

SEPARATION AND ANALYSIS OF MULTICHANNEL SIGNALS

A Thesis
Presented to
The Academic Faculty

by

Robert Mitchell Parry

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
December 2007

SEPARATION AND ANALYSIS OF MULTICHANNEL SIGNALS

Approved by:

Dr. Irfan Essa, Committee Chair
College of Computing
Georgia Institute of Technology

Dr. Aaron Bobick
College of Computing
Georgia Institute of Technology

Dr. Charles Isbell
College of Computing
Georgia Institute of Technology

Dr. Gil Weinberg
College of Architecture
Georgia Institute of Technology

Dr. Sumit Basu
Knowledge Tools Group
Microsoft Research

Dr. Daniel Ellis
Department of Electrical Engineering
Columbia University

Date Approved: 5 October 2007

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
SUMMARY	xi
I INTRODUCTION	1
1.1 Brief Background	3
1.2 Impact Statement	6
1.3 Technical Contributions	6
II BACKGROUND	8
2.1 Independent Component Analysis	8
2.2 More Sources than Mixtures	9
2.2.1 Time-Frequency Masking	9
2.2.2 Spectrogram Factorization	10
2.2.3 Source Cancellation	11
2.2.4 Instrument Separation and Transcription	11
2.3 Convolved Mixtures	12
2.4 Time-Frequency Distributions	15
2.5 Source Number Estimation	17
III INCORPORATING REPETITIVE STRUCTURE FOR BLIND SOURCE SEP- ARATION AND DETECTION*	18
3.1 Independent Component Analysis	18
3.1.1 Second-Order Structure	19
3.2 Source Structure	21
3.2.1 Non-Gaussian Structure	22
3.2.2 Time-lagged Covariance	24
3.2.3 Time-varying Energy	25
3.2.4 Time-Frequency Sparseness	27

3.2.5	Discussion	29
3.3	Repetitive Structure	30
3.4	Time-Time Representations	31
3.5	Application to Blind Source Separation	33
3.5.1	Simulated Sample-based Repetition	33
3.5.2	Simulated Spectrum-based Repetition	35
3.5.3	Separation of Clarinet Recordings with Repetitive Structure . . .	38
3.6	Application to Source Detection	46
3.6.1	Detection of Spectrally Similar Sources	49
IV	SEPARATING MORE SOURCES THAN MIXTURES BY NON-NEGATIVE SPECTROGRAM FACTORIZATION*	53
4.1	Fundamental Technologies	55
4.1.1	Non-negative Spectrograms	55
4.1.2	Non-negative Spectrogram Factorization	56
4.1.3	NMF-based Non-negative Spectrogram Factorization	57
4.1.4	ICA-based Non-negative Spectrogram Factorization	58
4.2	Multichannel Extensions	60
4.2.1	Extending NMF-based NSF to Multiple Channels	60
4.2.2	Extending ICA-based NSF to Multiple Channels	63
4.2.3	Experiments	65
4.3	Incorporating Phase Information	70
4.3.1	Probabilistic Representation of the Non-negative Mixture Spec- trogram	71
4.3.2	Two Components	73
4.3.3	Extension to More Than Two Components	84
4.4	Putting It All Together	104
4.4.1	Application to Musical Audio	110
4.5	Application to Real Sounds	123
4.5.1	Bass and Organ Example	123

4.5.2	Bass, Vocals, and Organ Example	130
V	SUMMARY AND FUTURE WORK	142
APPENDIX A	ICA EXAMPLE: JOINT APPROXIMATE DIAGONALIZATION OF EIGENMATRICES	145
APPENDIX B	DERIVATION OF MULTICHANNEL ICA-BASED NON-NEGATIVE SPECTROGRAM FACTORIZATION	149
REFERENCES	156

LIST OF TABLES

1	Average maximum ISR for each algorithm in decibels as a function of repetition similarity as SNR in decibels (<i>i.e.</i> , $10 \log_{10}(\text{ISR})$)	35
2	Average maximum ISR for each algorithm in decibels as a function of difference in center frequency, Δf (<i>i.e.</i> , $10 \log_{10}(\text{ISR})$)	38
3	ISR for each algorithm in decibels for clarinet example	45
4	Activation sequence of sources	49
5	Distribution of Component Position Error (ISR)	69
6	Summary of detection rate and lowest estimation error for $R = [2, 10]$. . .	96
7	Instrument recording abbreviation definitions	107
8	Cost function performance for separating pairs of instrument components .	109
9	Cost function performance for separating bass, flute, and soprano saxophone from two mixtures	111
10	Cost function performance for separating three instruments from two mixtures	112

LIST OF FIGURES

1	Joint distribution of correlated variables	21
2	Joint distribution of whitened and independent variables	22
3	Probability density function for super-Gaussian, Gaussian, and sub-Gaussian signals	23
4	Autocorrelation function for a periodic signal	25
5	Time-varying energy for series of piano notes	26
6	Time-frequency representation for overlapping organ notes	28
7	Self-similarity matrix for “March of the Pigs” by Nine Inch Nails	31
8	Time-time distribution matrices for each pair of sources	37
9	Time-frequency distribution for three clarinets	39
10	Time-time distribution matrices between and within clarinets	40
11	Time-time (\overrightarrow{tt}) autoterms (in black) for clarinets example	41
12	Time-time (\overleftarrow{tt}) autoterms (in black) for clarinets example	42
13	Time-frequency autoterms (in black) for clarinets	43
14	Lagged autocorrelation structure for clarinets	44
15	Similarity between each clarinet’s autocorrelation	44
16	Local autocorrelation structure for clarinets	45
17	Normalized frequency of the sources	50
18	Generating the mixture of sources	50
19	Computing the collection functions	51
20	Activation function for each source	52
21	ICA in the time domain	54
22	ICA in the frequency domain	54
23	Multichannel factorization for NMF-based NSF	62
24	Multichannel factorization for ICA-based NSF	64
25	Components extracted from drums and piano using multichannel NMF- (<i>left</i>) and ICA-based NSF (<i>right</i>)	68

