mathbf{z} \tag{2.25}$$

Disjointness of the sparse sources **s** is assumed, that is, **z** is so sparse that at most one source coefficient can be non-zero at a given transform index q. The previous equation becomes the following:

$$\mathbf{u}_q = \mathbf{a}_{nq} z_{nq} \tag{2.26}$$

Each vector $\mathbf{u}_q$ is a scaled version of one of the columns of the mixing matrix **A**. When **A** is known or discovered, n_q can be estimated for each value of q, this is, whichever n_q is non-zero, by finding the mixing matrix column $\mathbf{a}_j$ that is most correlated with $\mathbf{u}_q$. Note that in the following equation $|\,|$ represents the absolute value and value $||\,||_2$ represents the $\ell 2$ norm:

$$\widehat{n}_q = \arg\max_n \frac{|\mathbf{a}_n^T \mathbf{u}_q|}{||\mathbf{a}_n||_2} \tag{2.27}$$

Now a mask is constructed $\epsilon_{nq} = 1$ if $n = \widehat{n}_q$, otherwise $\epsilon_{nq} = 0$. Using this mask the active sources can be identified, and multiplying Equation 2.26 by $\mathbf{a}_{nq}^T$ and rearranging gives the following:

$$\widehat{z}_{nq} = \epsilon_{nq} \frac{\mathbf{a}_n^T \mathbf{u}_q}{||\mathbf{a}_n||_2} \tag{2.28}$$

Now the sources can be estimated by calculating estimates for the whole matrix $\widehat{\mathbf{z}}$ and using the following equation.

$$\widehat{\mathbf{s}} = \widehat{\mathbf{z}}\mathbf{\Phi}. \tag{2.29}$$

At this point, the sources $\mathbf{s}$ can be trivially recovered by Equation 2.29. This approach is called *binary masking*. When at most one source is active at a given time, the mixing matrix $\mathbf{A}$ can be estimated: a scatter plot of the sparse representations of the mixtures $\mathbf{u}$ would show 'spokes' or in the directions of the columns of $\mathbf{A}$.

Further statistical information is required when applying binary masking to single mixture source separation, since there is no longer a direction $\mathbf{a}_i$ that allows the active source to be determined.

Deep learning

Deep learning techniques have been used to discover the 'ideal' binary mask: that is, the binary mask that, when applied to a given mixture, achieves the best possible separation of the sources (Simpson, 2015; Huang et al., 2014; Grais, Sen, and Erdogan, 2014). This approach combines the masking function with a deep neural network (DNN), which can have many hidden layers, and optimises both (Huang et al., 2014). With this approach, the mask used need not necessarily be binary: so called soft masks, which are not restricted to the one source per coefficient rule, can also be used.

2.3.3 NMF for single mixture separation

NMF has become a successful tool for audio source separation (Grais and Erdogan, 2013; Chien and Hsieh, 2012; Lefèvre, 2012; Ewert et al., 2014), especially for underdetermined and single mixture problems. These approaches separate the sources by factorising the time-frequency representation of the mixture. Some approaches involve learning source-specific dictionaries (Lefèvre, 2012). Many of these approaches benefit from access to additional information about the sources, such as time activations for the instruments, which enable certain matrix values to be fixed as zero

(Ewert et al., 2014; Lefèvre, 2012).

The single mixture separation problem can be modelled as a mixture whose spectrogram is equivalent to an elementwise sum of the spectrogram of each source (Grais and Erdogan, 2013). Using $Y(t, f)$ to refer to the STFT of our single mixture $\mathbf{y}$, where t is the frame index and f is the frequency, the problem for m sources can be modelled as a linear sum of the STFT coefficients of the individual sources, S_i:

$$Y(t, f) = \sum_{i=1}^{m} S_i(t, f) \tag{2.30}$$

Phase angles from the STFT can be ignored: it has been found that the magnitude spectrogram is sufficient for source separation tasks (Helén and Virtanen, 2005; Grais and Erdogan, 2013). We can make use of an approximation of the model in Equation 2.30, which considers only the magnitude spectra and is shown in the equation below:

$$|Y(t, f)| \approx \sum_{i=1}^{m} |S_i(t, f)| \tag{2.31}$$

This can also be written in matrix form, where $\mathbf{Y}$ is the observed magnitude spectrogram of the single mixture and $\mathbf{S}_i$ is the unknown magnitude spectrogram of the ith source:

$$\mathbf{Y} \approx \sum_{i=1}^{m} \mathbf{S}_i \tag{2.32}$$

The NMF algorithm factorises the magnitude spectrogram $\mathbf{Y}$, which is nonnegative, into two nonnegative matrices: a basis matrix $\mathbf{B}$, also called a dictionary, and a gains matrix $\mathbf{G}$, which represents the amount by which each dictionary *atom* is active at a given time.

$$\mathbf{Y} \approx \mathbf{B}\mathbf{G} \tag{2.33}$$

In performing this factorisation, shown in Equation 2.33, the algorithm attempts to minimise some divergence cost function. Commonly used cost functions for source separation include the Kullback-Leibler (KL) and Itakura–Saito (IS) divergences (Grais and Erdogan, 2013). The NMF result for both divergences can be calculated by alternating multiplicative update rules.

Work by Chien and Yang (Chien and Yang, 2016) takes a variational Bayesian NMF approach to source separation, in which the reconstruction error was represented by a Poisson distribution and the nonnegative matrices **G** and **B** were characterised by exponential priors.

2.4 Criticisms of the literature

2.4.1 Nonrobustness of existing ICA methods for clipping distortion

Prior to this research, this author investigated the robustness of existing ICA methods for mixtures affected by clipping distortion (Turl, 2015). Experiments were conducted with three popular ICA algorithms: FastICA (Hyvärinen and Oja, 2000b), the Robust, Accurate, Direct ICA aLgorithm (RADICAL) (Learned-Miller and Fisher III, 2003) and Kernel Density Independent Component Analysis (KDICA) (Chen, 2006). All three approaches were found to be nonrobust for both hard clipped and soft clipped mixtures. A plot of the Overall Perceptual Score (OPS) (Emiya et al., 2011) against the clipping threshold for hard clipping in Figure 2.1. Note that dBFS refers to decibels at full scale: 0 dBFS is the maximum, or full scale, value of the system. When the threshold is at zero, no clipping occurs and the OPS is close to 100, which is the best performance. Once the clipping threshold has dropped to -1 dBFS, which is a very small amount of clipping, the OPS score for all three algorithms has dropped to approximately 20.

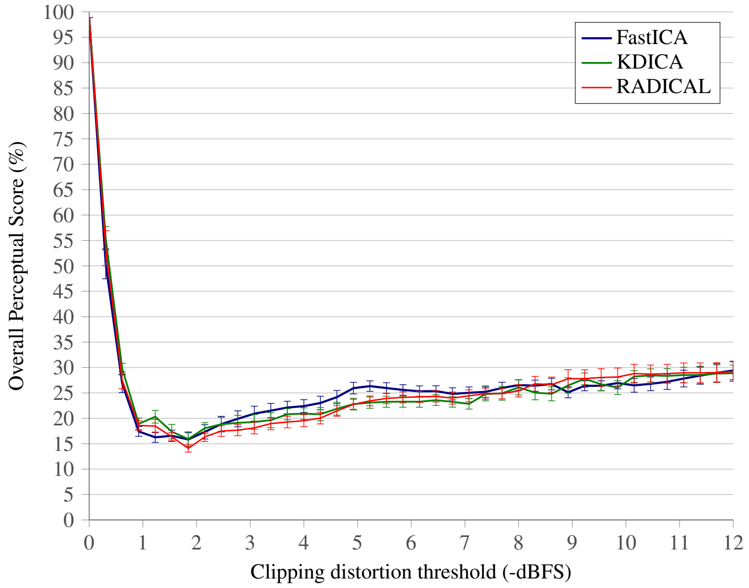


Figure 2.1: OPS versus clipping threshold (hard clipping)

Figure 2.2 shows the same plot for soft clipping by a tanh nonlinearity. That existing ICA algorithms appear more nonrobust at low levels of clipping is not surprising: unlike hard clipping, soft clipping distorts time points before they reach the clipping threshold.

2.4.2 No model for declipping and blind source separation

It is our hypothesis that solving the problems of source separation and repairing clipped mixtures jointly will achieve better performance than solving these tasks sequentially by minimising propagation of reconstruction errors. To date, there appears to be no model that jointly considers the problem of repairing the clipped mixtures, known as *declipping*, and blind source separation.

A paper by Bilen, Ozerov, and Pérez claims and appears to be the first to attempt to solve the declipping problem with *informed* single mixture source separation (Bilen, Ozerov, and Pérez, 2015b). They use a NTF framework for both the declipping and source separation tasks, but there is no principled justification in the paper for taking this approach over others. The problem is

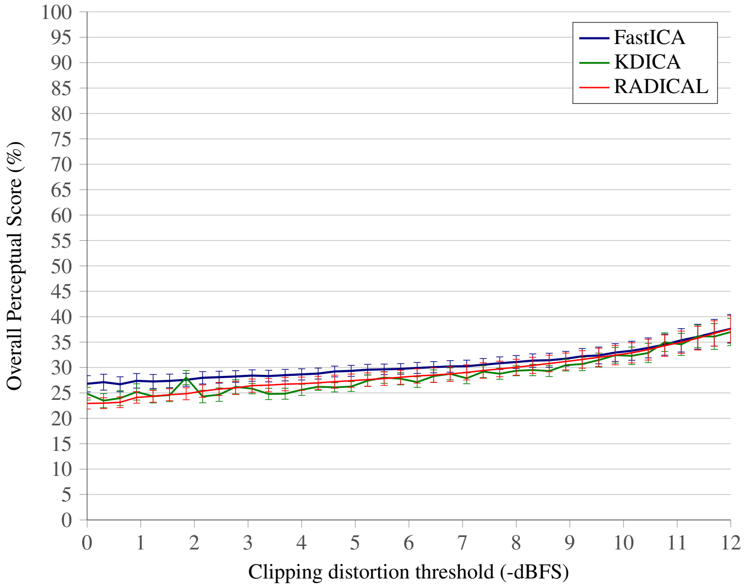


Figure 2.2: OPS versus clipping threshold (soft *tanh* clipping)

treated as partially observed source separation, where the distorted time points are considered to be missing data and the sources are estimated in the minimum mean square error sense. This ignores information known about the discarded clipped time points, crucially that we know their sign and that they should lie beyond the clipping threshold, which is assumed to be known. It is also assumed that there is no noise.

Bilen, Ozerov, and Pérez's algorithm can function either sequentially, first declipping the mixtures and then separating the sources, or jointly performing both tasks. They found better results from the joint approach compared to the sequential approach for severe clipping, but the reverse for very light clipping. Their results, however, do not feature any comparison with any existing source separation approaches. Moreover, their results show no error bars: it appears that their experiments were not repeated and the only comparisons made were between the joint, sequential and source separation only variants of their algorithm. Their results are obtained exclusively from musical mixtures: five different mixtures of three sources were used. In order to determine the rank, K, of the NTF matrices, it is arbitrarily assumed that there are five components for each source in the basis matrix. The joint version of this algorithm features an additional restriction that each five components are used *only* by their assigned source.

A further weakness of this approach is that the number of instruments and the precise times when each instrument is silent must be known, so this approach is not truly blind source separation. Even if such information was obtainable from a musical score, there is no temporal alignment of this information with the data: as such, the usefulness of this method appears to be severely limited. Since the information about when each instrument is silent appears to be crucial to this approach, it seems likely that it would fail to perform when all of the sources were never or rarely silent. Moreover, a diagram showing their experimental setup shows that the sources rarely overlap in time.

Soft clipped mixtures

Declipping usually refers to the process of repairing hard clipped rather than soft clipped signals. Bilen, Ozerov, and Pérez's work would not be appropriate in soft clipping scenarios, since it discards all distorted time points: it is often the case that soft clipping distorts most or all of the time points. There appears to be no work that seeks to solve soft clipping repair and source separation jointly.

2.4.3 No model for inverse DRC and source separation

No model appears to exist that jointly solves the problems of reversing DRC and source separation. This is a much more challenging problem to solve: DRC is a time-varying distortion of the signal with many parameters that may or may not be relevant depending on the compressor used. Given their nonrobustness to time-invariant distortions, it is highly unlikely that existing ICA algorithms would perform well for more complicated time-varying distortions. Mixtures processed by DRC are ubiquitous in music production, and it is our hypothesis that algorithms that exploit knowledge of these nonlinearities would achieve better performance than existing approaches.

Chapter Three

One Clipped Mixture and One Unclipped Mixture of Two Sources

3.1 Introduction

First we deal with a simplified problem, which we can later extend. We assume that we observe two mixtures and that only one suffers from clipping distortion. We also assume that there are the same number of sources as mixtures, the so-called determined case. The work is given in full by including a published paper by this author inline in Section 3.4 and below we present the main ideas.

We make the assumption that the source signals are disjoint in time, meaning at most one source in $\mathbf{s}$ can be non-zero at any given time point. We do this because of an interesting observation in Plumbley et al., 2010. This observation shows that time disjoint sources appear as straight lines in a scattergram of linear mixtures. The following Figure 3.1 shows this clearly. In the figure, s_1 and s_2 are our two sources, which have been generated to be disjoint in time. The mixtures x_1 and x_2 are plot in a scattergram, in which we see the straight lines discernible from the plot. We refer to the clipped version of mixture x_1 as x_1'.

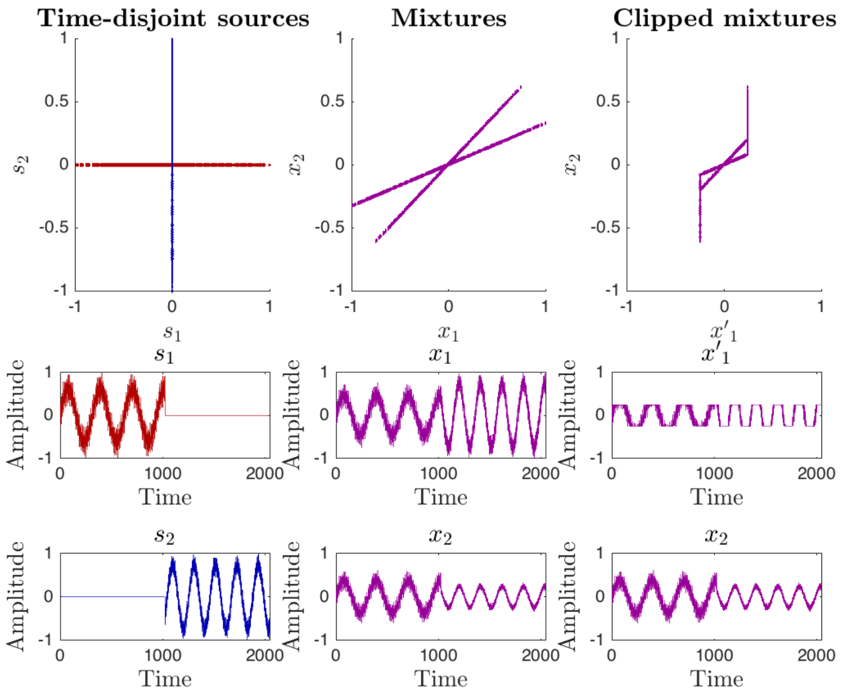


Figure 3.1: Effects of clipping on one of two mixtures.

This representation was exploited for separating more sources than mixtures in Plumbley et al., 2010. As we shall see, this particular representation is key to the novel geometric method described in the paper in this chapter.

A key benefit of the approach taken is that there is no need to assume that the sources are statistically independent as required in ICA. Additionally, we can handle the case of increasing the number of sources and whether or not both mixtures are clipped, which are discussed in Chapter Four.

We start with a simple approach in this chapter, where only one of our mixtures is clipped and we have two sources (D = 2).

3.2 Methodology summary

Throughout this thesis we assume noiseless linear mixtures, as most ICA approaches also assume, which proves to be sufficient for many source separation problems. We observe $F = 2$ mixtures of N time points each, denoted by $\mathbf{x} \in \mathbb{R}^{F \times N}$. Each of these F observations is a linear mixture of D sources, denoted by $\mathbf{s} \in \mathbb{R}^{D \times N}$, obtained by a mixing matrix $\mathbf{A} \in \mathbb{R}^{F \times D}$, as shown in the ICA model equation $\mathbf{x} = \mathbf{A}\mathbf{s}$. Since we assume that one of the mixtures is clipped, let us explain clipping distortion.

Clipping occurs when attempting to represent a signal whose amplitude values saturate a given maximum amplitude of a system, something ICA algorithms neglect to account for. In a digital system, input samples whose absolute value exceeds a given *clipping threshold* θ have an output limited by this threshold. Formally, let us denote by $x_{i,n}$ the nth sample of the ith mixture before clipping and $x'_{i,n}$ the equivalent sample after clipping. We see the clipping equation below:

$$x'_{i,n} = \begin{cases} \operatorname{sgn}(x_{i,n})\,\theta & \text{if } |x_{i,n}| > \theta, \\ x_{i,n} & \text{if } |x_{i,n}| \leq \theta. \end{cases} \tag{3.1}$$

Note that this type of clipping is specifically called hard clipping. There are smooth clipping functions known as soft clipping, which occur in analogue audio hardware, however this is beyond the scope of this thesis.

Now, let us combine the clipping model and the ICA model to fully describe our problem.

Equations (2.1) (from Chapters One and Two) and (3.1) combined give us Equation 3.2:

$$x'_{i,n} = \begin{cases} \operatorname{sgn}\left(\sum_{j=1}^{D} A_{i,j}s_{j,n}\right)\theta & \text{if } |\sum_{j=1}^{D} A_{i,j}s_{j,n}| > \theta, \\ \sum_{j=1}^{D} A_{i,j}s_{j,n} & \text{if } |\sum_{j=1}^{D} A_{i,j}s_{j,n}| \leq \theta. \end{cases} \tag{3.2}$$

Now our task is to find both $\mathbf{s}$ and $\mathbf{A}$, where D denotes the number of sources.

The original approach we put forward develops five sequential steps in order to solve the problem of estimating latent sources $\hat{\mathbf{s}}$ from clipped mixtures $\mathbf{x}'$ as explained briefly below:

1. Mixing matrix estimation (II-B in the inline paper in Section 3.4).
2. Quantisation of repairable time points (II-C in the inline paper in Section 3.4).
3. Declipping mixtures by ℓ_1 minimisation (II-D in the inline paper in Section 3.4).
4. Quantisation of declipped points to the source lines (II-E in the inline paper in Section 3.4).
5. Source estimation (II-F in the inline paper in Section 3.4).

These steps are explained in greater detail in the paper in Section 3.4. In brief, the geometric observations make retrieving the mixing directions and repairing some of the points trivial, and enable geometric constraints to be imposed on the ℓ_1 minimisation (Equation 13 in the inline paper in Section 3.4).

We also implement a sequential approach that, rather treats the declipping and source separation problems sequentially rather than jointly (II-G in the inline paper in Section 3.4. This is done to test our hypothesis that the joint approach is better due to minimisation of propagation of errors.

3.3 Experiment results summary

Experiments are described in III of the inline paper in Section 3.4 along with our hypotheses. III-A of the inline paper in Section 3.4 describes how we prepared the data. III-B of the inline paper describes the implementation, and III-C describes how we measured performance, using the **D** performance measure from Equation 2.16 in Chapter Two. III-D of the inline paper describes and analyses our results.

This published paper presented a novel framework for the separation of sources from mixtures affected by clipping distortion, which combined ℓ_1 minimisation with knowledge about the geometry of clipped mixtures. An algorithm was developed to recover the sources by solving a convex optimisation problem, with constraints derived from the clipping geometry under the assumption that the sources are disjoint in time. Comparative evaluation experiments showed a significant increase in objective recovery performance of our proposed joint method compared to sequential approaches for speech and synthetic signals.

3.4 Paper One

The first published work is reproduced in full after this mostly blank page.

Joint Blind Source Separation and Declipping: A Geometric Approach for Time Disjoint Sources

Alastair Turl
School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
Email: a.c.turl@cs.bham.ac.uk

Ata Kabán
School of Computer Science
University of Birmingham
Birmingham, UK, B15 2TT
Email: a.kaban@cs.bham.ac.uk

Abstract—**Source separation remains a challenging problem, made even more challenging when the mixtures are distorted. We present a novel framework for the source separation from mixtures affected by clipping distortion. Our method combines ℓ_1 minimisation with knowledge about the geometry of clipped mixtures when the sources are disjoint in time. Our algorithm recovers the sources by solving a convex optimisation problem, which is constrained by the clipping geometry. Comparative evaluation experiments show a significant increase in objective recovery performance of our proposed joint method compared to sequential approaches for speech and synthetic signals.**

I. Introduction

Blind source separation (BSS) is the problem of separating a set of source signals from a set of mixture signals. For example, these sources may be distinct voices but we observe microphone recordings of many voices. It is an ill-posed problem, requiring assumptions to solve it. It becomes even more challenging when the observed mixtures have been affected by clipping distortion. Existing independent component analysis (ICA) techniques are not robust to clipping [1].

To make this problem solvable, at least approximately, we shall assume that the sources are disjoint in time, meaning a maximum of one source is non-zero at a given time point. Time disjoint sources appear as straight lines in a scattergram of linear mixtures, as previously noted, and this was exploited for separating more sources than mixtures in [2]. As we shall see, this particular representation is key to the novel geometric method described in this paper.

In our approach, there is no need to assume the sources are statistically independent as required in ICA. In addition we can handle the case of increasing the number of sources and whether or not both mixtures are clipped. For simplicity of exposition, however, this paper details the core, where there are two sources ($D = 2$) and only one of the two mixtures is clipped. Section II describes the methodology, prefaced by a general overview. Section III provides a comprehensive testing of our proposed approach on both synthetic and speech data.

II. Methodology

Throughout this work we assume noiseless linear mixtures. We observe F mixtures of N time points each, denoted by $\mathbf{x} \in \mathbb{R}^{F \times N}$. Each of these F observations is a linear mixture of D sources, denoted by $\mathbf{s} \in \mathbb{R}^{D \times N}$, obtained by a mixing matrix $\mathbf{A} \in \mathbb{R}^{F \times D}$, as shown in the equation below.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

Clipping occurs in practice when attempting to represent a signal that contains amplitude values that saturate the maximum amplitude permitted by a system, yet most ICA algorithms neglect this. In a digital system, input samples whose absolute value exceeds a given *clipping threshold* θ have an output limited by this threshold. Formally, denote by $x_{i,n}$ the nth sample of the ith mixture before clipping and $x'_{i,n}$ the equivalent sample after clipping.

$$x'_{i,n} = \begin{cases} \operatorname{sgn}(x_{i,n})\,\theta & \text{if } |x_{i,n}| > \theta, \\ x_{i,n} & \text{if } |x_{i,n}| \leq \theta. \end{cases} \tag{2}$$

Equations (1) and (2) combined describe the model of the problem:

$$x'_{i,n} = \begin{cases} \operatorname{sgn}\left(\sum_{j=1}^{D} A_{i,j} s_{j,n}\right)\theta & \text{if } \left|\sum_{j=1}^{D} A_{i,j} s_{j,n}\right| > \theta, \\ \sum_{j=1}^{D} A_{i,j} s_{j,n} & \text{if } \left|\sum_{j=1}^{D} A_{i,j} s_{j,n}\right| \leq \theta. \end{cases} \tag{3}$$

Now the task is to find both $\mathbf{s}$ and $\mathbf{A}$.

A. Overview of proposed approach

Linear mixtures of time disjoint sources, where at most one source is non-zero at a given time index, yield straight lines in a scattergram of such mixtures [2] (Figure 1). We refer to these lines as *source lines*. Figure 1 shows an example. The left column shows the sources before mixing and the right column shows the effect of clipping the first mixture. Note that clipping distorts the source lines.

Five sequential steps were developed in order to solve the problem of estimating latent sources $\hat{\mathbf{s}}$ from clipped mixtures $\mathbf{x}'$ as follows:

1) Mixing matrix estimation (Section II-B).
   - The mixing matrix A is represented by the slopes of the source lines.
2) Quantisation of repairable time points (Section II-C).
   - There are some points which can be trivially repaired using knowledge of the clipping geometry.
3) Declipping mixtures by ℓ_1 minimisation (Section II-D).

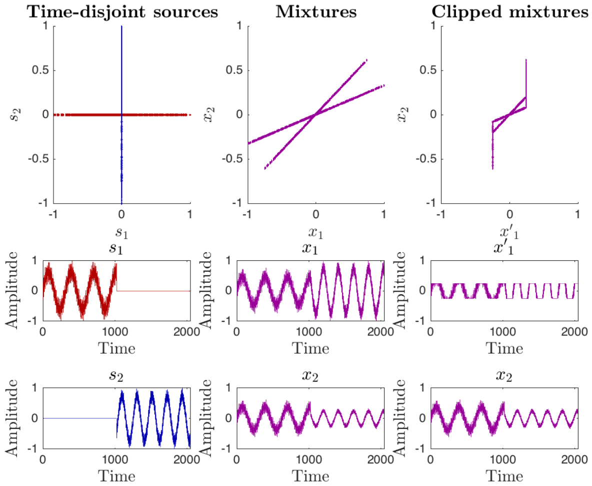


Fig. 1. Effects of clipping on one of two mixtures.

- Declipping by ℓ_1 minimisation allows information about the clipping process and mixture model to constrain the optimisation problem.

4) Quantisation of declipped points to the source lines (Section II-E).
   - The time points must lie on source lines because they are disjoint in time.
5) Source estimation (Section II-F).
   - The sources are recovered by the equation $\hat{\mathbf{s}} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{x}}$.

B. Mixing matrix estimation

Since clipping distorts the source lines, first discard all clipped points, where $\left|x'_{1,n}\right| = \theta$. All remaining time points, $\left|x'_{1,n}\right| \le \theta$, lie on a source line. Since each source line passes through the origin, each of the remaining time points $(x'_{1,n}, x'_{2,n})^T$ has enough information to recover one of the source line slopes, $\nabla_n = (x'_{2,n} - 0)/(x'_{1,n} - 0)$. We denote by M the number of unclipped time points, and calculate the slope of all of these as shown below. The symbol $\odot$ represents the elementwise product.

$$\nabla_{\mathbf{M}} = \left(x'_{2,1}, \ldots, x'_{2,M}\right) \odot \left(\frac{1}{x'_{1,1}}, \ldots, \frac{1}{x'_{1,M}}\right) \tag{4}$$

The set $\nabla_{\mathbf{M}}$ contains M elements, some of which may be duplicates. We define the set $\nabla_{\mathbf{s}}$ to contain only the D unique entries of $\nabla_{\mathbf{M}}$. We make each value of the second row of the mixing matrix $\hat{A}_{2,1} \ldots \hat{A}_{2,D} = 1$, which means the D elements in $\nabla_{\mathbf{s}}$ can be used directly as $\hat{A}_{1,1} \ldots \hat{A}_{1,D}$, the first row of estimated mixing matrix $\hat{\mathbf{A}}$. To remove scaling indeterminacy, we normalise by dividing each element in column i of $\hat{\mathbf{A}}$ by its ℓ_2-norm, $\hat{\mathbf{A}}_{1:2,i} = \hat{\mathbf{A}}_{1:2,i}\frac{1}{\|\hat{\mathbf{A}}_{1:2,i}\|_2}$.

C. Quantisation of repairable time points

Information from the unclipped mixture x'_2 can be used to repair some of the clipped points in x'_1. We split the set of entries of $\nabla_{\mathbf{s}}$ into positives, denoted $\nabla_{\mathbf{s}}^+$, and negatives, denoted $\nabla_{\mathbf{s}}^-$. Denote by ∇_+ the positives in ascending order and by ∇_- the negatives in descending order as follows:

$$\nabla_+ = \left[\frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}}^+ : \frac{x'_{2,i}}{x'_{1,i}} \le \frac{x'_{2,i+1}}{x'_{1,i+1}} \forall i = 1 \ldots \left|\nabla_{\mathbf{s}}^+\right|\right] \tag{5}$$

$$\nabla_- = \left[\frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}}^- : \frac{x'_{2,i}}{x'_{1,i}} \ge \frac{x'_{2,i+1}}{x'_{1,i+1}} \forall i = 1 \ldots \left|\nabla_{\mathbf{s}}^-\right|\right] \tag{6}$$

We define $[\nabla_+]_i$ as the ith element of ∇_+, the list of positive slopes in ascending order, and $[\nabla_-]_i$ as the ith element of ∇_-, the list of negative slopes in descending order. Denote by p_i the value of x'_2 when the source line with positive slope $[\nabla_+]_i$ intersects the clipping threshold ($x'_1 = \theta$). Denote by q_i the same for the source line with negative slope $[\nabla_-]_i$.

$$p_i = [\nabla_+]_i \theta \tag{7}$$

$$q_i = [\nabla_-]_i \theta \tag{8}$$

A time point with index n can be repaired if it satisfies two conditions:

1) It is clipped in x'_1, the first mixture $\left(\left|x'_{1,n}\right| = \theta\right)$.
2) In x'_2, it lies between the intersection points p_1 and p_2 or q_1 and q_2 (the two least steep positive slopes and negative slopes, respectively).

If a time point satisfies both conditions, it must belong to the least steep source line. Figure 2, which shows the mixtures before clipping has occurred, shows why this is true. When we observe clipped data, we know that a point $(x'_{1,n}, x'_{2,n})^T$ on the dashed line between p_1 and p_2 is repairable. We also know that the original point $(x_{1,n}, x_{2,n})^T$ was located on a horizontal straight line, because only the first mixture coordinate $x_{1,n}$ is unknown ($x_{2,n} = x'_{2,n}$). The only source line that intersects the partially bounded region $\left|x_{1,n}\right| \ge \theta \wedge p_1 \le \left|x_{2,n}\right| < p_2$ is the least steep, having slope $[\nabla_+]_1$. This region is shaded in Figure 2.

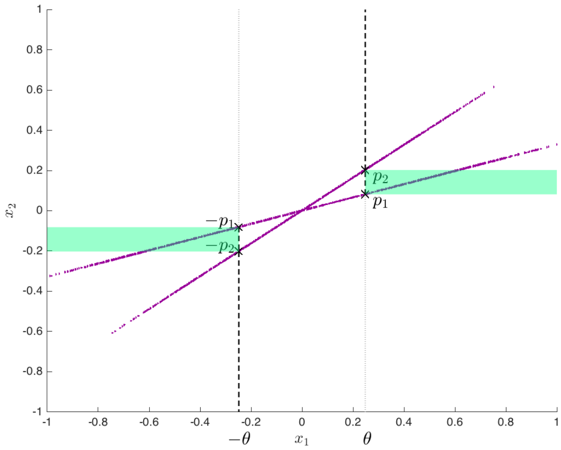


Fig. 2. Possible x_2 values for the first source at clipped points.

With knowledge of one coordinate $x'_{2,n}$, the slope $[\nabla_+]_1$ and that all source lines pass through the origin, we can calculate $x'_{1,n}$ geometrically for points in the region $\left|x_{1,n}\right| \ge \theta \wedge p_1 \le$

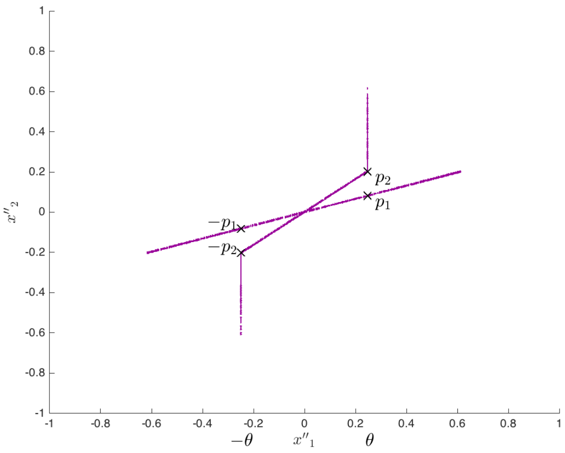


Fig. 3. Clipped mixtures after repairing repairable points.

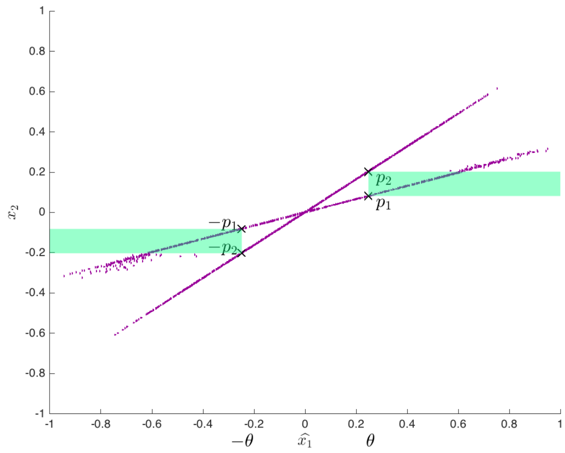


Fig. 4. Clipped mixtures after ℓ_1 mimimisation repair.

$\left|x_{2,n}\right| < p_2$. Note that this also applies to points in the region $\left|x_{1,n}\right| \geq \theta \land q_1 \geq \left|x_{2,n}\right| > q_2$ for reasons of symmetry:

$$x''_{1,n} = \begin{cases} x'_{2,n}/[\nabla_+]_1 & \text{if } \left|x'_{1,n}\right| \geq \theta \land p_1 \leq \left|x'_{2,n}\right| < p_2, \\ x'_{2,n}/[\nabla_-]_1 & \text{if } \left|x'_{1,n}\right| \geq \theta \land q_1 \geq \left|x'_{2,n}\right| > q_2, \\ x'_{1,n} & \text{otherwise.} \end{cases} \tag{9}$$

$$x''_{2,n} = x'_{2,n} \tag{10}$$

Figure 3 shows the result of this repair for our example.

D. Declipping by ℓ1 minimisation

At this stage, we attempt to reconstruct the clipped points that were not repairable in Section II-C. Our method uses a *sparse representation* of the source signals $\mathbf{s}$. Sparse representations have increasingly been used in research in the field of signal processing to solve problems such as declipping [3] and audio inpainting [4]. There are a number of benefits in using sparse representations, namely the ability to represent a signal with a small number of non-zero elements. The flexibility to choose a basis tailored to the type of signal is another advantage. We assume that source signals $\hat{\mathbf{s}}$ can be sufficiently represented by a small number of non-zero coefficients $\mathbf{r}$ in the basis matrix $\mathbf{\Psi}$ like so:

$$\hat{\mathbf{s}} = \mathbf{r}^T \mathbf{\Psi}^T \tag{11}$$

We choose the Discrete Cosine Transform (DCT) operator matrix, commonly used in signal processing, as $\mathbf{\Psi}$ for our example. It is likely that a sparser basis could be chosen, however the DCT will be sufficient to demonstrate our approach. We then formulate a convex optimisation problem that will give us a sparse solution with few non-zero elements. We use ℓ_1 minimisation because the number of non-zero elements, known as the ℓ_0 "norm", is not convex. The task is then to find a solution for the linear system below, where sparse representations $\mathbf{r} \in \mathbb{R}^{N \times D}$:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{D} \|\mathbf{r}_i\|_1 \\ \text{subject to} \quad & \hat{\mathbf{A}} \mathbf{r}^T \mathbf{\Psi}^T = \mathbf{x}''. \end{aligned} \tag{12}$$

The above optimisation problem is constrained by the mixture model $\mathbf{x}'' = \hat{\mathbf{A}}\hat{\mathbf{s}} = \hat{\mathbf{A}}\mathbf{r}^T\mathbf{\Psi}^T$. We know, however, that some time points in x''_1 were unrepairable in Section II-C and should be ignored by this constraint. We instead use a separate constraint for x''_1 and unclipped mixture x''_2. A masking matrix $\mathbf{C_u}$ is constructed, consisting of the $\mathbf{I}_N$ rows corresponding to indices of all unclipped and repairable points in x''_1, which can be used to isolate these points. Similar masking matrices, $\mathbf{C}_+$ and $\mathbf{C}_-$, contain $\mathbf{I}_N$ rows that correspond, respectively, to the positively clipped ($x'_{1,n} \geq \theta$) and negatively clipped ($x'_{1,n} \leq \theta$) points in x''_1. This pair of masking matrices is used to form constraints, similar to those developed in [3], that ensure the sparse sources $\mathbf{r}$ obey the clipping model in Equation (2). The final form of the optimisation problem, where $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$ refer to the first and second rows of the matrix $\hat{\mathbf{A}}$ respectively, is as follows:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{D} \|\mathbf{r}_i\|_1 \\ \text{subject to} \quad & \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_u}^T = x''_1 \mathbf{C_u}^T, \\ & \hat{\mathbf{A}}_2 \mathbf{r}^T \mathbf{\Psi}^T = x''_2, \\ & \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C}_-{}^T \leq -\theta, \\ & \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C}_+{}^T \geq \theta. \end{aligned} \tag{13}$$

After solving the optimisation problem, the following equation yields an estimate of the declipped mixture $\hat{x}_1$:

$$\hat{x}_1 = \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \tag{14}$$

To solve the problem in Equation (13) we used CVX, a package for specifying and solving convex problems [5], [6]. The results are shown in Figure 4.

E. Quantisation of declipped points to the source lines

The mixture points must lie on known source lines, which is not a guaranteed outcome of the optimisation problem. For this reason, we need to quantise the points to the source lines. Since the second mixture x_2 was not clipped, we should only adjust $\hat{x}_1$. In the scattergram, this quantisation translates to horizontal adjustments of the repaired points. Each repaired point is

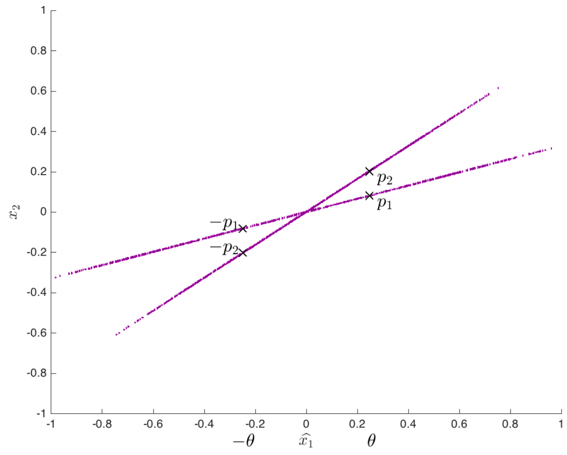


Fig. 5. Quantisation of mixtures repaired by $\ell 1$ mimimisation.