26	Extracted component envelopes, spectra, and positions for multichannel NMF-based NSF and three piano sources	70
27	Likelihood function for x when $c_1 = 2$ and $c_2 = 1$	74
28	Surface of $p(x c_1, c_2)$ as a function of c_1 and c_2 when $x = 1$	76
29	Surface of $p(x c_1, c_2)$ as a function of c_1 and c_2 with $x = 1$	76
30	Plots of the intermediate functions between $p(x c_1, c_2)$ and the optimization function	79
31	Contour plot of the cost function D_{smooth}	80
32	Scatter plot of bins for one representative trial	82
33	Histogram for all trials in units of 10^5	83
34	As the number of components increases, \mathbf{X} approaches a Rayleigh distribution	86
35	The shape of the likelihood functions derived from the 5 labeled cost functions for the case of two components and $x = 1$	87
36	Estimation error and detection rate for components drawn from a uniform distribution	90
37	Estimation error and detection rate initializing E_p , D_p , and D_s with the E_m solution.	91
38	Estimation error and detection rate initializing all methods with the true solution	92
39	Estimation error and detection rate for components drawn from a positive normal distribution	94
40	Estimation error and detection rate for components drawn from an exponential distribution	95
41	Estimation error and detection rate for components drawn from a uniform distribution	97
42	Estimation error and detection rate for components drawn from a positive normal distribution	98
43	Estimation error and detection rate for components drawn from an exponential distribution	99
44	Estimation error and detection rate for components initialized with true \mathbf{B} drawn from uniform distribution	100
45	Estimation error and detection rate for components initialized with true \mathbf{B} drawn from normal distribution	101

46	Estimation error and detection rate for components initialized with true B drawn from exponential distribution	102
47	The spectral shape of each of the 28 instrument recordings of middle C . . .	105
48	The amplitude envelope of each of the 28 instrument recordings of middle C	106
49	The lowest mean square error for each pair of instrument recordings	108
50	The relative difficulty of separating each instrument	108
51	The cost function that makes the minimum mean square error estimate for each pair of instruments	109
52	Original (<i>top</i>) and estimated spectral shapes for middle C on bass	113
53	Original (<i>top</i>) and estimated spectral shapes for middle E on bass	113
54	Original (<i>top</i>) and estimated amplitude envelopes for middle C on bass . . .	114
55	Original (<i>top</i>) and estimated amplitude envelopes for middle E on bass . . .	114
56	Original (<i>top</i>) and estimated audio signals for middle C on bass	115
57	Original (<i>top</i>) and estimated audio signals for middle E on bass	115
58	Original (<i>top</i>) and estimated spectral shapes for middle C on flute	116
59	Original (<i>top</i>) and estimated spectral shapes for middle E on flute	116
60	Original (<i>top</i>) and estimated amplitude envelopes for middle C on flute . . .	117
61	Original (<i>top</i>) and estimated amplitude envelopes for middle E on flute . . .	117
62	Original (<i>top</i>) and estimated audio signals for middle C on flute	118
63	Original (<i>top</i>) and estimated audio signals for middle E on flute	118
64	Original (<i>top</i>) and estimated spectral shapes for middle C on soprano sax- ophone	119
65	Original (<i>top</i>) and estimated spectral shapes for middle E on soprano sax- ophone	119
66	Original (<i>top</i>) and estimated amplitude envelopes for middle C on soprano saxophone	120
67	Original (<i>top</i>) and estimated amplitude envelopes for middle E on soprano saxophone	120
68	Original (<i>top</i>) and estimated audio signals for middle C on soprano saxophone	121
69	Original (<i>top</i>) and estimated audio signals for middle E on soprano saxophone	121
70	Original (<i>top</i>) and estimated positions for all six components	122

71	High energy points in the electric bass spectrogram	124
72	High energy points in the electric organ spectrogram	124
73	Overlapping high energy in the electric bass and electric organ	125
74	Three components extracted by E_m for bass and organ	126
75	Three components extracted by D_m for bass and organ	127
76	Three components extracted by E_p for bass and organ	127
77	Three components extracted by D_p for bass and organ	128
78	Three components extracted by D_s for bass and organ	128
79	Eight components extracted by E_m for bass and organ	131
80	Eight components extracted by D_m for bass and organ	132
81	Eight components extracted by E_p for bass and organ	133
82	Eight components extracted by D_p for bass and organ	134
83	Eight components by D_s for bass and organ	135
84	High energy points in the electric bass spectrogram	137
85	High energy points in the vocals spectrogram	138
86	High energy points in the electric organ spectrogram	138
87	Overlapping high energy in the electric bass, vocals, and electric organ . . .	139
88	Spectral shapes extracted by D_s for bass, vocals, and organ mixture	140
89	Amplitude envelopes extracted by D_s for bass, vocals, and organ mixture . .	141
90	Spatial positions extracted by D_s for bass, vocals, and organ mixture	141
91	Redundancy in cumulant matrix	146

SUMMARY

Music recordings contain the mixed contribution of multiple overlapping instruments. In order to better understand the music, it would be beneficial to understand each instrument independently. This thesis focuses on separating the individual instrument recordings within a song. In particular, we propose novel algorithms for separating instrument recordings given only their mixture.

When the number of source signals does not exceed the number of mixture signals, we focus on a subclass of source separation algorithms based on joint diagonalization. Each approach leverages a different form of source structure. We introduce repetitive structure as an alternative that leverages unique repetition patterns in music and compare its performance against the other techniques.

When the number of source signals exceeds the number of mixtures (*i.e.*, the underdetermined problem), we focus on spectrogram factorization techniques for source separation. We extend single-channel techniques to utilize the additional spatial information in multi-channel recordings, and use phase information to improve the estimation of the underlying components.

CHAPTER I

INTRODUCTION

Music recordings contain the mixed contribution of multiple overlapping instruments. In order to better understand the music, it would be beneficial to understand each instrument independently. This thesis focuses on separating the individual instrument recordings within a song. In particular, we propose novel algorithms for separating instrument recordings given only their mixture. In order to adapt technologies for source separation to music audio, we incorporate the repetitive structure in music, spatial information in stereo recordings, and phase information in audio spectra. Source separation in general is a broad field that applies to a wide variety of data. Although we apply the mathematics and theory derived in this thesis to musical audio, we believe that it could be applied to other types of data.

A motivating example for the separation of individual instrument tracks from a song recording is the potential to harness the advantages of both live and studio recording techniques in order to avoid the weaknesses of each. Live recording and studio recording are at opposite ends in the spectrum of recording techniques. In a studio setting, each instrument (or group of instruments) is isolated and recorded in its own track. Isolation booths or other physical barriers minimize the contribution of one instrument to another's track. Individual instrument tracks are then mixed to form the final recording. Constructing songs in this way affords great flexibility after a song has been recorded. For example, each instrument's volume and position in the stereo (or surround) image can be controlled independently. In addition, each track can be edited to affect timing, fix or remove mistakes, add effects, and even change pitch. New parts can be recorded at a later date and inserted into the mix. Therefore, one studio recording can result in many versions of a song, none of which were

performed in the traditional sense.