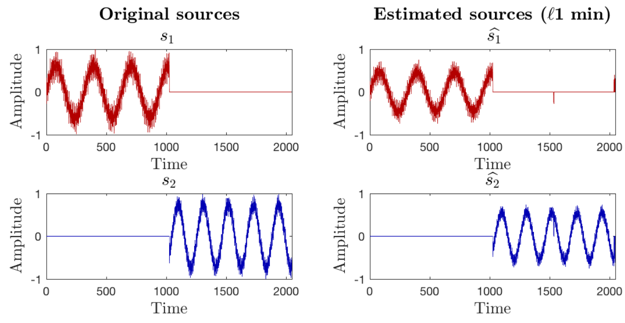


Fig. 6. Sources estimated from clipped mixtures by mixing matrix inverse $\hat{\mathbf{A}}^{-1}$.

quantised to the closest source line i using the following pair of equations:

$$l_n = \arg\min_i \left| \hat{x}_{1,n} - \frac{x_{2,n}}{\nabla_{\mathbf{s}i}} \right| \tag{15}$$

$$\hat{x}_{1,n} = \frac{x_{2,n}}{\nabla_{\mathbf{s}l_n}} \tag{16}$$

The results of quantising in this manner for our example are shown in Figure 5.

F. Source estimation

The final step is the estimation of the sources $\hat{\mathbf{s}}$. Since we have two mixtures, source recovery for two sources is trivial due the square mixing matrix. The estimated sources $\hat{\mathbf{s}}$ can be obtained, using the estimated mixtures $\hat{\mathbf{x}}$ and mixing matrix $\hat{\mathbf{A}}$, by the equation below:

$$\hat{\mathbf{s}} = \hat{\mathbf{A}}^{-1}\hat{\mathbf{x}} \tag{17}$$

Figure 6 shows the sources estimated for our example.

G. Sequential approach

In order to test our novel ideas of using knowledge of the mixture model to quantise repairable points (Section II-C) and inform the optimisation constraints (Section II-D), we omit the former step and modify the latter. The simplified approach is as follows:

1) Mixing matrix estimation (unchanged from Section II-B).
2) Declipping mixtures by ℓ_1 minimisation.
3) Quantisation of declipped points to the source lines (unchanged from Section II-E).
4) Source estimation (unchanged from Section II-F).

The optimisation problem to declip the clipped mixture x_1'' is shown below, where $w \in \mathbb{R}^N$ is a sparse representation of the first mixture:

$$\begin{aligned} \text{minimize} \quad & \|w\|_1 \\ \text{subject to} \quad & w^T\mathbf{\Psi}^T\mathbf{C_u}^T = x_1''\mathbf{C_u}^T, \\ & w^T\mathbf{\Psi}^T\mathbf{C}_-{}^T \leq -\theta, \\ & w^T\mathbf{\Psi}^T\mathbf{C}_+{}^T \geq \theta. \end{aligned} \tag{18}$$

H. Extensions

Our approach can be extended to handle an increasing number of sources ($D > 2$). For this case, the optimisation step (Section II-D) and the source estimation step (Section II-F) are altered. There exist some points for which we know, from the geometry, what the source label is not. This information allows new constraints for the optimisation problem. Source estimation cannot be performed using Equation (17) because the mixing matrix is not square. A different approach is employed, which involves rotating the mixtures to obtain the sources directly.

We can also handle the more challenging case where both mixtures are clipped. For this case, the quantisation of trivial points step (Section II-C) and the optimisation step (Section II-D) are altered. Points clipped in both mixtures cannot be trivially repaired. Points clipped in only one mixture are quantised using information from the unclipped mixture, similar to Section II-C. If only one source line has a positive slope, then we know the source label for any clipped point having a positive slope. The same is true for negative slopes. This knowledge informs additional constraints in the optimisation problem.

III. Experiments

A series of experiments were designed to evaluate our proposed method and two others:

1) The joint source separation and declipping algorithm, as described in Section II.
2) The sequential approach, as described in Section II-G.
3) FastICA [7] - one of the most well known and popular source separation algorithms.

For FastICA, we first declip the mixtures $\mathbf{x}'$ as in Equation (18) from Section II-G. The declipped mixtures are then processed by the FastICA code from [8].

Our hypotheses were that:

1) A larger proportion of clipped samples will lead to worse performance
   - More of the data will need to be approximated by the convex optimisation, increasing the chance of errors.

2) The joint algorithm will outperform the sequential approach in general
   - Peforming the source separation and declipping tasks jointly will avoid the propagation of errors.
3) FastICA will perform worse than our approaches
   - Our approaches exploit knowledge of the clipping geometry and time disjointness assumption, which FastICA does not.
4) Speech sources will yield worse performance than sine wave sources
   - The DCT basis, which consists of sinusoids, will enable a sparser representation of sine wave sources than speech sources.
5) Gaussian sources will yield worse performance than sine wave sources
   - For the same reason given for hypothesis 4.

A. Preparing the data

Three types of source signal are used:

1) *Speech*
   The audio files used are from [9] and available under the Creative Commons Attribution-NonCommercial-ShareAlike 2.0 (http://creativecommons.org/licenses/by-nc-sa/2.0/) license. The authors are Hendrik Kayser and Jorn Anemuller for the original data and Valentin Emiya for the modifications of each file. No changes were made.
   s_1 is the samples from $3900 + 1$ to $3900 + N$ of 'targetSrc.wav'. s_2 is the samples from $3900 + 1$ to $3900 + N$ of 'interfSrc1.wav'.
2) *Synthetic (sine wave mixtures)*
   $s_{1,n} = 3\sin(.02n) + .8\sin(3.5n) + .8\sin(4.1n) + .5\sin(4.5n)$
   $s_{2,n} = 3.4\sin(.03n) + .5\sin(2.2n) + .6\sin(1.4n) + .2\sin(3.7n)$
3) *Synthetic (Gaussian noise)*
   $s_{1,n} \sim \mathcal{N}(0, 0.5)$
   $s_{2,n} \sim \mathcal{N}(0, 0.5)$

Each source file is made disjoint in time in the following manner:
$s_{1,n} = 0 \quad \forall n = \frac{N}{2} + 1 ... N$
$s_{2,n} = 0 \quad \forall n = 1 ... \frac{N}{2}$
The time-disjoint sources are then normalised:
$s_{1,n} = s_{1,n} / \max |s_1|$
$s_{2,n} = s_{2,n} / \max |s_2|$

The mixing matrix is generated $\mathbf{A} \overset{iid}{\sim} \mathcal{U}(-1, 1)$. If $\mathbf{A}$ is singular, it is discarded and generated again. The mixing matrix is scaled $\mathbf{A} = \mathbf{A} / \max |\mathbf{A}|$. The mixtures $\mathbf{x}$ are generated as in Equation 1, clipping only mixture x_1, and scaled $x_{1:2,n} = x_{1:2,n} / \max |\mathbf{x}|$. The mixtures $\mathbf{x}$ are processed as described by Equation (2) to give $\mathbf{x}'$. Five clipping thresholds θ are selected, such that 10%, 20%, 30%, 40% and 50% of the samples of the sample are clipped each time.

B. Implementation and frame-based processing

The method proposed in Section II was implemented in MATLAB version 2016b. The length N of the mixtures and sources was fixed as $N = 2048$. Each experiment was repeated 50 times. For the optimisation problem in Section II-D, it is necessary to process the audio in short distinct frames sequentially and concatenate the results to form the declipped mixtures $\hat{\mathbf{x}}$, as in [3]. This is because of the short-time stationarity of audio signals, but has the added advantage of greatly decreasing processing time by making the basis matrix $\mathbf{\Psi}$ smaller. For our implementation, the mixtures $\mathbf{x}''$ are split into uniform-windowed non-overlapping frames of length $n = 256$.

C. Measuring performance

Recovery performance cannot be measured by evaluating the estimated mixing matrix $\mathbf{A}$ as the estimated sources $\widehat{\mathbf{s}}$ are also dependent on the declipping method used. For this reason, performance is measured by comparing the sources $\mathbf{s}$ and their estimates $\widehat{\mathbf{s}}$, using the following equation, cited in [10]:

$$\mathbf{D} := \min_{\epsilon=\pm 1} \left\| \frac{\widehat{s_i}}{||\widehat{s_i}||} - \epsilon \frac{s_i}{||s_i||} \right\|_2^2 \tag{19}$$

The final problem to solve is the permutation indeterminacy of the sources. We use the Hungarian algorithm, as implemented for MATLAB in [11]. Each source s_i is compared with each source estimate $\widehat{s_j}$ and the results are stored in a $D \times D$ matrix. The algorithm processes this matrix to determine the lowest performance cost and the corresponding estimated source to true source assignments.

D. Results

The following plots show D performance results for sine wave sources (Figure 7), Gaussian sources (Figure 8) and speech sources (Figure 9). Note that smaller is better and the error bar shows the standard error. The number above the error bar shows the number of times FastICA did not converge on a solution.

All of our results showed a trend of worsening performance as the proportion of clipped samples increased. This trend was not apparent in the FastICA approach with speech and sine wave sources. In these cases, however, FastICA performance was very poor at all clipping levels. Indeed, FastICA yielded the worst performance of all three methods in general, as predicted by our hypotheses. Our proposed joint methodology consistently yielded better performance with sine wave sources than the other source types, with very good performance (> 0.1) even when 50% of the data were clipped. This is likely in part due to our use of DCT as the sparse basis, which is known to be a good basis for sinusoid-based signals. Conversely, speech sources yielded worse performance than other source types. It is likely that this performance would be improved by repeating the experiments but choosing a basis better suited to speech signals.

In all of our experiments, our proposed joint approach yielded better performance than the sequential approach. It

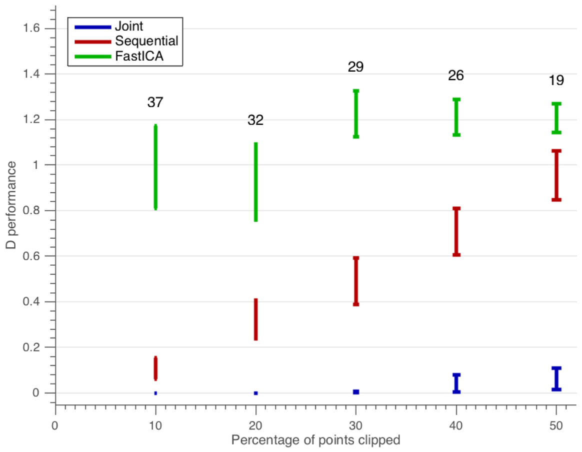


Fig. 7. D performance results for 2 sources, 1 clipped mixture, $n = 256$, $N = 2048$, 50 repetitions (Sine wave sources).

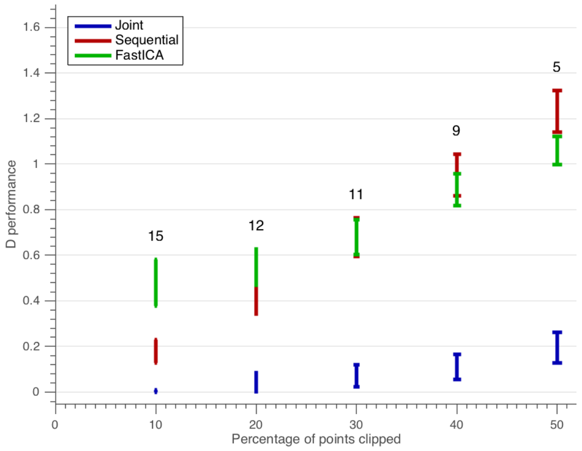


Fig. 8. D performance results for 2 sources, 1 clipped mixture, $n = 256$, $N = 2048$, 50 repetitions (Gaussian sources).

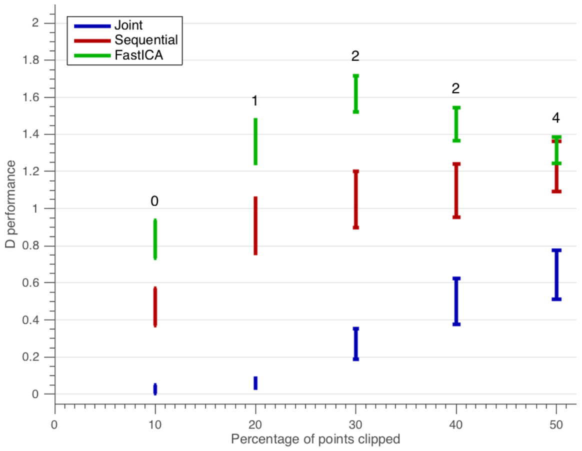


Fig. 9. D performance results for 2 sources, 1 clipped mixture, $n = 256$, $N = 2048$, 50 repetitions (Speech sources).

is likely that performance is improved by our novel idea to constrain the optimisation problem using knowledge of the mixing matrix and the clipped mixture geometry. Another factor is the quantisation of some points as described in Section II-C, which aids the optimisation problem by reducing the number of unknown values. The improved performance is less noticeable only at low clipping levels (10%) with sine wave sources (Figure 7). This is likely because it is the easiest case in our experiments: it has the smallest proportion of unknown values and the advantage that we used the DCT, in which sinusoids are very sparse, as our basis.

IV. Conclusion

This paper has presented a novel framework for the separation of sources from mixtures affected by clipping distortion, which combined ℓ_1 minimisation with knowledge about the geometry of clipped mixtures. An algorithm was developed to recover the sources by solving a convex optimisation problem, with constraints derived from the clipping geometry under the assumption that the sources are disjoint in time. Comparative evaluation experiments showed a significant increase in objective recovery performance of our proposed joint method compared to sequential approaches for speech and synthetic signals.

References

[1] A. C. Turl, "Robustness of ICA methods for clipping-distorted mixtures," MSc thesis, University of Birmingham, 2015.

[2] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse Representations in Audio and Music: From Coding to Source Separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[3] B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, "Declipping of Audio Signals Using Perceptual Compressed Sensing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2627–2637, 2013.

[4] V. Mach, "Inpainting of missing audio signal samples," Ph.D. dissertation, Brno, 2016.

[5] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[6] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110.

[7] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[8] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen, "FastICA for MATLAB." [Online]. Available: http://research.ics.aalto.fi/ica/fastica/

[9] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," no. 19 (7), pp. 2046–2057, 2011.

[10] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, 2006.

[11] Y. Cao, "Hungarian algorithm for linear assignment problems (v2.3)," 2011. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems--v2-3-?

Chapter Four

Two Clipped Mixtures of Many Sources

Now we extend the simplified problem from Chapter Three. We still assume that we observe two mixtures, but that both have a clipping distortion. In addition, we relax the assumption that there are the same number of sources as mixtures and tackle the so-called underdetermined case, where there are more sources than mixtures. The work is presented by including a published work by this author inline in Section 4.3.

Throughout this chapter we still make the assumption that the source signals are disjoint in time, meaning at most one source in $\mathbf{s}$ can be non-zero at any given time point, for the same reason as in Chapter Three.

4.1 Methodology summary

We still assume noiseless linear mixtures, as most ICA approaches also assume. As in Chapter Three, we observe $F = 2$ mixtures of N time points each, denoted by $\mathbf{x} \in \mathbb{R}^{F \times N}$. The key difference of our model here, as can seen in II of the inline paper in Section 4.3, is that now both mixtures are clipped. The model is the same as Equation 3.2 in Chapter Three.

Our task again is to find both **s** and **A**.

The five sequential steps developed in Chapter Three are summarised briefly below:

1. Mixing matrix estimation (Unchanged from Chapter Three. II-B in the inline paper in Section 4.3).
2. Quantisation of repairable time points (II-C in the inline paper in Section 3.4).
3. Declipping mixtures by ℓ_1 minimisation (II-D in the inline paper in Section 3.4).
4. Quantisation of declipped points to the source lines (II-E in the inline paper in Section 3.4).
5. Source estimation (II-F in the inline paper in Section 3.4).

While these sequence of steps are conceptually similar to those we have seen in the previous chapter, there are added complexity due to both clipping and underdeterminedness. These steps are explained in greater detail in the paper in Section 4.3. As in Chapter Three, the geometric observations make retrieving the mixing directions and repairing some of the points trivial. The more complicated geometry with a greater number of sources, however, requires a greater number of constraints to be formulated, and finally the problem is solved as an ℓ_1 minimisation (Equation 10 in the inline paper in Section 4.3).

As in Chapter Three, we also implement a sequential approach that, rather treats the declipping and source separation problems sequentially rather than jointly (II-F in the inline paper in Section 4.3. This is done again to test our hypothesis that the joint approach is better due to minimisation of propagation of errors.

4.2 Experiment results summary

Experiments are described in full in III of the inline paper in Section 4.3. This section includes our hypotheses, how we prepared the data, details about the implementation and how we measured performance, using the **D** performance measure from Equation 2.16 in Chapter Two. III-A of the inline paper describes and analyses our results.

This published paper presented a novel framework for underdetermined source separation from clipped mixtures, which combined ℓ_1 minimisation with knowledge about the clipping geometry. An algorithm was developed to recover the sources by solving a convex optimisation problem constrained by this knowledge. Comparative evaluation experiments of our proposed method showed a significant increase in objective recovery performance compared with sequential approaches for speech and synthetic sources.

4.3 Paper Two

The second published work is reproduced in full after this mostly blank page.

Underdetermined blind source separation for hard clipped stereophonic mixtures

Adahlia Charles
School of Computer Science
University of Birmingham
Birmingham, UK
A.S.Charles@cs.bham.ac.uk

***Abstract*—Underdetermined source separation becomes more challenging when the observed mixtures are distorted. A novel framework for source separation from mixtures affected by clipping distortion is presented. Our method exploits knowledge about the geometry of clipped mixtures when the sources are disjoint in time. This knowledge constrains a convex optimisation problem, which we formulate and solve to recover the sources. This paper develops work previously published substantially by implementing novel preprocessing steps inspired by geometric observations of clipped mixtures of time-disjoint sources. Evaluation experiments showed a significant increase in objective recovery performance for the proposed method compared with sequential approaches for speech and synthetic sources.**

***Index Terms*—underdetermined source separation, declipping, geometric method, convex optimisation**

I. Introduction

The ill-posed problem of blind source separation (BSS) requires the separation of a set of source signals from a set of mixture signals. For example, we say that each source is a guest at a cocktail party and each mixture is a microphone in the venue, which records each guest at a different level depending on factors such as their distance from the microphone. The independent component analysis (ICA) model, which models the sources as unobserved latent variables in a set of observed linear mixtures, is well known and can solve this [1]. Typically, the ICA model is noiseless and has the same number of latent sources as there are mixtures. The underdetermined case, discussed in this paper, considers more latent sources than there are observed mixtures. The underdetermined and ill-posed nature of this problem shares some of the challenges frequently encountered in high dimensional data mining.

The problem becomes even more challenging when all observed mixtures have been affected by clipping distortion. Existing ICA techniques are not robust to clipping [2]. To make this problem solvable, at least approximately, we assume time-disjointness of the sources, meaning a maximum of one source is non-zero at a given time point. Time disjoint sources appear as straight lines in a scattergram of linear mixtures. This fact was exploited for separating more sources than mixtures in [3]. This representation is key to the novel geometric methods described in this paper. Initial work on this topic, which assumed only one of two observed mixtures was clipped, produced promising results [4]. This work extends our approach: we assume two linear mixtures of the sources are observed and that both are clipped.

The model is proposed in Section II, which also details the novel method for underdetermined source separation (Section II-A). Empirical experiments and results are in Section III. In this approach, there is no need to assume the sources are statistically independent as required in ICA.

II. Methodology

Throughout this work we assume noiseless linear mixtures. We observe F mixtures of N time points each, denoted by $\mathbf{x} \in \mathbb{R}^{F \times N}$. Each of these F observations is a linear mixture of D sources, denoted by $\mathbf{s} \in \mathbb{R}^{D \times N}$, obtained by a mixing matrix $\mathbf{A} \in \mathbb{R}^{F \times D}$, as shown in the equation below.

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

Clipping occurs in practice when attempting to represent a signal that contains amplitude values that saturate the maximum amplitude permitted by a system, yet most ICA algorithms neglect this. In a digital system, input samples whose absolute value exceeds a given *clipping threshold* θ have an output limited by this threshold. Formally, denote by $x_{i,n}$ the nth sample of the ith mixture before clipping and $x'_{i,n}$ the equivalent sample after clipping.

$$x'_{i,n} = \begin{cases} \operatorname{sgn}(x_{i,n})\,\theta & \text{if } |x_{i,n}| > \theta, \\ x_{i,n} & \text{if } |x_{i,n}| \le \theta. \end{cases} \tag{2}$$

Equations (1) and (2) combined describe the model of the problem:

$$x'_{i,n} = \begin{cases} \operatorname{sgn}\left(\sum_{j=1}^{D} A_{i,j} s_{j,n}\right)\theta & \text{if } |\sum_{j=1}^{D} A_{i,j} s_{j,n}| > \theta, \\ \sum_{j=1}^{D} A_{i,j} s_{j,n} & \text{if } |\sum_{j=1}^{D} A_{i,j} s_{j,n}| \le \theta. \end{cases} \tag{3}$$

Now the task is to find both $\mathbf{s}$ and $\mathbf{A}$.

A. Overview of the proposed approach

Linear mixtures of time disjoint sources, where at most one source is non-zero at a given time index, yield straight lines in a scattergram of such mixtures [3]. We refer to these lines as source lines, which are visible in Figure 1. The left column shows the sources before mixing and the right column shows the effect of clipping both mixtures. Note how clipping distorts the source lines. Five sequential steps, explained in Sections II-A to II-E, were developed in order to solve the problem of estimating latent sources $\hat{\mathbf{s}}$ from clipped mixtures

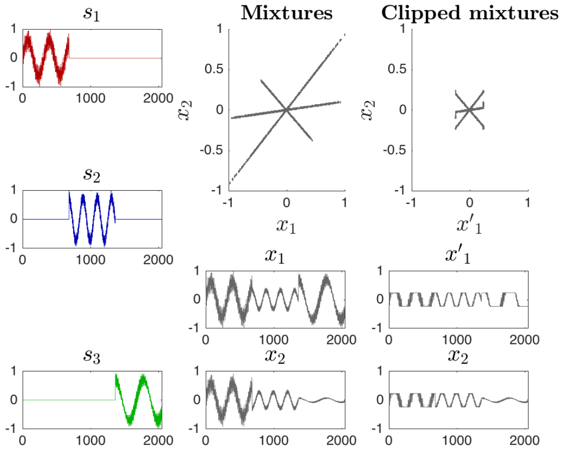


Fig. 1: Effects of clipping both mixtures of time-disjoint sources.

$\mathbf{x}'$. We first discard all clipped points, where $|x'_{1,n}| = \theta$. All remaining time points, $|x'_{1,n}| \le \theta$, lie on a source line. Since each source line passes through the origin, each of the remaining time points $(x'_{1,n}, x'_{2,n})^T$ has enough information to recover one of the source line slopes, $\nabla_n = (x'_{2,n} - 0)/(x'_{1,n} - 0)$. We denote by M the number of unclipped time points, and calculate the slope of all points as shown below. The symbol $\odot$ represents the elementwise product.

$$\nabla_{\mathbf{M}} = \left(x'_{2,1}, \dots, x'_{2,M}\right) \odot \left(1/x'_{1,1}, \dots, 1/x'_{1,M}\right) \tag{4}$$

The set $\nabla_{\mathbf{M}}$ contains M elements, some of which may be duplicates. We define the set $\nabla_{\mathbf{s}}$ to contain only the D unique entries of $\nabla_{\mathbf{M}}$. We make each value of the first row of the mixing matrix $\hat{A}_{1,1} \dots \hat{A}_{1,D} = 1$, which means the D elements in $\nabla_{\mathbf{s}}$ can be used directly as $\hat{A}_{2,1} \dots \hat{A}_{2,D}$, the second row of estimated mixing matrix $\hat{\mathbf{A}}$. To remove scaling indeterminacy, we normalise by dividing each element in column i of $\hat{\mathbf{A}}$ by its l_2-norm, $\hat{\mathbf{A}}_{1:2,i} = \hat{\mathbf{A}}_{1:2,i}/||\hat{\mathbf{A}}_{1:2,i}||_2$.

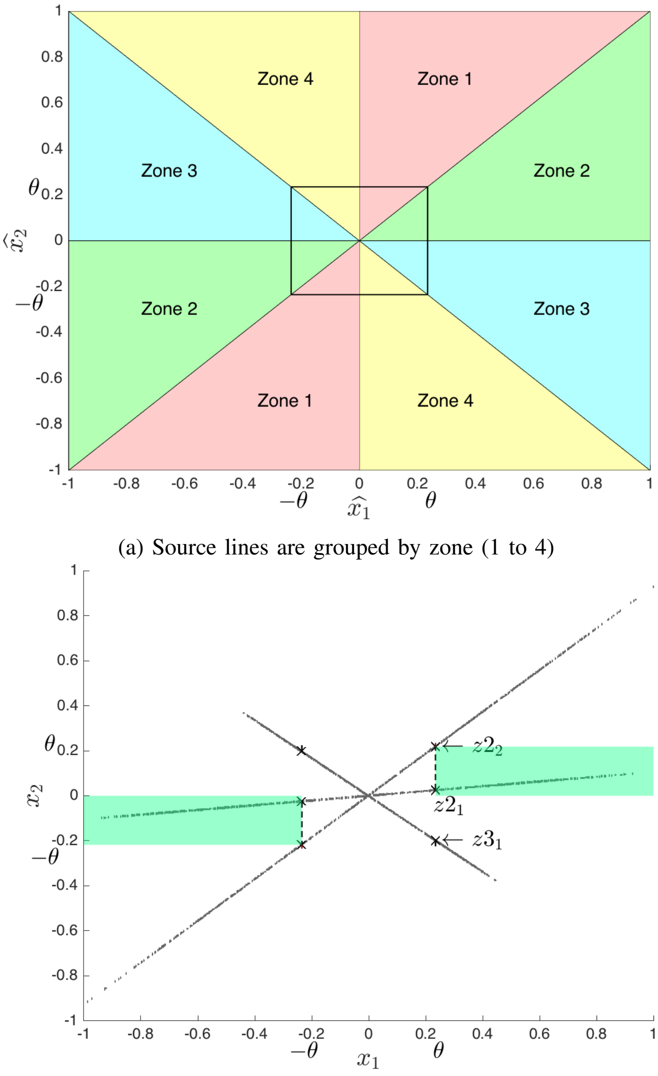
B. *Quantisation of repairable time points*

Points clipped in both mixtures cannot be repaired. Points clipped in only one mixture, however, can be repaired using information from the other mixture. We split the set of entries of $\nabla_{\mathbf{s}}$ into four subsets, denoted $\nabla_{\mathbf{z1}}, \nabla_{\mathbf{z2}}, \nabla_{\mathbf{z3}}$ and $\nabla_{\mathbf{z4}}$, each of which corresponds to one of four geometric *zones*, shown in Figure 2a. These subsets of slopes split into zones are defined by the equations below.

$$\begin{aligned}
\nabla_{\mathbf{z1}} &= \left\{ \frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}} : |x'_{2,i}| > |x'_{1,i}|, \frac{x'_{2,i}}{x'_{1,i}} > 0 \right\} \\
\nabla_{\mathbf{z2}} &= \left\{ \frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}} : |x'_{2,i}| < |x'_{1,i}|, \frac{x'_{2,i}}{x'_{1,i}} > 0 \right\} \\
\nabla_{\mathbf{z3}} &= \left\{ \frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}} : |x'_{2,i}| < |x'_{1,i}|, \frac{x'_{2,i}}{x'_{1,i}} < 0 \right\} \\
\nabla_{\mathbf{z4}} &= \left\{ \frac{x'_{2,i}}{x'_{1,i}} \in \nabla_{\mathbf{s}} : |x'_{2,i}| > |x'_{1,i}|, \frac{x'_{2,i}}{x'_{1,i}} < 0 \right\}
\end{aligned} \tag{5}$$

Denote by $\nabla_{\mathbf{z}}^{\mathbf{1}}$ and $\nabla_{\mathbf{z}}^{\mathbf{3}}$ the entries of $\nabla_{\mathbf{z1}}$ and $\nabla_{\mathbf{z3}}$, respectively, in descending order. Similarly, denote by $\nabla_{\mathbf{z}}^{\mathbf{2}}$ and $\nabla_{\mathbf{z}}^{\mathbf{4}}$ the entries of $\nabla_{\mathbf{z2}}$ and $\nabla_{\mathbf{z4}}$ in ascending order. This ensures slopes are ordered by proximity to whichever axis forms the zone boundary. Denote by $z1_i$ and $z4_i$ the values of x'_1 when the source lines with slopes $[\nabla_{\mathbf{z}}^{\mathbf{1}}]_i$ and $[\nabla_{\mathbf{z}}^{\mathbf{4}}]_i$, respectively, intersect the clipping threshold in the other mixture ($x'_2 = \theta$), calculated by $\theta/[\nabla_{\mathbf{z}}^{\mathbf{1}}]_i$ and $\theta/[\nabla_{\mathbf{z}}^{\mathbf{4}}]_i$. Denote by $z2_i$ and $z3_i$ the values of x'_2 when the sources lines with slopes $[\nabla_{\mathbf{z}}^{\mathbf{2}}]_i$ and $[\nabla_{\mathbf{z}}^{\mathbf{3}}]_i$, respectively, intersect the clipping threshold in the other mixture ($x'_1 = \theta$), calculated by $[\nabla_{\mathbf{z}}^{\mathbf{2}}]_i\theta$ and $[\nabla_{\mathbf{z}}^{\mathbf{3}}]_i\theta$. A time point with index n can be repaired if it satisfies two conditions:

1) It is clipped in x'_1, the first mixture $\left(|x'_{1,n}| = \theta\right)$, or x'_2, the second mixture $\left(|x'_{2,n}| = \theta\right)$, but not both.
2) If clipped in x'_1, the x'_2 coefficient lies between the intersection points $z2_1$ and $z2_2$ or $z3_1$ and $z3_2$ (the two least steep zone 2 slopes and zone 3 slopes, respectively). If clipped in x'_2, the x'_1 coefficient lies



(a) Source lines are grouped by zone (1 to 4)

(b) Shaded region contains only the least steep Zone 2 source line.

Fig. 2: Zone map (top), shaded region used in step 2 and 3 (bottom).

between the intersection points $z1_1$ and $z1_2$ or $z4_1$ and $z4_2$ (the two most steep zone 1 slopes and zone 4 slopes, respectively).

If a time point satisfies both conditions, it must belong to the source line nearest to whichever axis forms part of the zone boundary. Figure 2b, which shows the mixtures before clipping, illustrates why this is true. All points in the shaded region $|x_{1,n}| \geq \theta \wedge |z2_1| \leq |x_{2,n}| < |z2_2| \wedge x_{1,n}x_{2,n} > 0$ will still lie between $z2_1$ and $z2_2$ or $-z2_1$ and $-z2_2$ in mixture x'_2, which is unclipped for such points. The only line contained in this region, by definition, is the one with slope $[\nabla_{\mathbf{z}}^{2}]_1$. Therefore, all clipped points in the range $|x'_{1,n}| = \theta \wedge |z2_1| \leq |x'_{2,n}| < |z2_2| \wedge x'_{1,n}x'_{2,n} > 0$ must have belonged to this line. With knowledge of one coordinate $x'_{2,n}$, the slope $[\nabla_{\mathbf{z}}^{2}]_1$ and that all source lines pass through the origin, we can repair $x'_{1,n}$ geometrically for points in the range $|x'_{1,n}| = \theta \wedge |z2_1| \leq |x'_{2,n}| < |z2_2| \wedge x'_{1,n}x'_{2,n} > 0$. For reasons of symmetry, the same applies to zone 3 points in the range $|x'_{1,n}| = \theta \wedge |z3_1| \leq |x'_{2,n}| < |z3_2| \wedge x'_{1,n}x'_{2,n} < 0$ using the slope $[\nabla_{\mathbf{z}}^{3}]_1$:

$$x''_{1,n} = \begin{cases} x'_{2,n}/[\nabla_{\mathbf{z}}^{2}]_1 & \text{if } |x'_{1,n}| = \theta \wedge |z2_1| \leq |x'_{2,n}| < |z2_2| \wedge x'_{1,n}x'_{2,n} > 0, \\ x'_{2,n}/[\nabla_{\mathbf{z}}^{3}]_1 & \text{if } |x'_{1,n}| = \theta \wedge |z3_1| \leq |x'_{2,n}| < |z3_2| \wedge x'_{1,n}x'_{2,n} < 0, \\ x'_{1,n} & \text{otherwise.} \end{cases} \tag{6}$$

The same idea applies to the remaining zones 1 and 4 and, if one swaps the axes, the symmetry of the zones is apparent. With knowledge of one coordinate $x'_{1,n}$ and the slope $[\nabla_{\mathbf{z}}^{1}]_1$, we can repair $x'_{2,n}$ for points in the range $|x'_{2,n}| = \theta \wedge |z1_1| \leq |x'_{1,n}| < |z1_2| \wedge x'_{1,n}x'_{2,n} > 0$. For reasons of symmetry, the same applies to zone 4 points in the range $|x'_{2,n}| = \theta \wedge |z4_1| \leq |x'_{1,n}| < |z4_2| \wedge x'_{1,n}x'_{2,n} < 0$ using the slope $[\nabla_{\mathbf{z}}^{4}]_1$:

$$x''_{2,n} = \begin{cases} x'_{1,n}[\nabla_{\mathbf{z}}^{1}]_1 & \text{if } |x'_{2,n}| = \theta \wedge |z1_1| \leq |x'_{1,n}| < |z1_2| \wedge x'_{1,n}x'_{2,n} > 0, \\ x'_{1,n}[\nabla_{\mathbf{z}}^{4}]_1 & \text{if } |x'_{2,n}| = \theta \wedge |z4_1| \leq |x'_{1,n}| < |z4_2| \wedge x'_{1,n}x'_{2,n} < 0, \\ x'_{2,n} & \text{otherwise.} \end{cases} \tag{7}$$

C. Declipping mixtures by ℓ_1 minimisation

We reconstruct the unrepairable clipped points by assuming they have a *sparse representation*. We assume that estimated sources $\hat{\mathbf{s}}$ can be sufficiently represented by a small number of non-zero coefficients $\mathbf{r}$ in the basis matrix $\mathbf{\Psi}$, as shown below:

$$\hat{\mathbf{s}} = \mathbf{r}^T \mathbf{\Psi}^T \tag{8}$$

We use the Discrete Cosine Transform (DCT) operator matrix, commonly used in signal processing, as $\mathbf{\Psi}$ for our example. It is likely that a sparser basis could be chosen, however the DCT will be sufficient to demonstrate our approach. We then formulate a convex optimisation problem that will give us a sparse solution with few non-zero elements. We use ℓ_1 minimisation because the number of non-zero elements, known as the ℓ_0 "norm", is not convex. The task is then to find a solution for the linear system below, where sparse representations $\mathbf{r} \in \mathbb{R}^{N \times D}$ and $\mathbf{x}''$ is the clipped mixtures $\mathbf{x}'$ with repairable points repaired:

$$\text{minimize} \quad \sum_{i=1}^{D} \|\mathbf{r}_i\|_1 \quad \text{subject to} \quad \hat{\mathbf{A}}\mathbf{r}^T\mathbf{\Psi}^T = \mathbf{x}''. \tag{9}$$

We constrain this by the mixture model $\mathbf{x}'' = \hat{\mathbf{A}}\hat{\mathbf{s}} = \hat{\mathbf{A}}\mathbf{r}^T\mathbf{\Psi}^T$. We formulate a constraint each for the unclipped and repairable portions x''_1 and x''_2. A pair of masking matrices $\mathbf{C_{u1}}$ and $\mathbf{C_{u2}}$ are constructed, consisting of the $\mathbf{I}_N$ rows corresponding to indices of all unclipped and repairable points in x''_1 and x''_2 respectively. These matrices are used to form the first pair of constraints in Equation (10) below.

We also add constraints that ensure the clipping model in Equation (2) is obeyed, similar to those in [5]. Masking matrices $\mathbf{C_{+1}}$ and $\mathbf{C_{-1}}$ contain $\mathbf{I}_N$ rows that correspond to the indices of unrepairable points positively clipped ($x''_{1,n} = \theta$) and negatively clipped ($x''_{1,n} = -\theta$), respectively, in x''_1. For points clipped in the second mixture, x''_2, the equivalent matrices $\mathbf{C_{+2}}$ and $\mathbf{C_{-2}}$ are constructed. These four matrices enable clipped points to be constrained, forming the second and third pairs of constraints in Equation (10).