In-studio recording is expensive and generally reserved for more accomplished or established musicians. Currently, the only way to record isolated instrument tracks without the benefit of studio isolation booths is to record one track at a time. For example, the bass is recorded first, the drums second, guitar third, etc. Each track is recorded while a musician plays along to music through headphones. This sequential approach makes isolated recording possible but lacks the comfort and naturalness of live recording. In addition, expressiveness and improvisation are necessarily limited because of the rigid timing of previously recorded material.

In contrast to studio recording, live recording provides the musicians ultimate freedom during recording. For example, they have the comfort of sharing the same physical space and playing together and improvising without constraints. Live recording is characterized by a combined recording and mixing phase. Microphones placed at a distance from the instruments capture all of the instruments at once. The loudness of each instrument at each microphone depends on the instrument’s loudness and position relative to the microphone. The resulting recording can then be sent to a pair of speakers for the stereo effect. However, once the recording is finished, there is little that can be done to change it. One small mistake requires rerecording an entire song.

We are motivated by the potential to allow musicians the freedom of expression afforded to live recording while allowing additional flexibility from studio-style mixing. Because live recordings are already mixed, this thesis focuses on the task of “unmixing” the underlying instrument recordings from the mixture. The technical contributions in this thesis approach this goal by leveraging the repetitive structure evident in musical recordings (Chapter 3) and enhancing spectrogram factorization techniques for separating more instruments than microphones (Chapter 4).

The rest of this chapter briefly discusses the background, potential impact, and technical contributions. Chapter 2 provides more background with related work. In Chapter 3, we

review a subclass of source separation algorithms based on joint diagonalization. Each approach leverages a different form of source structure. We introduce repetitive structure as an alternative that leverages unique repetition patterns in music and compare its performance against the other techniques. In Chapter 4, we focus on the underdetermined problem of separating more source signals than mixture signals. We extend single-channel source separation techniques to utilize the additional spatial information in multichannel recordings. In addition, we use information about the phase in audio spectrograms to improve the estimation of the underlying spectral components that combine to form the mixture spectrogram. Finally, in Chapter 5 we summarize our contributions indicating directions for future work.

1.1 Brief Background

Generally, increasing the separation of the instruments during the unmixing phase leads to increased flexibility during remixing. Even if each instrument track contains sounds from other tracks, there is still flexibility in placement. For example, if the amplitude of the first source is α times the amplitude of the second source in the left channel and vice versa in the right channel, we have complete freedom in setting the amplitude and position of one source. However, after doing this, the left and right amplitude of the other source has a limited range. Let the amplitude of the i th source in the left and right channel be l_i and r_i , respectively. If the amplitudes are related as follows:

$$l_1 = \alpha l_2 \tag{1}$$

$$r_2 = \alpha r_1 , \tag{2}$$

and we have already set the amplitude and position of source 1 (*i.e.*, l_1 and r_1), source 2 is limited as follows:

$$l_1/\alpha \leq l_2 \leq \alpha l_1 \tag{3}$$

$$r_1/\alpha \leq r_2 \leq \alpha r_1 . \tag{4}$$

The goal of separation is to make α as large as possible thereby increasing the range of values for the second source.

In order to separate the instruments from a particular recording, we draw from the source separation literature. However, instrument separation and source separation in general are unsolved problems except in restricted scenarios. Early approaches use domain knowledge about instruments to separate them. For instance, knowledge of frequency and amplitude modulations, non-overlapping frequency ranges, characteristic attack, or spectral templates of instruments in the mixture inform separation algorithms [100]. More general formulations include blind source separation (BSS) and computational auditory scene analysis (CASA). Blind source separation is characterized by separating underlying source signals without prior knowledge of them, while CASA focuses on emulating human auditory perception [31,34,36,50,105,107]. This work focuses on techniques that leverage spatial separateness as well as other forms of structure in music recordings.

Independent component analysis (ICA) is a class of algorithms for BSS [58]. ICA requires at least as many mixtures as sources and a known and unchanging number of sources. In general, we expect the number of sources to outnumber the number of microphones. Although we will know the total number of sources, they will not always be playing. Therefore, within a recording ICA can separate the sources when their number does not exceed the number of microphones. An important first step is to determine which and how many sources are active at each point in time. Source number estimation is still an unsolved problem, although several solutions have been proposed [6]. We introduce a novel approach for source detection based on repetitive structure in Section 3.6.

The most common formulation of ICA employs an instantaneous mixing model that assumes each source arrives at each microphone at the same time and that there are no reflections in the environment. In real recording environments the different distance from each source to each microphone introduces a time-delay and the reflections in the environment cause reverberation (*i.e.*, convolved mixtures). In addition, ICA requires that the

number of source signals not exceed the number of mixture signals. Separating convolved mixtures or underdetermined mixtures represent very challenging unsolved problems in the source separation literature. This thesis addresses underdetermined and instantaneous mixtures but not convolved mixtures. Part of the reason to focus on instantaneous mixtures is that reverberation caused by the recording environment changes the aesthetics of the recording and is often a desirable quality. Although many applications attempt to diminish this effect (*e.g.*, to improve the intelligibility of speech), we want to preserve it.

Even though this work focuses on instantaneous mixtures, we discuss four ways in which this work is relevant to convolved mixtures. First, convolutive source separation is equivalent to multiple instantaneous separation problems in the frequency domain [108]. Therefore, algorithms for instantaneous source separation, such as those we present in Chapter 3, can be applied to each complex frequency channel independently to separate convolved mixtures. Second, single-channel mixtures can be regarded as instantaneous mixtures of sources that happen to contain reverberations. Because we want to preserve these reverberations, non-negative spectrogram factorization techniques such as those we propose in Chapter 4 can estimate source components including the reverberations. Third, a carefully designed microphone setup can turn a reverberant mixing environment into an approximately instantaneous mixing environment. Using a coincident boundary microphone removes the relative delay between microphones and magnifies the direct path signal thereby reducing the relative contribution of reverberation. In experimental tests, instantaneous separation algorithms outperform the convolutive separation algorithms for this microphone setup [103]. Finally, the joint diagonalization approaches we discuss in Chapter 3 can all be generalized to convolutive mixtures using joint block-diagonalization [39].

1.2 Impact Statement

Our motivating example of separating instrument recordings from a live recording most directly affects recording musicians. Perfect separation provides an ideal solution for combining the freedom of expression during live recording sessions and the flexibility of instrument placement and volume during the mixing process. However, any level of separation increases the flexibility during remixing.

Because of the nature of the instrument sources we want to separate, we incorporate a novel form of source structure for source separation. Music contains repetitions that can simplify separation. This repetitive structure is not limited to music and exists in other audio signals such as speech and natural recordings. Words, syllables, and phonemes repeat in a conversation. The sounds of keyboards, telephones, and printers permeate an office building. These repetitions inform the separation process. Even when the number of sources exceeds the number of recordings, we can leverage repetitive structure to inform a source detection algorithm. This work adds to the extensive literature on source separation and detection.