The problem can also be constrained by knowledge about what the source label is not. It is helpful now to refer back to Figure 2b in Section II-B. Extending the upper bound of this region from $z2_2$ to $z2_i$ encompasses an additional $2 - i$ source lines, the steepest of which has the slope $[\nabla_{\mathbf{z}}^{2}]_{i-1}$. Therefore, $[\nabla_{\mathbf{z}}^{2}]_{i-1}$ is the steepest slope that points in the range $|x''_{1,n}| = \theta \wedge |z2_1| \leq |x''_{2,n}| < |z2_i| \wedge x''_{1,n}x''_{2,n} > 0$ could have had. At these points, sources with steeper slopes ($> [\nabla_{\mathbf{z}}^{2}]_{i-1}$) must be zero, as must sources insufficiently steep to belong to zone 2 ($< [\nabla_{\mathbf{z}}^{2}]_1$). Denote by $\mathbf{B_{z2i}}$ a matrix of $\mathbf{I}_D$ rows corresponding to the labels of the source lines too steep ($\arg_i \hat{\mathbf{A}}_{2,i}/\hat{\mathbf{A}}_{1,i} > [\nabla_{\mathbf{z}}^{2}]_{i-1}$) or insufficiently steep ($\arg_i \hat{\mathbf{A}}_{2,i}/\hat{\mathbf{A}}_{1,i} < [\nabla_{\mathbf{z}}^{2}]_1$) to belong to zone 2. Denote by $\mathbf{C_{z2i}}$ a matrix of $\mathbf{I}_N$ rows corresponding to indices of those zone 2 clipped points in the range $|x''_{1,n}| = \theta \wedge |z2_1| \leq |x''_{2,n}| < |z2_i| \wedge x''_{1,n}x''_{2,n} > 0$. There is an exception when $i = 1$, in which case these matrices are more general and used to ensure zone 1, 3 and 4 sources are zero for clipped points in zone 2. Denote by $\mathbf{B_{z21}}$ a matrix of $\mathbf{I}_D$ rows corresponding to the labels of all source lines not in zone 2. Denote by $\mathbf{C_{z21}}$ a matrix of $\mathbf{I}_N$ rows corresponding to indices of all clipped points in zone 2. Equivalent matrices created for the other zones are almost identical, for reasons of symmetry, and a more thorough explanation is omitted for brevity. These masking matrices form the third set of constraints in Equation (10), which exploit this partial knowledge of the source labels.

We can also constrain points clipped in both mixtures. Since we assume noiseless mixing and time disjoint sources, we know the sign of the slope of a point will not be altered by clipping. Each estimated source $\hat{s}_i$ which has a positive slope when mixed ($\hat{\mathbf{A}}_{2,i}/\hat{\mathbf{A}}_{1,i} > 0$) must be zero at those indices n where the clipped mixtures $\mathbf{x}''$ yield a negative slope ($x''_{2,n}/x''_{1,n} < 0$). At these indices, we know the slope is negative from the mixtures, so the sources whose lines have a positive slope must be zero because of the time disjointness. Denote by $\mathbf{B_p}$ a masking matrix consisting of $\mathbf{I}_D$ rows that correspond to the indices of the rows of $\hat{\mathbf{A}}$ that yield a source line with a positive slope ($\hat{\mathbf{A}}_{2,i}/\hat{\mathbf{A}}_{1,i} > 0$). An equivalent masking matrix $\mathbf{B_q}$ is constructed to isolate source lines with a negative slope ($\hat{\mathbf{A}}_{2,i}/\hat{\mathbf{A}}_{1,i} < 0$). We denote by $\mathbf{C_{bp}}$ a masking matrix consisting of $\mathbf{I}_N$ rows that correspond to the indices of points clipped in both x''_1 and x''_2 that have a positive slope ($x''_{2,n}/x''_{1,n} > 0$). An equivalent masking matrix $\mathbf{C_{bq}}$ is constructed to isolate points clipped in both mixtures that have a negative slope ($x''_{2,n}/x''_{1,n} < 0$). These four masking matrices are used to form the final pair of constraints in Equation (10). The final form, where $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$ refer to the first and second rows of $\hat{\mathbf{A}}$ respectively, is as follows:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{D} \|\mathbf{r}_i\|_1 \\
\text{subject to} \quad & \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{u1}}^T = x''_1 \mathbf{C_{u1}}^T, \\
& \hat{\mathbf{A}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{u2}}^T = x''_2 \mathbf{C_{u2}}^T, \\
& \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{+1}}^T \geq \theta, \\
& \hat{\mathbf{A}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{-1}}^T \leq -\theta, \\
& \hat{\mathbf{A}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{+2}}^T \geq \theta, \\
& \hat{\mathbf{A}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{-2}}^T \leq -\theta, \\
& \mathbf{B_{z1i}} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{z1i}}^T = 0 \quad \forall i = 1 ... |\nabla_{\mathbf{z1}}|, \\
& \mathbf{B_{z2i}} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{z2i}}^T = 0 \quad \forall i = 1 ... |\nabla_{\mathbf{z2}}|, \\
& \mathbf{B_{z3i}} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{z3i}}^T = 0 \quad \forall i = 1 ... |\nabla_{\mathbf{z3}}|, \\
& \mathbf{B_{z4i}} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{z4i}}^T = 0 \quad \forall i = 1 ... |\nabla_{\mathbf{z4}}|, \\
& \mathbf{B_p} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{bq}}^T = 0, \\
& \mathbf{B_q} \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{bp}}^T = 0.
\end{aligned} \tag{10}$$

CVX, a package for specifying and solving convex problems [6], [7] was used to solve Equation (10). After obtaining $\mathbf{r}$, declipped mixtures $\hat{\mathbf{x}}$ are estimated as below:

$$\hat{\mathbf{x}} = \hat{\mathbf{A}} \mathbf{r}^T \mathbf{\Psi}^T \tag{11}$$

D. Quantisation of declipped points

Each point is given a source label by choosing whichever slope is nearest, as shown in Equation (12). Those points clipped in only one mixture are quantised to the source line given by the label l, as shown in Equation (13). Those points clipped in both mixtures cannot be quantised using information from either mixture, so they are left unaltered. For the source estimation step, however, it is not necessary for all points to lie exactly on a source line.

$$l_n = \arg\min_i \left| (\hat{x}_{2,n}/\hat{x}_{1,n}) - \nabla_{\mathbf{s}i} \right| \tag{12}$$

$$\begin{aligned}
\hat{x}_{1,n} &= x'_{2,n}/\nabla_{\mathbf{s}l_n} \quad \text{if } |x'_{1,n}| \geq \theta \wedge |x'_{2,n}| < \theta \\
\hat{x}_{2,n} &= x'_{1,n}\nabla_{\mathbf{s}l_n} \quad \text{if } |x'_{2,n}| \geq \theta \wedge |x'_{1,n}| < \theta
\end{aligned} \tag{13}$$

E. Source estimation

We estimate each coefficient $\hat{s}_{i,n}$ as a weighted sum of mixture coefficients, $\hat{x}_{1,n} + \nabla_{\mathbf{s}i}\hat{x}_{2,n}$ if the known source label l_n from Equation (12) is i. If the label does not match, we set the coefficient to zero because the sources are disjoint in time. The equation below shows this method.

$$\hat{s}_{i,n} = \begin{cases} \hat{x}_{1,n} + \nabla_{\mathbf{s}l_n}\hat{x}_{2,n} & \text{if } l_n = i, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

F. Sequential approach

In order to test our novel ideas of using knowledge of the mixture model to quantise repairable points (Section II-B) and inform the optimisation constraints (Section II-C), we omit the former step and modify the latter. The simplified approach is as follows:

1) Mixing matrix estimation (unchanged from Section II-A).
2) Declipping mixtures by $\ell 1$ minimisation.
3) Quantisation of declipped points to the source lines (unchanged from Section II-D).
4) Source estimation (unchanged from Section II-E).

The optimisation problem to declip each clipped mixture x''_i is shown below, where $w_i \in \mathbb{R}^N$ is a sparse representation of the ith mixture:

$$\begin{aligned}
\text{minimize} \quad & \|w_i\|_1 \quad \forall i = 1...F \\
\text{subject to} \quad & {w_i}^T \mathbf{\Psi}^T \mathbf{C_{ui}}^T = \\
& x''_i \mathbf{C_{ui}}^T, {w_i}^T \mathbf{\Psi}^T \mathbf{C_{-i}}^T \leq -\theta, {w_i}^T \mathbf{\Psi}^T \mathbf{C_{+i}}^T \geq \theta.
\end{aligned} \tag{15}$$

III. Experiments

A series of experiments were designed to evaluate our proposed methods and two others:

1) The joint source separation and declipping algorithm as described in Section II.
2) The sequential version, described in Section II-F.
3) A state of the art beamformer algorithm [8], [9].

For the beamformer algorithm, we first declip the mixtures $\mathbf{x}'$ as in Equation (15) from Section II-F. The declipped mixtures are then processed by the MMSEbfm code from [9].

Three types of source signal are used: sine wave mixtures, Gaussian noise and speech. The mixing matrix is generated $\mathbf{A} \overset{iid}{\sim} \mathcal{U}(-1, 1)$. If $\mathbf{A}$ is singular, it is discarded and generated again. The mixing matrix is scaled $\mathbf{A} = \mathbf{A}/\max|\mathbf{A}|$. The mixtures $\mathbf{x}$ are generated as in Equation 1 and processed as described by Equation (2) to give $\mathbf{x}'$. Five clipping thresholds θ are selected, such that 10%, 20%, 30%, 40% and 50% of the samples of the sample are clipped each time. The length N of the mixtures and sources was fixed as $N = 2048$. Each experiment was repeated 50 times.

For the optimisation in Section II-C, it is necessary to process the audio sequentially in short frames and concatenate them to form the declipped mixtures $\hat{\mathbf{x}}$, as in [5]. This greatly decreases processing time by making the matrix $\mathbf{\Psi}$ smaller. For our implementation, the mixtures $\mathbf{x}''$ are split into uniform-windowed non-overlapping frames of length $n = 256$.

Recovery performance cannot be measured by evaluating the estimated mixing matrix $\mathbf{A}$ as the estimated sources $\widehat{\mathbf{s}}$ are also dependent on the declipping method used. For this reason, performance is measured by comparing the sources $\mathbf{s}$ and their estimates $\widehat{\mathbf{s}}$, using the following equation, cited in [10]:

$$\mathbf{D} := \min_{\epsilon = \pm 1} \left\| \frac{\widehat{s_i}}{\|\widehat{s_i}\|} - \epsilon \frac{s_i}{\|s_i\|} \right\|_2^2 \tag{16}$$

The final problem is the permutation indeterminacy of the sources. We use the Hungarian algorithm, as implemented in [11]. Each source s_i is compared with each source estimate $\widehat{s_j}$ and the results are stored in a $D \times D$ matrix. The algorithm processes this matrix to determine the lowest performance cost and the corresponding estimated source to true source assignments.

A. Results

The following plots show **D** performance results for sine wave sources (Figures 3a and 4a), Gaussian sources (Figures 3b and 4b) and speech sources (Figures 3c and 4c) for our experiments with 3 sources (Figure 3) and 10 sources (Figure 4). For **D** performance, smaller is better. The error bar shows the standard error. It should be noted that performance scores cannot be directly compared between 3 and 10 sources because the **D** score grows with the number of sources.

Our proposed joint method performed consistently better than the other two methods, yielding significantly lower performance scores in every experiment. Performance was likely improved by the application of clipping geometry knowledge to constrain the optimisation. The best performance was attained with sine wave sources, as we expected: in the chosen DCT basis, which consists of sinusoids, it is likely that a sparser representation exists for our synthetic sine wave sources than the other types. In general, speech performed a little better than the Gaussian noise, however, it is likely that speech performance could be improved by choosing a different basis better suited to speech.

The constraints that exploit knowledge about what the source label is not (in Section II-C) appear to be crucial with 10 sources. Indeed, preliminary experiments omitted these constraints and performance with 10 sources was similar to the sequential method. For 3 sources, omitting these constraints did not appear to affect the results, which still yielded lower performance for our joint method, indicating that these improvements were due to the quantisation of repairable points (Section II-B). This makes sense: with fewer sources, it is more likely the angle between the source lines will be large and, as a result, there will be more repairable points. With more sources, this becomes less likely and the information about what the source label is not becomes more important. Performance in general decreased at higher levels of clipping, as expected, because more of the data points were missing.

IV. Conclusions

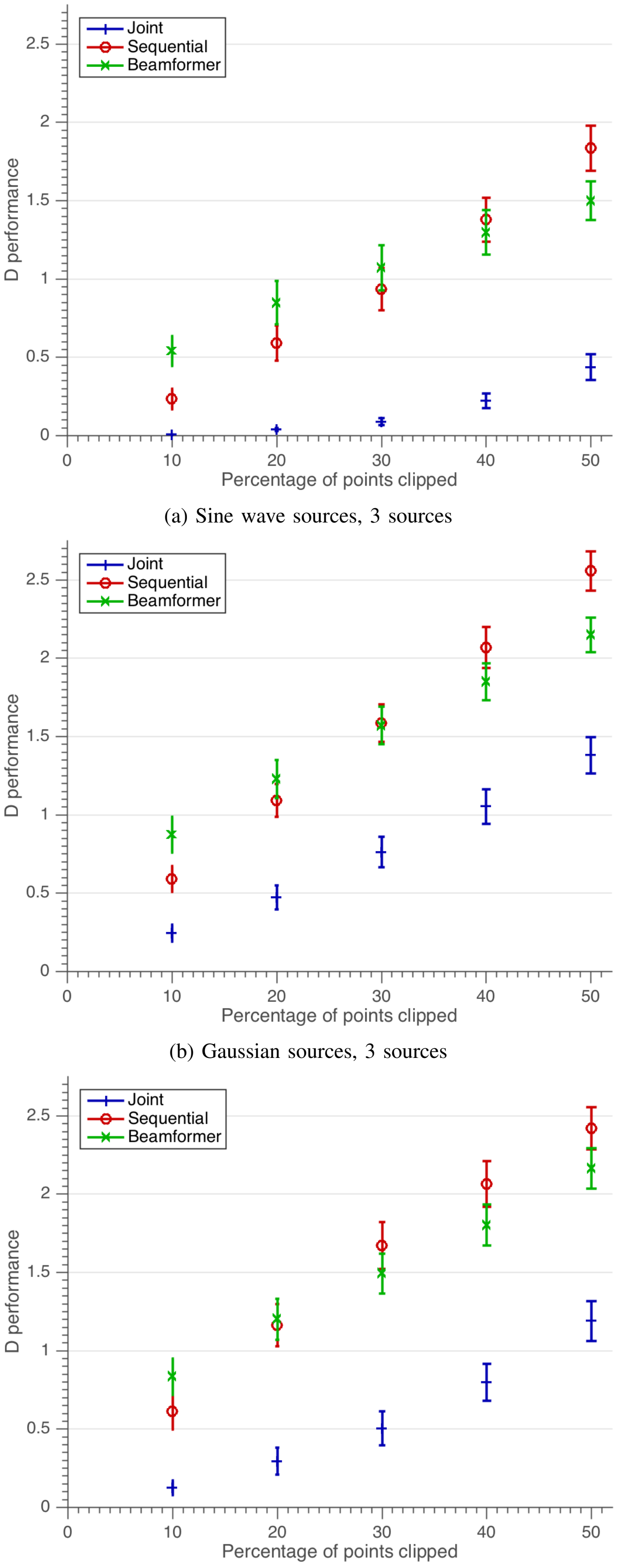
This paper has presented a novel framework for underdetermined source separation from clipped mixtures, which combined ℓ_1 minimisation with knowledge about the clipping geometry. An algorithm was developed to recover the sources by solving a convex optimisation problem constrained by this knowledge. Comparative evaluation experiments of our proposed method showed a significant increase in objective recovery performance compared with sequential approaches for speech and synthetic sources.

References

[1] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[2] A. Charles, “Robustness of ICA methods for clipping-distorted mixtures,” msc thesis, University of Birmingham, 2015.

[3] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, “Sparse Representations in Audio and Music: From Coding to Source Separation,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

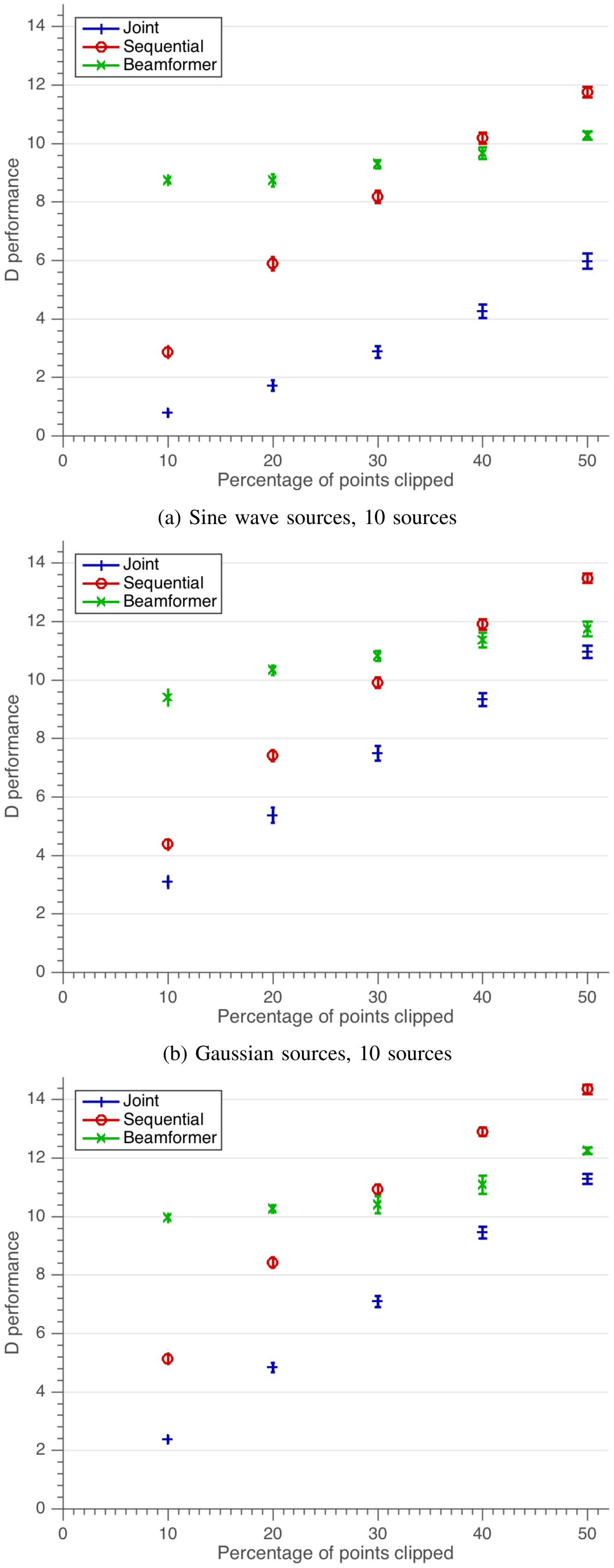


(a) Sine wave sources, 3 sources

(b) Gaussian sources, 3 sources

(c) Speech sources, 3 sources

Fig. 3: Results for 3 sources



(a) Sine wave sources, 10 sources

(b) Gaussian sources, 10 sources

(c) Speech sources, 10 sources

Fig. 4: Results for 10 sources

[4] A. Turl and A. Kabán, “Joint blind source separation and declipping: A geometric approach for time disjoint sources,” in *The 17th IEEE International Symposium on Signal Processing and Information Technology*, December 2017.

[5] B. Defraene, N. Mansour, S. De Hertogh, T. van Waterschoot, M. Diehl, and M. Moonen, “Declipping of Audio Signals Using Perceptual Compressed Sensing,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 12, pp. 2627–2637, 2013.

[6] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1.” http://cvxr.com/cvx, Mar. 2014.

[7] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008.

[8] Z. Koldovský, P. Tichavsky, A. H. Phan, and A. Cichocki, “A Two-Stage MMSE Beamformer for Underdetermined Signal Separation,” *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1227–1230, 2013.

[9] Z. Koldovský, “Udsep.” http://itakura.ite.tul.cz/zbynek/udsep.htm.

[10] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech and Language Processing*, 2006.

[11] Y. Cao, “Hungarian algorithm for linear assignment problems (v2.3),” 2011.

Chapter Five

Two Clipped Mixtures of Many Sources with some Time-Disjointness

5.1 Introduction

In this chapter, we relax the strict time-disjointness assumption made thus far and assume that only a small percentage of the time points of the sources are time disjoint rather than all of them. With this assumption, it turns out we can still estimate the mixing directions. We assume that a sparse representation of the sources exists in a given basis, formulate a convex optimisation and solve it. Our novel method exploits knowledge about the geometry of clipped mixtures and the mixing matrix.

We solve both the determined and underdetermined cases. We assume that two linear mixtures of the sources are observed. Our method is outlined and explained in Section 5.2. Section 5.3 provides a comprehensive testing of our proposed approach on both synthetic and speech data. Our results are shown and discussed in Section 5.4, and our conclusions summarised in Section 5.5.

5.2 Methodology

In practice, clipping occurs when representing a signal containing amplitude values that saturate the maximum amplitude permitted by the system, yet most ICA algorithms neglect this. In a digital system, input samples whose absolute values exceed a given threshold θ have their output limited by this threshold. Denote by $x_{i,n}$ the nth sample of the ith mixture before clipping and $x'_{i,n}$ the equivalent sample after clipping.

$$x'_{i,n} = \begin{cases} \operatorname{sgn}(x_{i,n})\,\theta & \text{if } |x_{i,n}| > \theta, \\ x_{i,n} & \text{if } |x_{i,n}| \leq \theta. \end{cases} \tag{5.1}$$

Equations (3.1) and (3.1) combined describe the model of the problem:

$$x'_{i,n} = \begin{cases} \operatorname{sgn}\left(\sum_{j=1}^{D} A_{i,j} s_{j,n}\right)\theta & \text{if } |\sum_{j=1}^{D} A_{i,j} s_{j,n}| > \theta, \\ \sum_{j=1}^{D} A_{i,j} s_{j,n} & \text{if } |\sum_{j=1}^{D} A_{i,j} s_{j,n}| \leq \theta. \end{cases} \tag{5.2}$$

Now the task is to find both $\mathbf{s}$ and $\mathbf{A}$.

5.2.1 Overview of the proposed approach

Three sequential steps were developed to solve the model in Equation (3.2) and yield estimates of latent sources $\hat{\mathbf{s}}$ from clipped mixtures $\mathbf{x}'$:

1. Mixing matrix estimation (Section 5.2.2).

2. Declipping mixtures by ℓ_1 minimisation (Section 5.2.3).

3. Source estimation (Section 5.2.4).

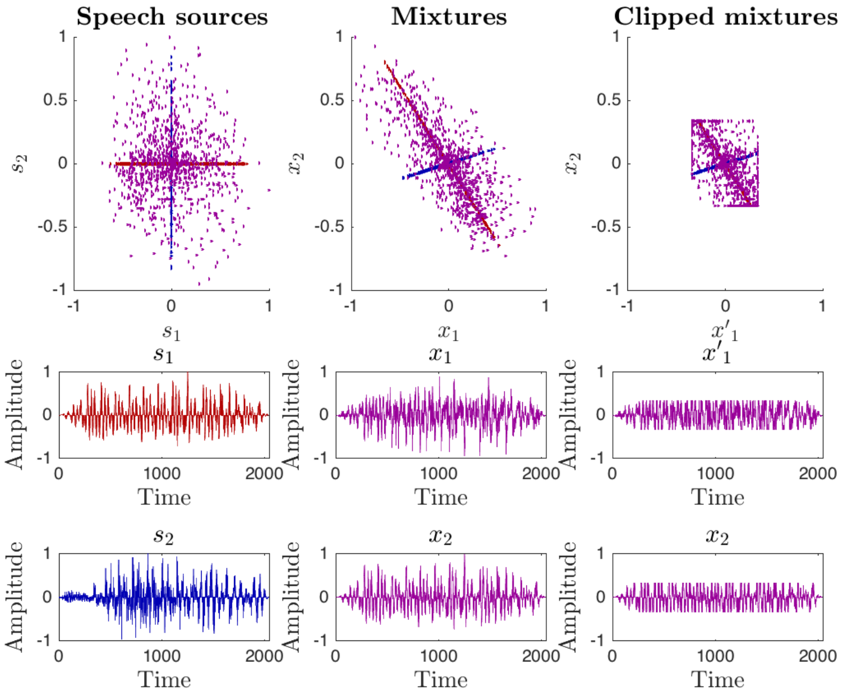


Figure 5.1: Two speech sources, 50% time disjoint (left); linear mixtures of these sources (middle); clipped mixtures, 20% clipped (right).

5.2.2 Mixing matrix estimation

First we obtain clipping threshold θ, which is the largest absolute value among all the individual elements within the vector $\mathbf{x}'$.

$$\theta = \max |\mathbf{x}'| \tag{5.3}$$

As discussed in Chapter 3.1, in a scattergram of linear mixtures of time disjoint sources, the sources can be seen to be straight lines (Plumbley et al., 2010), which represent the mixing directions of **A**. If only some of the points are time disjoint, however, it turns out these lines can still be detected. An example, with sources that are 50% time disjoint to make them visible in the plot, is shown in Figure 5.1.

Since clipping distorts the source lines, we discard all clipped points, where $|x'_{1,n}| = \theta \vee |x'_{2,n}| = \theta$. We denote by M the number of unclipped time points, and calculate the slope of all of these as shown below. The symbol $\odot$ represents the elementwise product.

$$\nabla_{\mathbf{M}} = \left(x'_{2,1}, \ldots, x'_{2,M}\right) \odot \left(\frac{1}{x'_{1,1}}, \ldots, \frac{1}{x'_{1,M}}\right) \tag{5.4}$$

The set $\nabla_{\mathbf{M}}$ contains M entries, some of which may be duplicates as a result our assumption that a small percentage of the columns of $\mathbf{x}$ are time disjoint. We define the set $\nabla_{\mathbf{s}}$ to contain only the D most frequent unique entries of $\nabla_{\mathbf{M}}$. Denote by ∇_1 the first slope, calculated as the mode of $\nabla_{\mathbf{M}}$, and by ∇_2 the second slope, which is the second most frequent entry of $\nabla_{\mathbf{M}}$. We use these slopes to construct an estimated mixing matrix $\hat{\mathbf{A}}$ in the following manner:

$$\hat{\mathbf{A}} = \begin{bmatrix} 1 & 1 \\ \nabla_1 & \nabla_2 \end{bmatrix} \tag{5.5}$$

In the optimisation step, it is important for each of our sources to have the same relative amplitude scaling, so we multiply each column i of $\hat{\mathbf{A}}$ by a scaling factor γ_i. For each ith source, we use the slope ∇_i to naively demix the clipped mixtures $\mathbf{x}'$ and take the maximum absolute value of the ith distorted source as our scaling factor γ_i, as shown below:

$$\gamma_i = \max |\nabla_i x'_1 - x'_2| \tag{5.6}$$

We then use scaling factor γ_i to obtain our scaled mixing matrix estimate $\hat{\hat{\mathbf{A}}}$:

$$\hat{\hat{\mathbf{A}}} = \begin{bmatrix} \max |\nabla_1 x'_1 - x'_2| & \max |\nabla_2 x'_1 - x'_2| \\ \nabla_1 \max |\nabla_1 x'_1 - x'_2| & \nabla_2 \max |\nabla_2 x'_1 - x'_2| \end{bmatrix} \tag{5.7}$$

5.2.3 Declipping by ℓ_1 minimisation

We reconstruct the clipped points under the assumption they have a *sparse representation*. We assume that estimated sources $\hat{\mathbf{s}}$ can be sufficiently represented by a small number of non-zero coefficients $\mathbf{r}$ in the basis matrix $\mathbf{\Psi}$, as shown below:

$$\hat{\mathbf{s}} = \mathbf{r}^T \mathbf{\Psi}^T \tag{5.8}$$

We use the DCT operator matrix, commonly used in signal processing, as $\mathbf{\Psi}$ for our experiments in Section 5.3. It is likely a sparser basis could be chosen, for example for speech, but the DCT is sufficient to demonstrate our approach. We formulate a convex optimisation problem that will yield a sparse solution. We use ℓ_1 minimisation because the number of non-zero elements, known as the ℓ_0 "norm", is not convex. The tasks is to find a solution to the linear system below, where sparse source representations $\mathbf{r} \in \mathbb{R}^{N \times D}$:

$$\text{minimize} \quad \sum_{i=1}^{D} ||\mathbf{r}_i||_1 \quad \text{subject to} \quad \hat{\hat{\mathbf{A}}}\mathbf{r}^T \mathbf{\Psi}^T = \mathbf{x}'. \tag{5.9}$$

We constrain this problem by the mixture model $\mathbf{x}' = \hat{\hat{\mathbf{A}}}\hat{\mathbf{s}} = \hat{\hat{\mathbf{A}}}\mathbf{r}^T \mathbf{\Psi}^T$. We formulate a constraint each for the unclipped portions x'_1 and x'_2. A pair of masking matrices $\mathbf{C_{u1}}$ and $\mathbf{C_{u2}}$ are constructed, consisting of the $\mathbf{I}_N$ rows corresponding to indices of all unclipped points in x'_1 and x'_2 respectively. These matrices are used to form the first pair of constraints in Equation (5.10).

We formulate constraints that ensure the clipping model in Equation (3.1) is obeyed, similar to those in (Defraene et al., 2013). Masking matrices $\mathbf{C_{+1}}$ and $\mathbf{C_{-1}}$ contain $\mathbf{I}_N$ rows corresponding to the indices of positively clipped ($x'_{1,n} = \theta$) and negatively clipped ($x'_{1,n} = -\theta$) points, respectively. Equivalent masking matrices $\mathbf{C_{+2}}$ and $\mathbf{C_{-2}}$ are constructed for those points clipped positively ($x'_{2,n} = \theta$) and negatively ($x'_{2,n} = -\theta$) in the second mixture, respectively. These four matrices allow us to constrain the reconstructed points to obey the clipping model, forming the second and third pairs of constraints in Equation (5.10).

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{D} \|\mathbf{r}_i\|_1 \\ &\text{subject to} && \hat{\hat{\mathbf{A}}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{u1}}^T = x'_1 \mathbf{C_{u1}}^T, \quad \hat{\hat{\mathbf{A}}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{u2}}^T = x'_2 \mathbf{C_{u2}}^T, \\ & && \hat{\hat{\mathbf{A}}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{+1}}^T \geq \theta, \qquad \hat{\hat{\mathbf{A}}}_1 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{-1}}^T \leq -\theta, \\ & && \hat{\hat{\mathbf{A}}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{+2}}^T \geq \theta, \qquad \hat{\hat{\mathbf{A}}}_2 \mathbf{r}^T \mathbf{\Psi}^T \mathbf{C_{-2}}^T \leq -\theta. \end{aligned} \tag{5.10}$$

After solving the optimisation problem, the following equation yields an estimate of the declipped mixture $\hat{x}_1$:

$$\hat{\mathbf{x}} = \hat{\hat{\mathbf{A}}} \mathbf{r}^T \mathbf{\Psi}^T \tag{5.11}$$

To solve the problem in Equation (5.10) we used CVX, a package for specifying and solving convex problems (Grant and Boyd, 2014; Grant and Boyd, 2008).

5.2.4 Source estimation

The source estimates $\hat{\mathbf{s}}$ by the equation below.

$$\hat{\mathbf{s}} = \hat{\hat{\mathbf{A}}}^{-1} \hat{\mathbf{x}} \tag{5.12}$$

5.2.5 Sequential method

To test our novel ideas of using knowledge of the mixture model and estimated relative source scaling to constrain the optimisation (Section 5.2.3), we formulate an alternative problem to declip the mixtures without this information. The mixing matrix estimation (Section 5.2.2) and source estimation (Section 5.2.4) steps remain the same.

The optimisation problem for declipping each clipped mixture x'_i is shown below, where

$w_i \in \mathbb{R}^N$ is a sparse representation of the ith mixture:

$$\begin{aligned} \text{minimize} \quad & \|w_i\|_1 \quad \forall i = 1...F \\ \text{subject to} \quad & {w_i}^T \boldsymbol{\Psi}^T {\mathbf{C_{ui}}}^T = x_i' {\mathbf{C_{ui}}}^T, \\ & {w_i}^T \boldsymbol{\Psi}^T {\mathbf{C_{-i}}}^T \leq -\theta, \\ & {w_i}^T \boldsymbol{\Psi}^T {\mathbf{C_{+i}}}^T \geq \theta. \end{aligned} \tag{5.13}$$

5.3 Experiments

A series of experiments were designed to evaluate our proposed method and three other approaches:

1. The joint source separation and declipping algorithm, described in Section 5.2.
2. The sequential version, described in Section 5.2.5.
3. FastICA (Hyvärinen and Oja, 2000a) - one of the most well known and popular source separation algorithms. This was used for the determined case.
4. Beamformer (Koldovský et al., 2013) - a popular algorithm for solving the underdetermined case.

For FastICA and Beamformer, we first declip the mixtures $\mathbf{x}'$ as in Equation (5.13) from Section 5.2.5. The declipped mixtures are then processed by the FastICA code from (Gävert et al., 2017).

Each experiment ran for 50 repetitions. The length N of the sources and mixtures was fixed at 2048 time points. In each case, we varied the number of sources (2, 3, 5 and 10), the window size (64, 128, 256), the source type (Gaussian, sine, speech) and percentage time-disjointness (0.5%, 1%, 2% and 5%).

5.3.1 Hypotheses

1. The joint algorithm will yield better performance than the sequential approach in general
   - The sources are likely to be sparser than the mixtures, so a sparsity-based declipping approach that assumes sparseness of the sources $\mathbf{s}$ rather than the mixtures $\mathbf{x}$ will yield a result with fewer errors.
2. Speech sources will yield worse performance than sine wave sources
   - The DCT basis, which consists of sinusoids, will enable a sparser representation of sine wave sources than speech sources.
3. FastICA and Beamformer will perform worse than our approaches
   - Our approach exploits knowledge of the clipping geometry and partial time disjointness assumption to estimate mixing matrix $\hat{\hat{\mathbf{A}}}$, which FastICA and Beamformer do not.
4. A larger proportion of clipped samples will lead to worse performance
   - More of the data will need to be approximated by the optimisation, increasing the chance of errors.
5. Zero time-disjointness will lead to worse performance
   - The time-disjointness is required to estimate the mixing directions.

5.3.2 Preparing the data

There types of source signal were used: synthetic sources constructed from a few sine waves, synthetic Gaussian sources and speech.

1. *Synthetic (sine wave mixtures)*

   (a) Determined case

   $s_{1,n} = 3\sin(0.02n) + 0.8\sin(3.5n) + 0.8\sin(4.1n) + 0.5\sin(4.5n)$

   $s_{2,n} = 3.4\sin(0.03n) + 0.5\sin(2.2n) + 0.6\sin(1.4n) + 0.2\sin(3.7n)$

   (b) For the underdetermined case with three sources we added an additional source

   $s_{3,n} = 3.7\sin(0.015n) + 0.8\sin(3.6n) + 0.3\sin(2.4n) + 0.5\sin(2.7n)$

   (c) For the underdetermined case with ten sources we generated the following sources

   $s_{1,n} = 0.02\sin(0.054n) + 0.09\sin(0.031n)$

   $s_{2,n} = 0.03\sin(0.029n) + 0.07\sin(0.924n)$

   $s_{3,n} = 0.08\sin(0.074n) + 0.04\sin(0.430n)$

   $s_{4,n} = 0.15\sin(0.189n) + 0.04\sin(0.185n)$

   $s_{5,n} = 0.43\sin(0.687n) + 0.03\sin(0.905n)$

   $s_{6,n} = 0.16\sin(0.018n) + 0.05\sin(0.034n)$

   $s_{7,n} = 0.06\sin(0.036n) + 0.05\sin(0.025n)$

   $s_{8,n} = 0.07\sin(0.062n) + 0.08\sin(0.040n)$

   $s_{9,n} = 0.06\sin(0.078n) + 0.07\sin(0.094n)$

   $s_{10,n} = 0.45\sin(0.081n) + 0.07\sin(0.092n)$

   The sources were then normalised:

   $s_{i,n} = s_{i,n}/\max|s_i| \quad \forall i = 1...D$

2. *Gaussian*

   Gaussian sources were generated using MATLAB's *normrnd* function

3. *Speech*

   Ten speech recordings of a feminine voice saying the numbers one to ten were produced.

These recordings were 2048 samples long and had a sampling rate of 22050Hz at a bit depth of 16.

At the start of each experiment, the sources $\mathbf{s}$ were processed such that a random percentage ζ of the points were made disjoint in time. This was done by selecting ζ of the points at random and, for each of those points, making a randomly selected source zero.

The mixing matrix was generated $\mathbf{A} \overset{iid}{\sim} \mathcal{U}(-1, 1)$. If $\mathbf{A}$ was singular, it was discarded and generated again. The mixing matrix was scaled $\mathbf{A} = \mathbf{A}/\max|\mathbf{A}|$. The mixtures $\mathbf{x}$ were generated as in Equation (3.1) and processed as described by Equation (3.1) to give $\mathbf{x}'$. Five clipping thresholds θ were selected, such that 10%, 20%, 30%, 40% and 50% of the samples are clipped each time. The length N of the mixtures and sources was fixed as $N = 2048$. Each experiment was repeated 50 times.

5.3.3 Implementation and frame-based processing

For the constrained optimisation in Equation (5.10), it is necessary to process the audio in short sequential frames and concatenate the results to form the declipped mixtures $\hat{\mathbf{x}}$, as done in (Defraene et al., 2013). This greatly decreases processing time by making the basis $\mathbf{\Psi}$ smaller. For our experiments, the mixtures $\mathbf{x}'$ were split into uniform-windowed non-overlapping frames of length $n = 64, 128$ and 256.

5.3.4 Performance measurement

We cannot measure performance by comparing the estimated mixing matrix $\hat{\hat{\mathbf{A}}}$ with the true mixing matrix $\mathbf{A}$, as the Amari index does for ICA, because the accuracy of the estimated sources $\widehat{\mathbf{s}}$ depends on the declipping performance. For this reason, we measured performance by comparing the sources $\mathbf{s}$ with the estimates $\widehat{\mathbf{s}}$, using the following equation, cited in (Vincent, Gribonval, and Fevotte, 2006):

$$\mathbf{D} := \min_{\epsilon=\pm 1} \left\| \frac{\widehat{s_i}}{||\widehat{s_i}||} - \epsilon \frac{s_i}{||s_i||} \right\|_2^2 \tag{5.14}$$

The final problem to solve is the permutation indeterminacy of the sources. We used the Hungarian algorithm, as implemented in (Cao, 2011). Each source s_i is compared with each source estimate $\widehat{s_j}$ and the results are stored in a $D \times D$ matrix. The algorithm processes this matrix to determine the lowest performance cost and corresponding estimated source to true source assignments.