Separating live music recordings into instrument tracks also potentially benefits music information retrieval research. Music analysis algorithms excel when applied to a single instrument recording, yet are typically confounded by overlapping instruments. Separating the instruments as preprocessing step would likely improve the performance of these algorithms. In addition, while musical scores often exist for studio recorded music, some world music is never written and only exists as live recordings. Stereo recordings of this type may allow separation of instruments for further analysis and transcription.

1.3 Technical Contributions

This work has led to the following technical contributions:

- When the number of sources does not exceed the number of mixtures, we incorporate the unique long-term repetitive structure of sources to separate them. We present a novel source separation algorithm based on spatial time-time representations that

capture the repetitive structure in audio. We show that repetitive structure and source dissimilarity are sufficient to separate source signals [88].

- We address the issue of source detection when more sources than mixtures overlap in time and frequency. We show that repetitive structure in the form of time-time correlation matrices informs when each source is active [90].
- We extend single channel source separation algorithms based on spectrogram factorization to apply to multiple mixture signals. We introduce novel factorizations of magnitude spectrograms from multiple recordings and derive update rules that extend ICA- and NMF-based spectrogram factorization to concurrently estimate the spectral shape, amplitude envelope and spatial position of each component. We show that estimated component positions are near the position of their corresponding source, and show advantages and limitations of the approach for a three piano mixture [89].
- We investigate the role of phase in spectrogram factorization techniques used for single channel source separation. Typically the phase information is discarded but we show that by introducing a probabilistic representation of phase, we can improve the estimation for two source components [91].
- We incorporate a probabilistic representation of phase for the case of an arbitrary number of source components and derive a novel cost function. This cost function improves the estimation of the underlying source components but is more affected by detection errors [92].

CHAPTER II

BACKGROUND

In this chapter we review previous work dealing with source separation, starting with the instantaneous linear model and discussing approaches for convolutive, underdetermined, and nonlinear mixtures. The classic instantaneous mixing model dictates that the M mixture signals, $x_i(t)$, are a linear combination of the N source signals, $s_j(t)$:

$$x_i(t) = a_{i1}s_1(t) + a_{i2}s_2(t) + \cdots + a_{iN}s_N(t) . \quad (5)$$

Stacking the mixture and source signals into time-varying vectors produces the matrix-vector representation:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) , \quad (6)$$

where \mathbf{A} is the $M \times N$ mixing matrix with elements a_{ij} . As long as the number of source signals does not exceed the number of mixture signals ($N \leq M$), the inverse or pseudoinverse of matrix \mathbf{A} recovers the source signals from the mixtures:

$$\mathbf{s}(t) = \mathbf{A}^\# \mathbf{x}(t) , \quad (7)$$

where $^\#$ is the Moore-Penrose pseudoinverse.

2.1 Independent Component Analysis

Independent component analysis (ICA) is a class of algorithms that estimate the source signals or “unmixing” matrix $\mathbf{A}^\#$ leveraging the independence of the sources. The classic approach is to treat each signal as a random variable and focus on the non-Gaussian distribution of the sources. ICA algorithms optimize different criteria such as minimizing mutual information between sources [5, 32], maximizing the combined information in sources [13], and high-order decorrelation [24]. Reviews of ICA are available from several

sources [22, 58, 110]. Additionally, ICA can be constrained to favor particular positions or components [75, 76, 79, 78].

The previous techniques assume that the sources have non-Gaussian probability density functions. They distinguish between source and mixture signals based on their closeness to the Gaussian distribution. Because the mixture signals are a sum of independent random variables (the sources), the mixtures are more Gaussian than the sources due to the central limit theorem. The sources are recovered by transforming the mixtures so that they are as non-Gaussian as possible. If the sources already have a Gaussian distribution, the sources and mixtures cannot be distinguished. In this case some other form of structure must be present.

Treating the signals as random variables ignores any time-varying characteristics of the signals. Other algorithms leverage the time structure of the source signals, including time-varying energy profiles, autocorrelation, and sparseness in the time-frequency domain. We discuss these approaches in more detail in Chapter 3.

2.2 More Sources than Mixtures

When the number of sources exceeds the number of mixtures, it is not possible to construct an unmixing matrix that separates the sources as shown in Equation 7. In the extreme case, only one mixture signal is available. Some techniques incorporate specific information about the sources.

2.2.1 Time-Frequency Masking

A general approach for single-channel separation is time-frequency masking. Using a time-frequency representation such as the short-time Fourier transform (STFT), the sources can be separated by applying a mask that removes the contribution of all other sources [101, 8]. The inverse STFT applied to each masked STFT provides each time-varying source signal. The difficulty is determining which time-frequency bins belong to each source. Roweis

uses isolated source recordings to train an HMM for every source, then constructs a factorial HMM to represent mixtures of these sources [101]. Then the most likely state sequence for each source in the factorial HMM given the mixture signal determines which source is (more) active at each time-frequency point. This is an example of a separation technique that needs specific source information to perform. Alternatively, a semi-supervised approach models how a harmonic source changes over time without specific information about each source [98]. If more than one mixture signal is available, speculation about the sources can be avoided. Instead, the spatial information at each time-frequency point determines its assignment to a source.

Algorithms based on the DUET approach [62, 124] cluster time-frequency points according to the amplitude and delay between two STFTs assuming exactly one source is active at each point. The cluster centers approximate the mixing parameters for each source in the anechoic model and the grouping assigns time-frequency points to source signals. Alternatively, time-frequency representations such as the pseudo Wigner distribution compute the correlations between signals at time-frequency points. If exactly one source is active at a time-frequency point, these spatial correlations reveal its spatial position [85]. Similar clustering on spatial position provides mixing parameters for the instantaneous model. Inverting the masked pseudo Wigner distributions provides the source signal estimates.

If more than one but not more than M sources are active at the same time-frequency point, the contributions of each source can be recovered using the mixing parameters for the active sources [82]. The difficulty here is determining which subset of sources is active at each time-frequency point. However, if the number of sources at a time-frequency point is greater than the number of mixtures, M , there is once again no hope in separating them.