5.4 Results

Our results plot the obtained performance **D** and smaller values are better. We report average performances from 50 independent repetitions, and the error bar shows the standard error. The number above the error bar shows the number of times FastICA did not converge on a solution.

As expected from previous chapters, we find that increasing the percentage of clipped time points decreases performance. We can see this across the plots in this section. This is in line with our expectations and hypotheses in Section 5.3.1. When more of the data are missing, more of the points need to be estimated by the convex optimisation, increasing the chance of errors.

Our proposed joint method, in general, performed better when compared with the sequential approach and FastICA on the declipped mixtures. The sequential method yielded better performance than the joint method when the window size was 256 samples in length, as we see in Figure 5.2. This could suggest that 256 is a suboptimal window size for our method, but it might also be explained by the signal characteristics of the chosen sources. More experimentation would

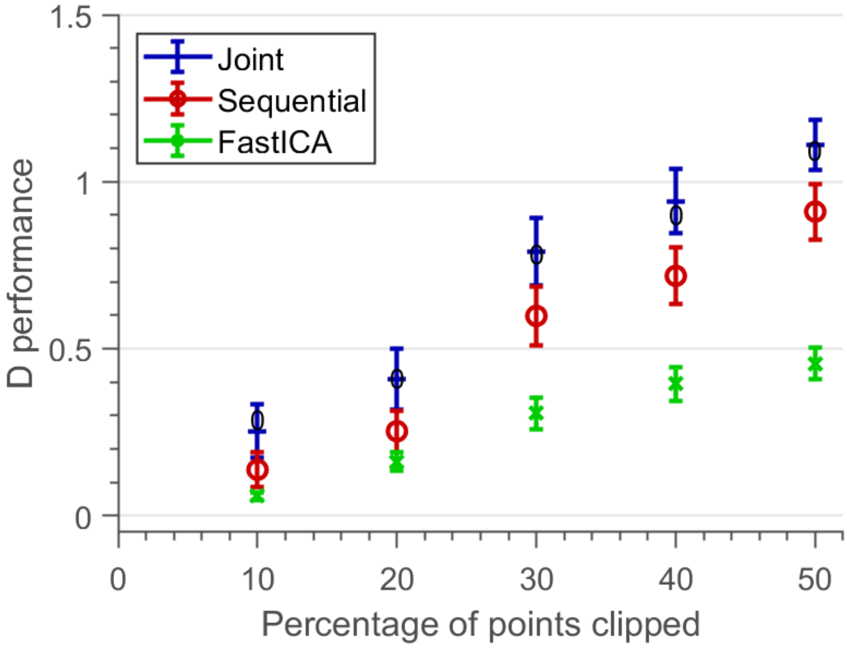


Figure 5.2: Sequential method yields better performance than joint method at 256 window size (2 sine sources, 256 window size, 0.5% time-disjointness)

be needed to determine the cause of this anomaly.

It is likely that performance was improved using our joint method by modelling the sources themselves as having sparse representations rather than the mixture signals, as in the sequential method. This finding matches our first hypothesis in Section 5.3.1. In preliminary experiments, we discovered that altering the relative scaling of the estimated mixing matrix columns affected the performance. By scaling the estimated mixing matrix such that each source had approximately the same peak amplitude, we attempted to ensure that the sources were equally weighted in the optimisation in Section 5.2.3 such that no source was prioritised.

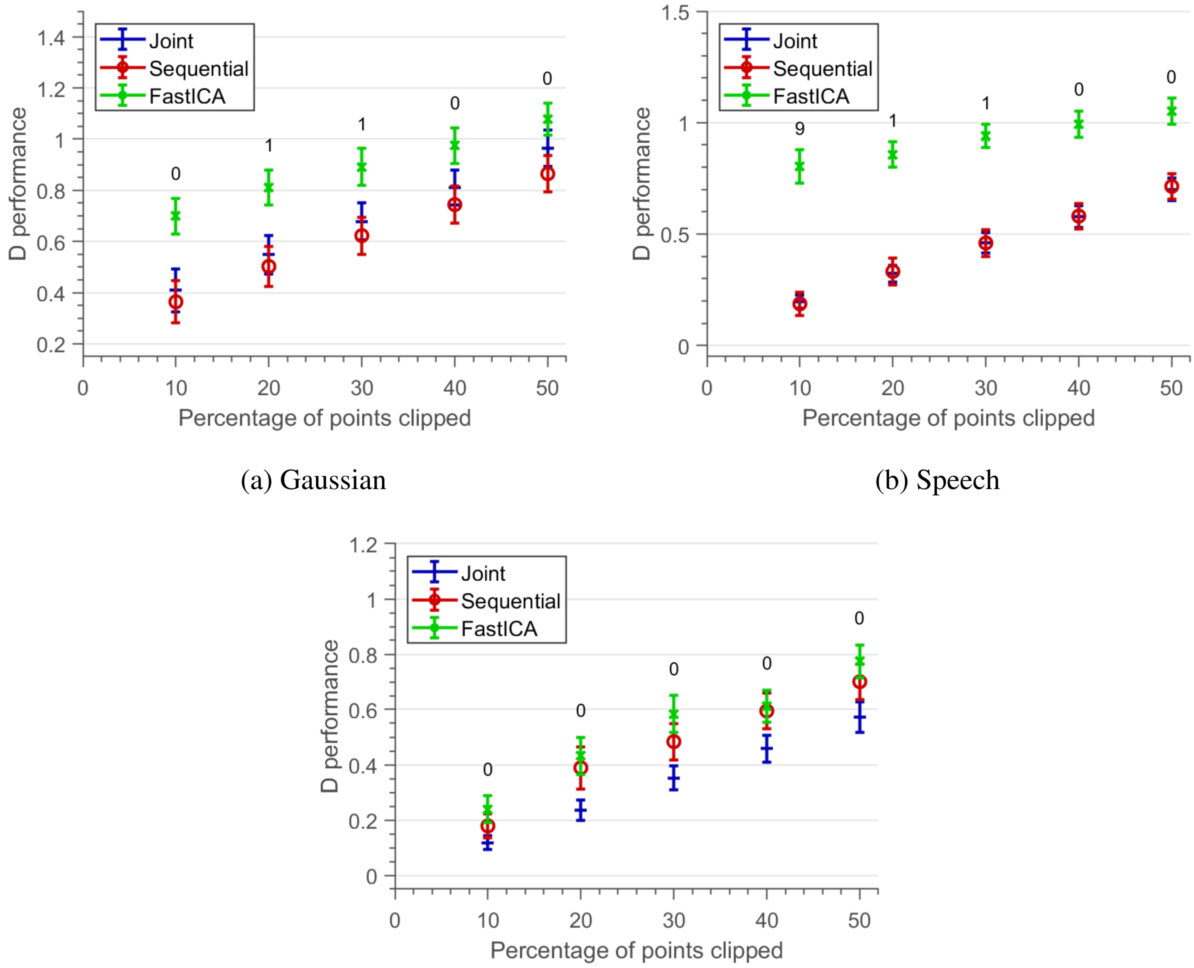
As seen in Figure 5.3, sine sources attained better performance than both Gaussian and speech sources in every case. There appeared to be little difference with this approach when comparing Gaussian and speech sources for our approach, which one does not observe using traditional ICA approaches like FastICA that cannot handle Gaussian sources. Our method for obtaining the mixing directions from the partial time-disjointness geometry is likely what facilitates the separation of Gaussian sources in our approach.

The FastICA approach yielded better performance scores with all source types at zero time-disjointness, as seen in Figure 5.4. Our approach exploits partial time-disjointness, so when there is no time-disjointness at all, the mixing directions cannot be estimated accurately. The presence of some disjointness is essential to our approach. Conversely, FastICA and Beamformer do not exploit time-disjointness at all, so it is expected that these methods would perform less well without this advantage in comparison with our approach when there is some time-disjointness to be exploited.

Increasing the number of sources worsens performance, as seen in Figure 5.5. As the number of sources increases, the maximum angle between the sources decreases, and as such this makes geometric observations more difficult.

At 0.5% time-disjointness FastICA, the joint and sequential methods yielded similar performance, however, at time-disjointness higher than 0.5% our method yielded better performance than FastICA with the best performance being at 2% time-disjointness. It is likely that 2% is a sweet spot, where we have just enough time points to accurately estimate the mixing directions, but not too many such that the data is not sparse enough for our approach.

Beamformer performed worse than our methods in every experiment regardless of time-disjointness. From our third and fifth hypotheses in Section 5.3.1, we see this is as expected. In the presence of clipping distortion, the Beamformer algorithm cannot accurately determine the mixing directions and fails to separate the sources.

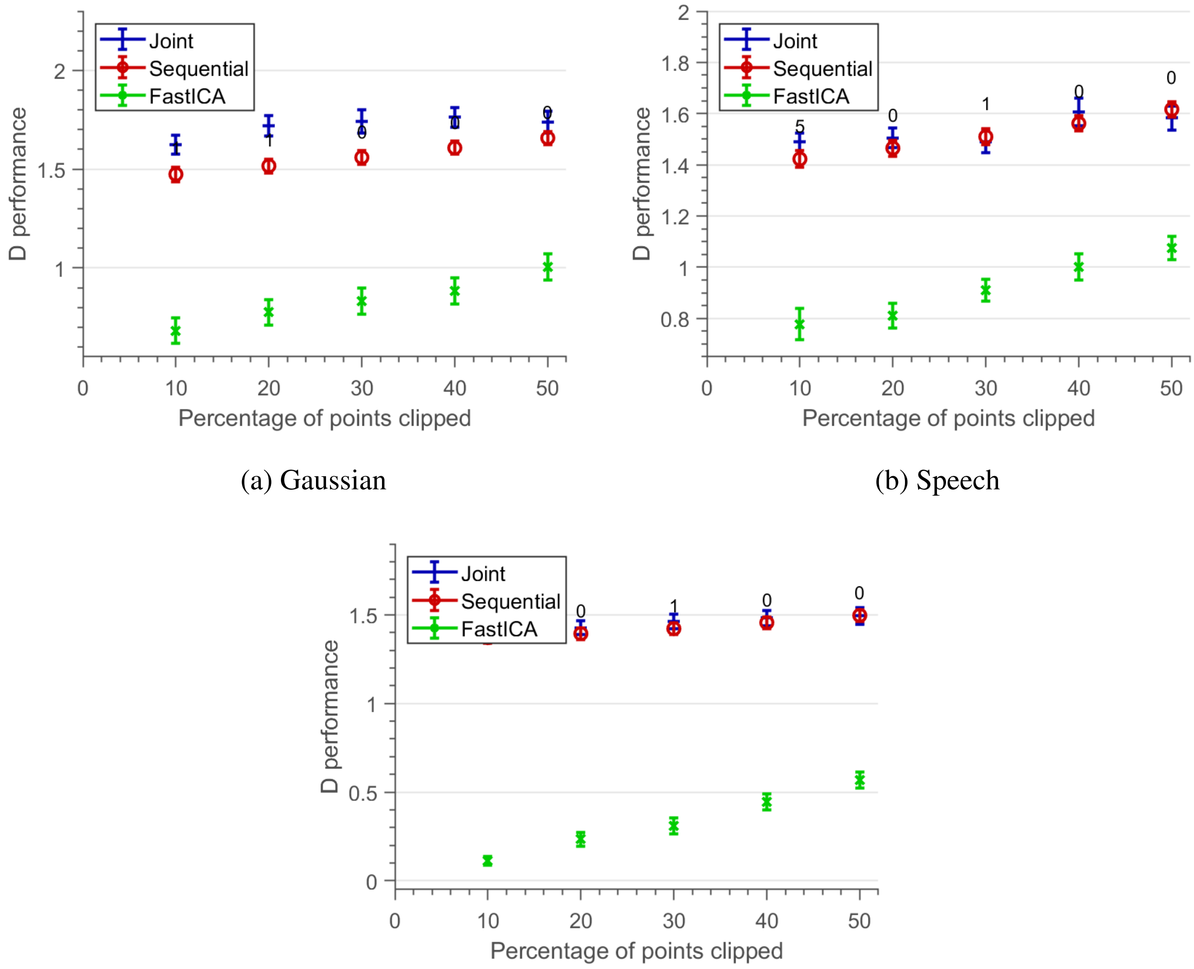


(a) Gaussian

(b) Speech

(c) Sine

Figure 5.3: Sine sources yield better performances than speech and Gaussian (2 sources, 64 window size, 5% time-disjointness)

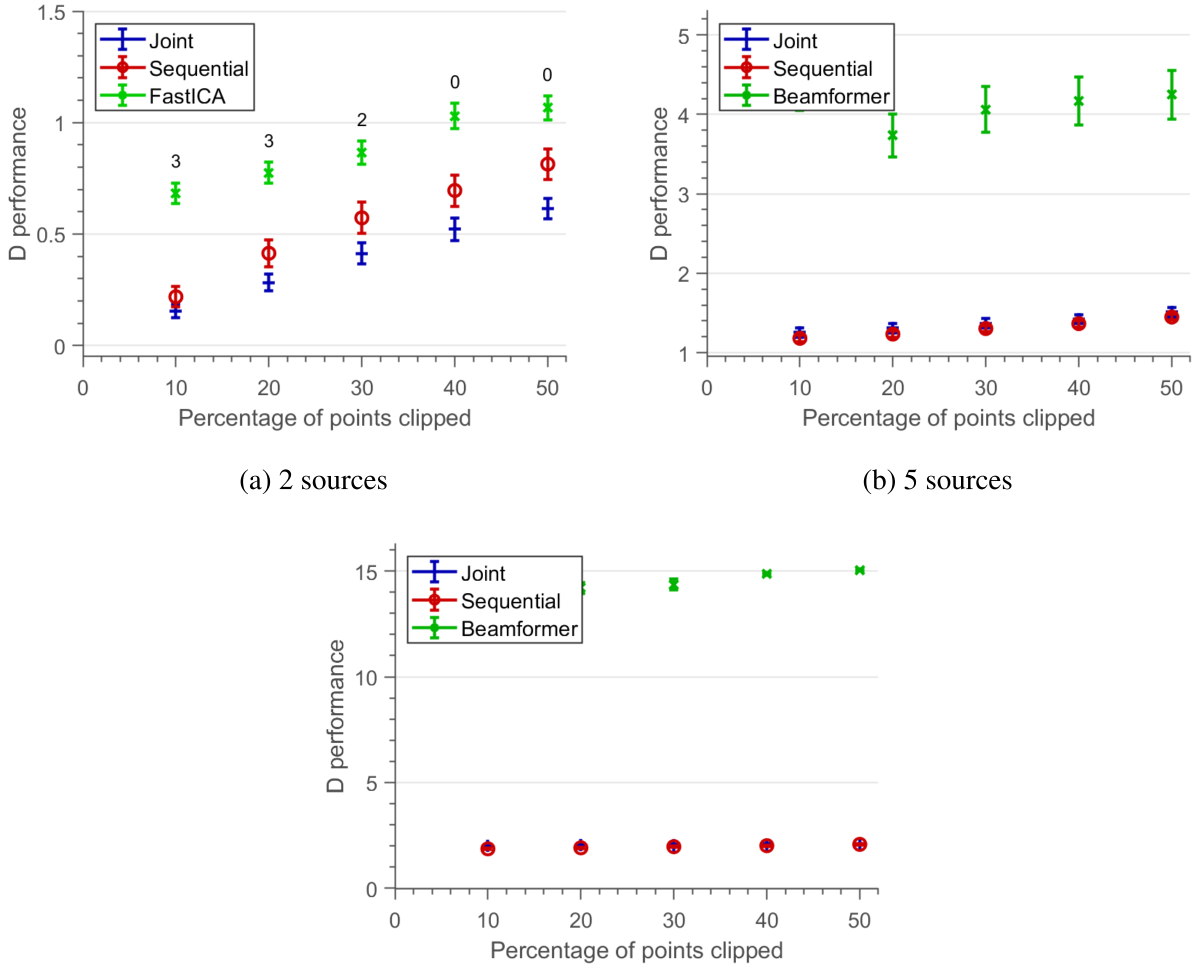


(a) Gaussian

(b) Speech

(c) Sine

Figure 5.4: FastICA yields better performs than our approach at 0% time-disjointness (2 sources, 256 window size, 0% time-disjointness)



(a) 2 sources

(b) 5 sources

(c) 10 sources

Figure 5.5: Increasing the number of sources worsens performance (Sine sources, 256 window size, 2% time-disjointness)

5.5 Conclusion

In this chapter, we presented our novel source separation method for mixtures affected by clipping distortion and we relax the strict time-disjointness assumption of the previous chapters. Our method used a scaled estimate of the mixing matrix, combined with knowledge about the geometry of clipped mixtures, to constrain a convex optimisation problem. Evaluation experiments comparing our method with existing approaches showed a significant increase in objective recovery performance for our proposed method for both speech and synthetic source signals when there was some time-disjointness, with peak performance indicating a sweet spot of 2% time-disjointness. Future work would include disjointness in other spaces, rather than only time-disjointness to seek to improve performance. One key area to investigate is time-frequency disjointness. This is more realistic because in nature sounds often overlap but they don't as often overlap in both time and frequency.

Chapter Six

Discussions and Conclusions

6.1 Research summary

The research in this thesis sought to solve the problem of source separation for mixtures affected by clipping distortion. We started with the determined case, two mixtures and two sources, where only one of the mixtures was affected by clipping distortion. This was then expanded to include to the underdetermined case where there are more sources than mixtures and where both mixtures were clipped. These first two chapters (Chapters Three and Four) made an assumption of strict time-disjointness. Our final experiments in Chapter Five relaxed this assumption to only partial time-disjointness for both cases of determinedness.

Existing source separation methods are not suitable for solving the combined problem of source separation and clipping distortion (Turl, 2015).

We devised a series of approaches based on formulating convex optimisation problems that exploited knowledge discerned from the properties of the geometry of clipped mixtures. The underdetermined case allowed us to formulate additional constraints based on the geometry of the clipped mixtures which improved the performance of the method. Finally, we extended our approach so that the strict time-disjointness assumption was not required by devising a new method to discern the mixing matrix.

Our method was tested with linear mixtures of 3 different source types, sine, Gaussian and speech. In Chapter Three we varied the percentage of the samples that were clipped. Chapter Four expanded this to vary the number of sources. Then in Chapter Five we also varied the percentage of time-disjointness and the size of the window. In all experiments, performance was measured by comparing the sources with the estimates from our method using the D performance metric cited in (Vincent, Gribonval, and Fevotte, 2006). For the determined case we compared our method to the performance of FastICA (Hyvärinen and Oja, 2000a), for the underdetermined case our method was compared to Beamformer (Koldovský et al., 2013).