2.2.2 Spectrogram Factorization

Spectrogram factorization provides a way to decompose a single mixture spectrogram into a collection of components that represent very simple signals roughly corresponding to

musical notes or voiced speech. Applications include source separation and music transcription [2, 41, 20]. First, the signal is transformed into the time-frequency domain via an STFT. The phase information is discarded yielding the absolute value (magnitude) or absolute square (power) spectrogram. Then a matrix factorization method such as ICA or non-negative matrix factorization is applied. This provides a number of components comprising a static spectral shape and amplitude envelope. Although each component is not complex enough to represent a real source, their combination can. For example, each piano note roughly corresponds to one component. Therefore, the 88 keys on a piano are roughly captured by 88 spectral components. Spectrogram factorization will be discussed in more detail in Chapter 4. The advantage of spectrogram factorization is that it does not require specific source models and it handles multiple overlapping components. This benefit comes at the cost of the expressiveness of each source, requiring each source to be the combination of multiple signals with static spectral shape.

2.2.3 Source Cancellation

Source cancellation is a related approach that applies when there are more sources than mixture signals [10]. The most popular of which is vocal cancellation for karaoke systems. If a source's position is known and instantaneously mixed, it can be subtracted from a stereo recording. For example, if a source is scaled by α in the left speaker and β in the right, it can be removed to generate the mono recording $M = L - \frac{\alpha}{\beta}R$. If more mixture channels are present, more sources can be canceled. Even if the number of simultaneous sources is greater than the number of mixtures, one fewer source than mixtures can be canceled.

2.2.4 Instrument Separation and Transcription

Another related problem is automatically transcribing a music recording into the notes, onsets, and durations required to synthesize the composition [66, 31, 34, 35, 37]. Monophonic music requires at most one instrument and one note playing at a time. Therefore, standard pitch detection and onset detection techniques apply. When multiple notes or instruments

play simultaneously the problem is more complicated [96]. For instance, different notes in a song often have overlapping harmonics and therefore similar spectra. Identifying the right notes given the evidence is a daunting task.

Advanced pitch detection attempts to identify multiple pitches at once [65, 64], while blackboard systems combine information and incorporate domain knowledge to disambiguate simultaneous notes [81, 80]. A set of knowledge sources provides evidence for different hypotheses in the system. In the end, one hypothesis wins out as the most likely candidate. For instance, harmonics occurring at integer multiples of a frequency provides evidence for a fundamental at that frequency. Some approaches integrate psychological grouping principles such as temporal and frequency proximity, common onset and offset, harmonicity, and common frequency movement [50]. In addition, practical knowledge of the sources such as frequency and amplitude modulations, non-overlapping frequency ranges, characteristic attack, or spectral templates inform separation [100]. Alternatively, a multiple-cause model can simultaneously learn the spectrum of notes and their amplitudes as a function of time [67], much like the spectrogram factorization approaches discussed in Chapter 4. Others employ harmonic modeling [27, 49, 117]. Once the notes are separated, they may be combined into instrument streams [63, 102].

2.3 *Convolved Mixtures*

A common assumption of ICA algorithms is that the sources are mixed simultaneously (*i.e.*, there are no delays or reverberation). However, reverberation is introduced in real recordings when a source sound may travel in multiple paths to the same microphone. This is called the multipath problem or convolutive mixing [70, 114]. The multipath problem is formulated as follows:

$$x_i(t) = \sum_j h_{ij}(n) * s_j(t) , \quad (8)$$

where each mixture, $x_i(t)$, is the sum of the sources $s_j(t)$ convolved with an FIR filter $h_{ij}(n)$. There is a unique filter for every source-mixture pair. If there is only one source,

ICA algorithms can solve for the FIR filter by assuming the source is independent across time [13]. However, the general case is much more difficult.

One approach to the multipath problem is to generalize the existing ICA framework to incorporate FIR filter matrices [70, 71]. The standard formulation of ICA given in Equation 5 uses a scalar mixing matrix \mathbf{A} . Lambert extends this so that each element of the mixing matrix is a FIR filter. If each filter contains exactly one nonzero entry at zero lag, this reduces to a simultaneous mixture. Otherwise, existing ICA algorithms may be applied using a FIR matrix algebra where FIR matrix multiplication is interpreted as convolution. It is natural to process FIR matrices in the frequency domain because convolution becomes multiplication. This leads to other frequency domain multipath blind source separation techniques.

Smaragdis converts the multipath problem into a series of instantaneous ICA problems [108, 109]. Each mixture is converted into the frequency domain using the short-time Fourier transform. Each complex time-varying frequency channel is an instantaneous mixture of the sources. The frequency domain components of each FIR matrix at that frequency compose the mixing matrix:

$$\mathbf{X}_f = \mathbf{A}_f \mathbf{S}_f, \quad (9)$$

where \mathbf{X}_f and \mathbf{S}_f are the time-varying frequency domain mixture and source signals at frequency f , and the mixing matrix \mathbf{A}_f contains the frequency domain coefficients of each FIR filter at frequency f . Independent components are extracted from each frequency bin and the FIR matrix is assembled. However, because ICA algorithms are permutation invariant, the filter components will not generally align across frequencies. Therefore, Smaragdis suggests zero-padding the FFT so that the frequency spectrum is smoothly varying and adjacent ICA calculations are likely to converge on the same permutation. Additionally, he imposes a smoothness constraint on the unmixing matrix computed at adjacent frequency bins. Algorithms based on this approach [60] differ in how they solve the permutation and amplitude ambiguity.

Similarly, Pham et al. use the short-time periodogram for multivariate signals [94]. The periodogram is then smoothed over adjacent frequencies. The authors leverage the nonstationarity of the source signals by jointly diagonalizing a set of frequency-specific correlation matrices taken from different blocks in time. A matrix that jointly diagonalizes these matrices contains the FIR filter components at that frequency. Once again, permutations are a problem. To disambiguate the permutations, the authors rely on smoothly varying FIR coefficients. Other approaches also leverage the nonstationarity of sources [69, 93].

Abdallah emulates the frequency domain approach in the time domain using ICA [1]. Each mixture is partitioned into short frames represented as a time-varying vector of length L . Each vector mixture is then stacked so that the combined mixed signal is $\mathbf{x}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \dots, \mathbf{x}_m(t)]^T$, where $\mathbf{x}_i(t) = [x_1(t), x_2(t), \dots, x_L(t)]^T$. The vector \mathbf{x} contains LM mixed signals. Applying an ICA algorithm to these stacked frames provides m basis vectors of length L for each independent component. Basis vectors can be clustered by geometric dependency and combined to form separated sources. Alternatively, the residual dependency between components may be used to form a topographic ordering with which to cluster components [59].

A number of joint diagonalization algorithms capture different structure in the source signals within multiple spatial correlation matrices. For the instantaneous case, these matrices are $M \times M$. Févotte and Doncarli [39] show that all instantaneous joint diagonalization algorithms can be generalized to the multipath problem by constructing $LM \times LM$ correlation matrices that capture the source structure between L time-lags of each of the signals. The joint block-diagonalization of these multipath correlation matrices results in the estimation of the sources up to an unknown filter.