6.2 Results and conclusions

In all of our experiments our method was shown to outperform FastICA and Beamformer when there was some time-disjointness. At zero time-disjointness FastICA outperformed our method but Beamformer did not. These results are expected, as our method is designed to take time-disjointness into consideration in order to solve clipped source separation whereas neither FastICA nor Beamformer do this. In Chapter Five, there appeared to be an ideal time-disjointness of 2%. As expected, increasing the amount of clipping distortion worsened the performance of all methods, however, in general our method still performed better than the others.

The method within this thesis is the first to explicitly look at the combined problem of clipping and source separation. Our experiments show there is benefit to solving both these problems together rather than sequentially. In general, solving the problem jointly resulted in better source recovery than sequentially.

6.3 Future work

6.3.1 Time-frequency disjointness

Future work would broaden the scope of mixtures that we can tackle to include mixtures that are disjoint in the time-frequency domain. In nature sounds often overlap but less frequently overlap in both time and frequency so this would make the problem more realistic. Expanding into this domain may result in increased performance across a wider range of types of mixtures.

6.3.2 Additional performance measures

In this research the primary performance measure used was D performance (Vincent, Gribonval, and Fevotte, 2006). It would be interesting to test our methodology using other performance metrics. An example of a set of additional metrics that could be used is the Perceptual Evaluation methods for Audio Source Separation (PEASS) Toolkit (Emiya et al., 2010) which contains a set of subjective perceptual measures of audio source separation. Another example of performance measures is Gribonval, Vincent, and Févotte's MATLAB toolbox called *BASS-dB* (Gribonval, Vincent, and Févotte, 2006) which contains a set a source separation performance measures.

It would be highly desirable to build on the work of Emiya et al. and devise an application-agnostic performance metric for ICA. This would be challenging, since each application has different priorities when it comes to measuring performance, especially when comparing subjective and objective measurements. Nonetheless, a rigorous study of state-of-the-art performance measures would be beneficial in the early stages of this project.

An additional area of performance that could be considered for future work is that of efficiency. The methods used in this paper could be compared in terms of their time taken to find a solution, the processing power required and energy efficiency. Those who have written about ICA performance have mostly indicated that further work is necessary. (Emiya et al., 2011) hoped that their study could become the basis for a more sophisticated and standardised subjective testing protocol in the future. Cabras (2011) noted in his thesis that there were still several unresolved issues with ICA; one of his recommendations was a further study focusing on the tradeoff between computational power and ICA performance.

6.3.3 Additional source separation algorithms

We could also compare our method to additional algorithms that solve the problem of source separation. One potential algorithm that could be tested against is Bilen, Ozerov, and Pérez's NMF method. Another is the maximum likelihood ICA algorithm cited in (MacKay, 2002).

6.3.4 Computational learning theory

A more challenging area of future research could be to quantify the conditions required in order to guarantee separation and the likelihood of separation occurring. Computational learning theory provides the basis for these investigations. This theory attempts to answer questions such as *under what conditions is learning possible?* and *under what conditions is learning guaranteed to be successful?* (Mitchell, 1997). Another important question deals with sample complexity, that is, *how many samples do I need in order to learn?*.

There has been surprisingly little work done to apply learning theory principles to ICA. A study by Vempala and Xiao (2014) presented a technique for reducing the sample complexity of matrix and tensor decomposition algorithms such as those used to perform ICA. Vempala and Xiao's study produced a polynomial-time complexity algorithm for standard ICA with nearly linear sample complexity: a substantial improvement on previous polynomial time algorithms. This was achieved using a recursive version of Fourier principal component analysis (PCA). While it is encouraging to see some work in this area, it demonstrates that there is plenty of room for a deeper learning theory driven study of ICA.

6.3.5 Problematic mixtures

Perhaps a more interesting problem involves looking at the performance of ICA from the perspective of the mixtures themselves rather than the results of the algorithms. There does not appear to have been much work done to identify which particular mixtures prove most problematic for ICA to decompose into components. Reiss and Uhle (2010) conducted a study to improve source separation by using infinite impulse response (IIR) filters to remove direct crosstalk: that is, where the signal from microphone *m1* contains a portion of the signal from microphone *m2* due to the imperfections of audio mixing circuitry. They discovered that separation performance was degraded by fractional time delays and the directivity of the microphones and the sources.

Another obvious candidate for problematic mixtures would be low bit-rate lossy compressed audio files. Although studies have used ICA to facilitate low bit-rate audio compression (Ben-Shalom, Werman, and Dubnov, 2003), there do not appear to have been any studies on ICA applied to such audio files. Since lossy compression tends to discard information considered irrelevant (Kuhn, 2005), it would be interesting to discover the extent to which this affects performance for ICA working on such mixtures.

Convolutive mixtures, could be another option to extend this research into. These mixtures contain multiple time delayed instances of the sources which is more realistic as it mimics the multiple reflections of sound waves heard in a physical location.

Other candidates include phonographic recordings affected by wow and flutter: two common irregularities in analogue audio recordings (Howarth and Wolfe, 2004). Variation in the turntable motor speed leads to time irregularities known as flutter; a poorly mounted phonograph record leads to pitch irregularities known as wow.

References

Adler, Amir et al. (2012). “Audio Inpainting”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.3, pp. 922–932.

Bell, Anthony J and Terrence J Sejnowski (1995). “Blind separation and blind deconvolution: an information-theoretic approach”. In: *1995 International Conference on Acoustics, Speech, and Signal Processing* 5, pp. 3415–3418.

Ben-Shalom, Adiel, Michael Werman, and Shlomo Dubnov (2003). “Study of mutual information in perceptual coding with application for low bit-rate compression.” In: *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pp. 849–854. URL: http://www.kecl.ntt.co.jp/icl/signal/ica2003/cdrom/data/0178.pdf.

Bilen, Çağdaş, Alexey Ozerov, and Patrick Pérez (2015a). “Audio Declipping via Nonnegative Matrix Factorization.” In: *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 1–5.

— (Aug. 2015b). “Joint Audio Inpainting and Source Separation”. In: *Latent Variable Analysis and Signal Separation*. Liberec, Czech Republic: Springer International Publishing, pp. 251–258.

Bingham, Ella, Ata Kabán, and Mark Girolami (2001). “Finding Topics in Dynamical Text: Application to Chat Line Discussions.” In: URL: http://users.ics.aalto.fi/ella/publications/bingham_www10.pdf.

Blumensath, Thomas et al. (2012). *Greedy algorithms for compressed sensing.*

Cabras, Guiseppe (2011). “Advanced component analysis techniques for signal decomposition and their applications to audio restoration and volcanic seismology”. PhD thesis. Università degli Studi di Udine.

Candès, Emmanuel J, Justin Romberg, and Terence Tao (2006). “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on information theory* 52.2, pp. 489–509.

Candès, Emmanuel (n.d.). *The ℓ_1-MAGIC Toolbox*. https://candes.su.domains/software/l1magic/. Accessed on 17th Sept 2023.

Cao, Yi (2011). *Hungarian Algorithm for Linear Assignment Problems (V2.3)*. URL: https://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems--v2-3-?.

Chen, Aiyou (2006). “Fast Kernel Density Independent Component Analysis”. In: *Independent Component Analysis and Blind Signal Separation: 6th International Conference, ICA 2006, Charleston, SC, USA, March 5-8, 2006. Proceedings*. Ed. by Justinian Rosca et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 24–31.

Chen, Scott Shaobing, David L Donoho, and Michael A Saunders (2001). “Atomic decomposition by basis pursuit”. In: *SIAM review* 43.1, pp. 129–159.

Chien, Jen-Tzung and Hsin-Lung Hsieh (2012). “Convex Divergence ICA for Blind Source Separation.” In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.1, pp. 302–313.

Chien, Jen-Tzung and Po-Kai Yang (2016). “Bayesian Factorization and Learning for Monaural Source Separation”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.1, pp. 185–195.

Crain, Thomas R and Dianne J Van Tasell (1994). “Effect of peak clipping on speech recognition threshold.” In: *Ear and hearing* 15.6, pp. 443–453.

Defraene, B et al. (2013). “Declipping of Audio Signals Using Perceptual Compressed Sensing”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.12, pp. 2627–2637.

Donoho, D.L. (2006). “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52.4, pp. 1289–1306. DOI: 10.1109/TIT.2006.871582.

Emiya, Valentin et al. (2010). “The PEASS Toolkit - Perceptual Evaluation methods for Audio Source Separation”. In: *Latent Variable Analysis and Signal Separation*.

— (2011). “Subjective and Objective Quality Assessment of Audio Source Separation”. In: *IEEE Transactions on Audio, Speech and Language Processing* 19 (7), pp. 2046–2057. DOI: 10.1109/tasl.2011.2109381. URL: http://hal.inria.fr/inria-00567152/PDF/emiya2011.pdf.

Ewert, Sebastian et al. (2014). “Score-Informed Source Separation for Musical Audio Recordings: An overview.” In: *IEEE Signal Processing Magazine* 31.3, pp. 116–124.

Funaro, Maria, Erkki Oja, and Harri Valpola (2003). “Independent component analysis for artefact separation in astrophysical images”. In: *Neural networks* 16.3, pp. 469–478.

Gävert, H. et al. (July 15, 2017). *FastICA for MATLAB*. Version 2.5. URL: http://research.ics.aalto.fi/ica/fastica/.

Grais, Emad M and Hakan Erdogan (May 2013). “Regularized nonnegative matrix factorization using Gaussian mixture priors for supervised single channel source separation”. In: *Computer Speech & Language* 27.3, pp. 746–762.

Grais, Emad M, Mehmet U Sen, and Hakan Erdogan (2014). “Deep neural networks for single channel source separation”. In: *Acoustics*, pp. 3734–3738.

Grant, Michael and Stephen Boyd (2008). “Graph implementations for nonsmooth convex programs”. In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, pp. 95–110.

— (Mar. 2014). *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. http://cvxr.com/cvx.

Gribonval, R., E. Vincent, and C. Févotte (2006). *BASS-dB: Blind Audio Source Separation Evaluation Database*. URL: http://bass-db.gforge.inria.fr/BASS-dB/.

Harvilla, Mark J (Nov. 2014). “Compensation for Nonlinear Distortion in Noise for Robust Speech Recognition”. PhD thesis. Pittsburgh, PA.

Helén, Marko and Tuomas Virtanen (Sept. 2005). “Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine”. In: *Signal Processing Conference, 2005 13th European*. IEEE. Antalya: IEEE, pp. 1–4.

Howarth, Jamie and Patrick J. Wolfe (Oct. 2004). “Correction of Wow and Flutter Effects in Analog Tape Transfers”. In: *Audio Engineering Society Convention 117*. URL: http://www.aes.org/e-lib/browse.cfm?elib=12870.

Huang, Po-Sen et al. (2014). “Deep learning for monaural speech separation”. In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence: IEEE, pp. 1562–1566.

Huber, Peter J (June 1985). “Projection Pursuit”. In: *The Annals of Statistics* 13.2, pp. 435–475.

Hyvärinen, A. (1999). “Fast and Robust Fixed-Point Algorithms for Independent Component Analysis”. In: *IEEE Trans. on Neural Networks* 10(3), pp. 626–634.

Hyvärinen, A. and E. Oja (2000a). “Independent component analysis: algorithms and applications”. In: *Neural Networks* 13.4-5, pp. 411–430. DOI: 10.1016/s0893-6080(00)00026-5.

Hyvärinen, Aapo (Apr. 2001). “Complexity pursuit: Separating interesting components from time series”. In: *Neural Computation* 13.4, pp. 883–898.

— (Feb. 2013). “Independent component analysis: recent advances”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1984.

Hyvärinen, Aapo and Erkki Oja (May 2000b). “Independent Component Analysis: Algorithms and Applications”. In: *Neural Networks* 13.4-5, pp. 411–430.

Hyvärinen, Aapo, Juha Karhunen, and Erkki Oja (2001). *Independent component analysis*. John Wiley & Sons.

Ji, Shihao, Ya Xue, and Lawrence Carin (2008). “Bayesian compressive sensing”. In: *IEEE Transactions on signal processing* 56.6, pp. 2346–2356.

Jutten, Christian and Juha Karhunen (Apr. 2003). “Advances in Nonlinear Blind Source Separation”. In: *4th International Symposium on Independent Component Analysis and Blind Source Separation*. Nara, Japan, pp. 245–256.

Kitic, Srdjan et al. (2013). “Consistent Iterative Hard Thresholding for Signal Declipping.” In: *ICASSP*, pp. 5939–5943.

Koldovský, Zbyněk (n.d.). *UDSEP*. http://itakura.ite.tul.cz/zbynek/udsep.htm.

Koldovský, Zbyněk et al. (2013). “A Two-Stage MMSE Beamformer for Underdetermined Signal Separation”. In: *IEEE Signal Processing Letters* 20.12, pp. 1227–1230.

Kolenda, Thomas (2002). *ICA:DTU Toolbox*. IMM, Technical University of Denmark. URL: https://orbit.dtu.dk/en/publications/dtutoolbox/ (visited on 03/12/2023).

Kounades-Bastian, Dionyssos et al. (2015). “A Variational EM Algorithm for the Separation of Time-Varying Convolutive Audio Mixtures.” In: *CoRR abs/1510.04595*.

Kuhn, Markus (2005). *Information Theory and Coding - Image, video and audio compression*. URL: https://www.cl.cam.ac.uk/teaching/2002/InfoTheory/mgk/slides.pdf.

Kulkarni, Amey M, Houman Homayoun, and Tinoosh Mohsenin (2014). “A parallel and reconfigurable architecture for efficient OMP compressive sensing reconstruction”. In: *Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI*, pp. 299–304.

Learned-Miller, Erik G and John W Fisher III (Dec. 2003). “ICA using spacings estimates of entropy”. In: *The Journal of Machine Learning Research* 4, pp. 1271–1295.

Lee, Te-Won et al. (2000). “A Unifying Information-Theoretic Framework for Independent Component Analysis”. In: *Computers & Mathematics with Applications* 39.11, pp. 1–21.

Lefèvre, Augustin (2012). “Dictionary learning methods for single-channel source separation”. PhD thesis. École normale supérieure de Cachan - ENS Cachan: General Mathematics.

Mach, Václav (2016). “Inpainting of missing audio signal samples”. PhD thesis. Brno.

MacKay, David J C (Oct. 2002). *Maximum likelihood and covariant algorithms for independent component analysis*. Tech. rep. Cambridge: University of Cambridge.

Mitchell, Tom M (1997). *Machine Learning*. McGraw-Hill, p. 201.

Mogi, Reona and Hiroyuki Kasai (2012). “Noise-Robust environmental sound classification method based on combination of ICA and MP features”. In: *AIR* 2.1. DOI: 10.5430/air.v2n1p107.

Nesbit, Andrew et al. (July 2010). “Audio Source Separation using Sparse Representations”. In: *Machine Audition*. Ed. by Wenwu Wang. IGI Global, pp. 246–265.

Oja, Erkki, Kimmo Kiviluoto, and Simona Malaroiu (2000). “Independent component analysis for financial time series”. In: *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*. IEEE, pp. 111–116.

Oja, Erkki and Zhijian Yuan (2006). “The FastICA algorithm revisited: Convergence analysis”. In: *Neural Networks, IEEE Transactions on* 17.6, pp. 1370–1381.

Pajunen, Petteri and Mark Girolami (2000). “Implementing decisions in binary decision trees using independent component analysis”. In: *Proceedings of ICA*, pp. 477–481.

Parvaix, Mathieu and Laurent Girin (Aug. 2011). “Informed Source Separation of Linear Instantaneous Under-Determined Audio Mixtures by Source Index Embedding”. In: *IEEE Transactions on Audio, Speech and Language Processing* 19.6, pp. 1721–1733.

Plumbley, Mark D et al. (2010). “Sparse Representations in Audio and Music: From Coding to Source Separation”. In: *Proceedings of the IEEE* 98.6, pp. 995–1005.

Qin, Shun (2020). “Simple algorithm for L1-norm regularisation-based compressed sensing and image restoration”. In: *IET Image Processing* 14.14, pp. 3405–3413.

Reiss, Joshua and Christian Uhle (Nov. 2010). “Determined Source Separation for Microphone Recordings Using IIR Filters”. In: *Audio Engineering Society Convention 129*. URL: http://www.aes.org/e-lib/browse.cfm?elib=15611.

Richard, Gaël (Jan. 2014). “Informed Audio Source Separation”. In: *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. URL: http://www.aes.org/e-lib/browse.cfm?elib=17123.

Schnass, Karin (2015). “A personal introduction to theoretical dictionary learning”. In: *Internationale Mathematische Nachrichten* 228, pp. 5–15.

Selesnick, Ivan (Apr. 2013). *Least Squares with Examples in Signal Processing*. URL: https://www.semanticscholar.org/paper/Least-Squares-with-Examples-in-Signal-Processing-1-Selesnick/7164db0fa52a1a7e1be91586dd02695394c34925.

Simpson, Andrew J R (Mar. 2015). "Probabilistic Binary-Mask Cocktail-Party Source Separation in a Convolutional Deep Neural Network". In: *arXiv.org*. arXiv: 1503.06962.

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 58.1, pp. 267–288.

Tošić, Ivana and Pascal Frossard (2011). "Dictionary learning". In: *IEEE Signal Processing Magazine* 28.2, pp. 27–38.

Turl, Alastair C (2015). "Robustness of ICA methods for clipping-distorted mixtures". MSc thesis. University of Birmingham.

— (2017). "Joint Blind Source Separation and Declipping: A Geometric Approach for Time Disjoint Sources". In: *IEEE International Symposium on Signal Processing and Information Technology*.

Vempala, Santosh and Ying Xiao (2014). "Max vs Min: Independent Component Analysis with nearly Linear Sample Complexity". In: *CoRR* abs/1412.2954. URL: http://arxiv.org/abs/1412.2954.

Vincent, Emmanuel, Rémi Gribonval, and C Fevotte (2006). "Performance measurement in blind audio source separation". In: *IEEE Transactions on Audio, Speech and Language Processing* 14.4, pp. 1462–1469. DOI: 10.1109/tsa.2005.858005.

Vincent, Emmanuel et al. (2014). "From Blind to Guided Audio Source Separation: How models and side information can improve the separation of sound". In: *IEEE Signal Processing Magazine* 31.3, pp. 107–115.

Welling, Max and Markus Weber (2001). "A Constrained EM Algorithm for Independent Component Analysis". In: *Neural Computation* 13.3, pp. 677–689.

Winter, S, H Sawada, and S Makino (2005). “Geometrical Interpretation of the PCA Subspace Approach for Overdetermined Blind Source Separation”. In: *EURASIP Journal on Applied Signal Processing* 2006.1, pp. 1–11.

Xie, Jun et al. (2014). “Single depth image super resolution and denoising via coupled dictionary learning with local constraints and shock filtering”. In: *2014 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 1–6.

Zhao, Huan et al. (2012). “Improved MFCC feature extraction combining symmetric ICA algorithm for robust speech recognition”. In: *Journal of multimedia* 7.1, pp. 74–81.