The microphone setup also plays an important role in separation algorithms. If the microphones are close enough together the time delay between microphones is captured by the phase of the STFT allowing the DUET-style algorithms to estimate sources with the

same amplitude but different delay. Alternatively, coincident boundary microphones remove the delay altogether and amplify the direct path signal resulting in mixtures that are dominated by the first tap in each FIR filter. These mixtures are approximately instantaneous and instantaneous separation algorithms outperform their multipath counterparts for a preliminary experiment [103, 104].

2.4 Time-Frequency Distributions

Time-frequency distributions provide an alternative way for us to represent mixture signals and provide insight into new ways to separate them. There are many ways to represent the time-varying frequency content in a signal [52, 54]. We have already mentioned the short-time Fourier transform and spectrogram. The short-time Fourier transform is a linear time-frequency distribution (TFD) [52]:

$$\text{STFT}(t, f) = \int x(\tau)g^*(\tau - t)e^{-j2\pi f\tau}d\tau, \quad (10)$$

where g is a short time window that localizes the Fourier transform. A quadratic form of this is the short-time power spectrum, also known as the spectrogram¹ [52]:

$$\text{SPEC}(t, f) = |\text{STFT}(t, f)|^2. \quad (11)$$

Quadratic TFDs are 2-dimensional functions of the energy in a signal. Because of the uncertainty principal, energy cannot be pinpointed in time and frequency. Instead, a quadratic TFD estimates the energy in a time-frequency region. The spectrogram samples linearly in time and frequency, computing energy in identically shaped rectangles in time-frequency. The wavelet transform is another TFD where every sample covers the same area, but differently shaped rectangles in the time-frequency plane. As frequency increases, the sampled rectangle becomes narrower along the time axis and wider along the frequency axis. Both

¹Although the literature on time-frequency analysis refers to the absolute square of the STFT as the spectrogram, the term spectrogram commonly refers to all STFT based representations such as the STFT itself, its absolute value, or its absolute square. We use “magnitude spectrogram” or “power spectrogram” to differentiate between the two and “spectrogram” when the meaning is understood from the context.

the spectrogram and the wavelet transform apply to a single signal. However, TFDs based on the Wigner distribution (WD) can be computed *between* signals [54]:

$$\text{WD}_{x_1 x_2}(t, f) = \int x_1(t + \frac{\tau}{2}) x_2^*(t - \frac{\tau}{2}) e^{-j2\pi f \tau} d\tau. \quad (12)$$

When $x_1 = x_2$, the WD replaces the window in the short-time Fourier transform with a time-reversed version of the signal itself. When $x_1 \neq x_2$, we would like the WD to represent the shared energy between signals at each time-frequency point. Unfortunately, because of the uncertainty principal the WD cannot be interpreted as an energy distribution and is often negative. To address this issue, the pseudo Wigner distribution localizes the computation in the time domain, creating a “short-time” Wigner distribution [29]:

$$\text{PWD}_{x_1 x_2}(t, f) = \int h(\tau) x_1(t + \frac{\tau}{2}) x_2^*(t - \frac{\tau}{2}) e^{-j2\pi f \tau} d\tau. \quad (13)$$

Localizing the computation in the time domain smoothes the data along the frequency axis. In addition, the smoothed pseudo Wigner distribution smoothes along the time axis, further improving its interpretation as an energy distribution [29]:

$$\text{SPWD}_{x_1 x_2}(t, f) = \int h(\tau) \int g(s - t) x_1(s + \frac{\tau}{2}) x_2^*(s - \frac{\tau}{2}) ds e^{-j2\pi f \tau} d\tau. \quad (14)$$

Belouchrani and Amin view time-frequency distributions computed between every pair of mixture signals as a spatial correlation matrix for every time-frequency point [19]. After whitening, the authors identify time-frequency points containing only one source as spatial correlation matrices with rank one, called autoterms. Autoterms for the same source have the same principal eigenvector. Belouchrani and Amin jointly diagonalize the autoterm matrices for blind source separation. Of course, this requires that every source have at least one autoterm. Other work improves the way autoterms are selected [40, 55]. In addition, more sources than mixtures may be extracted if there is minimal overlap in their time-frequency distributions [85, 124]. The source number is estimated by the number of unique autoterms.

We adapt the pseudo Wigner distribution so that it captures repetitive structure in Chapter 3. Specifically, we extend the joint diagonalization of spatial time-frequency correlation matrices to spatial time-time matrices. In Chapter 4, we use the magnitude and power spectrogram to estimate spectral components in the underdetermined mixing problem.

2.5 Source Number Estimation

In order to separate sources using any of the preceding techniques, we must estimate the number of sources (or components) in a mixture. Casey uses principal component analysis to keep a fraction of the total variance in the mixtures [25]. He chooses the source number corresponding to the size of the most significant set of eigenvectors that explains a specified amount of the variance in the data. For time-frequency distributions, the number of unique autoterms indicates the number of sources when each source has at least one autoterm. Both techniques use singular values to inform the process.

In general singular values can be used to approximate the rank of a matrix [68]. Aouada et al. review three common techniques for source number estimation [6]. These techniques include the minimum description length, Bayesian information criterion, and the use of Gershgorin radii [26, 84, 120, 121]. In addition, simultaneous denoising and source number estimation are provided by the discrete wavelet transform [87]. Support vector machines have been used to estimate the number of sources for convolved mixtures [122]. Blind source separation with changing source number is also considered [77, 123]. One contribution of this work is to use repetitive structure to inform a source detection algorithm that estimates when each source is active in the mixture (Section 3.6).

CHAPTER III

INCORPORATING REPETITIVE STRUCTURE FOR BLIND SOURCE SEPARATION AND DETECTION*

Blind source separation techniques attempt to decompose multiple mixture signals into their constituent sources. For instantaneous mixtures, this amounts to inverting the following mixing system:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) , \quad (15)$$

where $\mathbf{x} = [x_1(t), \dots, x_M(t)]^T$ is a time varying vector representing the mixture signals, $x_i(t)$, $\mathbf{s} = [s_1(t), \dots, s_N(t)]^T$ represents the source signals, $s_i(t)$, $\mathbf{n}(t)$ is white noise, and \mathbf{A} is the $M \times N$ real mixing matrix. Each mixture signal, $x_i(t)$, is a weighted sum of the source signals. The weights are stored in the i th row of matrix \mathbf{A} . The “location” of each source, $s_j(t)$, indicates how it is spread across the different mixtures and is contained in the j th column of \mathbf{A} . The goal is to estimate \mathbf{A} , \mathbf{A}^{-1} , or $\mathbf{s}(t)$ given only $\mathbf{x}(t)$ without specific knowledge of the sources or mixing system.

3.1 Independent Component Analysis

Independent component analysis (ICA) leverages the statistical independence of source signals to separate them. One major limitation to using ICA for BSS is that there must be at

*This chapter contains parts of the following copyrighted material:

PARRY, R. M. and ESSA, I., “Blind source separation using repetitive structure,” in *Proceedings of International Conference on Digital Audio Effects*, (Madrid, Spain), pp. 143–148, September 2005.
©2005 by the authors.

PARRY, R. M. and ESSA, I., “Source detection using repetitive structure,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Toulouse, France), pp. 1093–1096, May 2006.
©2006 IEEE. Reprinted with permission.

least as many mixture signals as source signals. Unfortunately, this restriction and the independence assumption are not enough to blindly separate sources. In addition, sources must exhibit some form of structure, for example, non-Gaussian structure or temporal structure.

3.1.1 Second-Order Structure

All interesting signals contain 2nd-order structure (*i.e.*, non-zero variance). Because the source signals are independent and therefore uncorrelated, their covariance matrix is diagonal. The diagonality of this matrix captures their 2nd-order structure (*i.e.*, it measures something that each source has but mixtures do not). Without loss of generality, the sources are assumed to have zero mean and unit variance. Therefore, the source covariance is the identity matrix:

$$\mathbf{R}_s = E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}_N, \quad (16)$$

where E is the expectation operator, H is the conjugate transpose, and \mathbf{I}_N is the $N \times N$ identity matrix. The mixing matrix \mathbf{A} introduces second order correlations so that the covariance of \mathbf{x} is not diagonal:

$$\mathbf{R}_x = E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{A}\mathbf{R}_s\mathbf{A}^H = \mathbf{A}\mathbf{A}^H. \quad (17)$$

Therefore, a typical first step for separation algorithms is to remove this correlation using principal component analysis. Principal component analysis provides a translation and rotation that makes the mixtures uncorrelated, essentially diagonalizing this covariance matrix.

ICA can be seen as an extension of principal component analysis (PCA). PCA eliminates 2nd-order cross-correlations in the data by diagonalizing the covariance matrix. Statistical independence requires n th order decorrelation (for all integers n). Therefore, PCA can be used as a preprocessing step for ICA. If the desired number of sources is less than the number of mixtures, the directions of lesser variance can be removed during the PCA step. Additionally, under the Gaussian white noise assumption, the mean variance of the

removed dimensions is used to estimate the variance of the noise in the mixture. The variance of the noise can then be subtracted from the covariance matrix to diminish its effect. Finally, the variances of the projected data are normalized so that the covariance matrix is the identity matrix and the sources have unit variance. The $N \times M$ whitening matrix \mathbf{W} accomplishes this precisely and can be computed from an eigen-decomposition of \mathbf{R}_x :

$$\mathbf{z}(t) = \mathbf{W}\mathbf{x}(t) \quad (18)$$

$$\mathbf{R}_z = \mathbf{W}\mathbf{R}_x\mathbf{W}^H = \mathbf{W}\mathbf{A}\mathbf{A}^H\mathbf{W}^H = \mathbf{I}_N . \quad (19)$$

Now any rotation of the whitened mixtures, \mathbf{z} , produces uncorrelated signals. If $\mathbf{U} = \mathbf{W}\mathbf{A}$, \mathbf{U} is unitary (due to Equation 19). This reduces the problem of estimating \mathbf{A} to the estimation of an $N \times N$ unitary rotation matrix \mathbf{U} that reveals the sources:

$$\hat{\mathbf{A}} = \mathbf{W}^\# \mathbf{U} \quad (20)$$

$$\hat{\mathbf{s}} = \mathbf{U}^H \mathbf{z}(t) . \quad (21)$$

What makes each ICA algorithm different is how to estimate the rotation that makes the signals statistically independent.

3.1.1.1 ICA Example¹

Figure 1 provides a visual depiction of this process in two dimensions. The source data are two-dimensional random variables from a uniform distribution in the interval $[0, 1]$. These data are rotated and scaled by the mixing matrix,

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} ,$$

to have the joint distribution shown in Figure 1(a). The PCA step identifies the high-variance directions and rotates them so they are on the primary axes (Figure 1(b)). Whitening makes the variance in each dimension the same (Figure 2(b)).

¹This example is based on that of Paris Smaragdis in his doctoral dissertation [110].

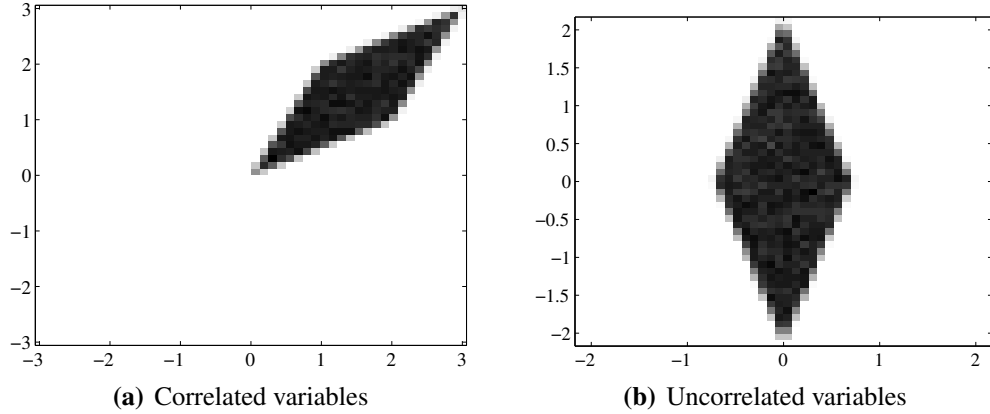


Figure 1: Joint distribution of correlated variables

In this case, PCA identifies two uncorrelated components in the mixture. However, statistical independence means that knowing the value of one source provides no information about the other source. As can be seen in Figure 2(a), knowing the value of the y-dimension limits the range of values in the x-dimension. Therefore, these sources are not yet independent. However, because of the whitening step, we can now rotate the axes freely without affecting the correlation between sources. ICA attempts to find the best rotation that provides a maximally independent set of sources, as depicted in Figure 2(b).

3.2 Source Structure

The first ICA algorithms focused on non-Gaussian structure for source separation. That is, sources that do not have a Gaussian probability distribution exhibit structure in the form of n th order correlations, where $n > 2$. Algorithms that leverage non-Gaussian structure optimize different criteria such as minimizing mutual information between sources [5, 32], maximizing the combined information in sources [13], and fourth-order decorrelation [24]. Algorithms that apply to Gaussian signals can leverage time-varying energy [83], lagged covariance [17], or time-frequency sparseness [19]. Additionally, ICA can be constrained to favor particular positions or components [75, 76, 79, 78]. Reviews of ICA are available from several sources [58, 12, 22, 110].

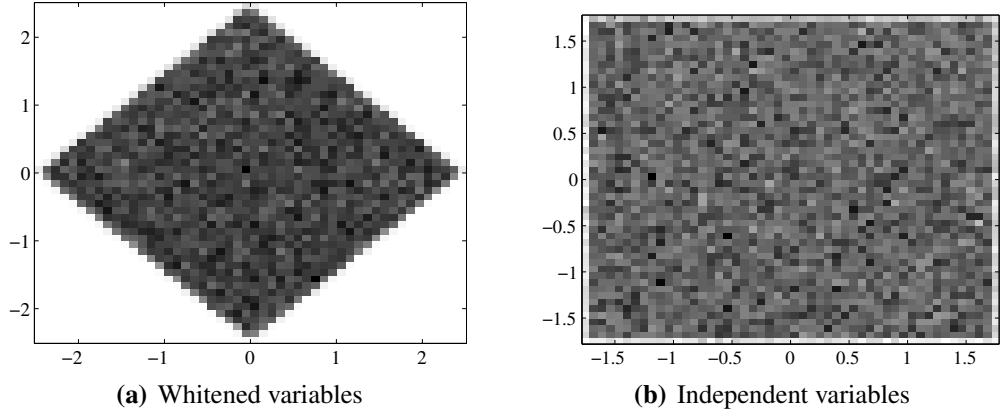


Figure 2: Joint distribution of whitened and independent variables

Cardoso and Souloumiac [24] introduce the idea of diagonalizing multiple correlation matrices in order to maximize the independence of the estimated source signals. When taken as a general source separation strategy, the *joint diagonalization* approach can be applied to multiple types of source structure. This approach is a generalization of principal component analysis that applies to a set of matrices instead of a single covariance matrix.

After whitening, the second step of joint diagonalization ICA algorithms is to estimate a set of correlation matrices that are diagonal for the sources and non-diagonal for the mixtures. These correlation matrices capture structural information about the sources that inform separation.

3.2.1 Non-Gaussian Structure

If the source signals do not have a Gaussian probability density function (*e.g.*, they are super-Gaussian or sub-Gaussian), they contain higher-order correlations that can be used for separation. Figure 3 shows super- and sub-Gaussian probability density functions compared to a Gaussian. A signal drawn from a super-Gaussian distribution is more peaked at zero and has flatter tails. A sub-Gaussian distribution is flatter at zero and has longer tails.

In the same way a covariance matrix captures the 2nd-order structure of the sources, Cardoso and Souloumiac [24] use multiple cumulant matrices to capture the 4th-order

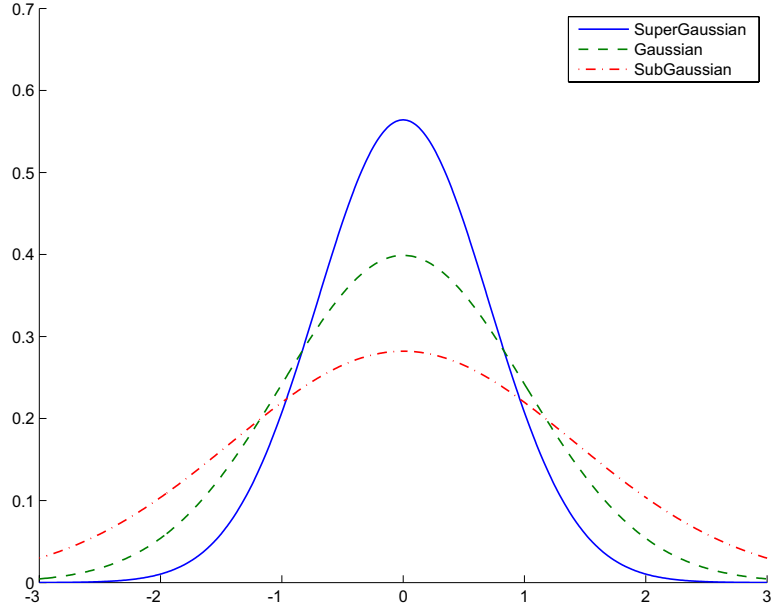


Figure 3: Probability density function for super-Gaussian, Gaussian, and sub-Gaussian signals

structure of non-Gaussian sources. Their JADE algorithm operates on an $N \times N \times N \times N$ cumulant tensor, \mathbf{Q}_z :

$$\mathbf{Q}_z(i, j, k, l) = \text{Cum}(z_i, z_j^*, z_k, z_l^*), \quad 1 \leq i, j, k, l \leq N, \quad (22)$$

where $*$ indicates the complex conjugate. If the sources are independent, the tensor is diagonal. In order to diagonalize the tensor, Cardoso and Souloumiac instead approximately diagonalize each of the $N^2 N \times N$ matrix slices of the cumulant tensor:

$$[\mathbf{R}_z^{4th}(i, j)]_{kl} = \text{Cum}(z_i, z_j^*, z_k, z_l^*), \quad (23)$$

where 4th labels this as a 4th-order correlation matrix. The unitary matrix \mathbf{U} that jointly diagonalizes the matrix slices is estimated by maximizing the following criterion [24]:

$$\sum_r |\text{diag}(\mathbf{U}^H \mathbf{N}_r \mathbf{U})|^2, \quad (24)$$

where the \mathbf{N}_r are the matrices to be diagonalized. In this case \mathbf{N}_r is one of the $\mathbf{R}_z^{4th}(i, j)$. The mixing matrix and sources can then be estimated from Equations 20 and 21.

Methods based on non-Gaussianity do not depend on the ordering of the samples. However, if the data is a time-varying signal the ordering of the data may contain valuable information. For example, time structure has been utilized for source separation in the form of time-lagged covariance [17], time-varying energy [83], and time-frequency sparseness [19]. Each of these can be implemented as a joint diagonalization algorithm.

3.2.2 Time-lagged Covariance

If the sources have a Gaussian distribution, there are no higher-order statistics between sources. In this case, time structure can be utilized. Belouchrani et al. [17] use lagged autocovariance to separate source signals in their SOBI algorithm. When the sources are time-varying signals it is often the case that they have 2nd-order autocorrelations at time-lags. The sampled version of the autocorrelation function captures this information for each source signal:

$$\text{ACF}_x(\tau) = \sum_t x(t)x(t + \tau) . \quad (25)$$

The autocorrelation function represents a correlation of the signal, x , with a time-lagged version of itself at all time-lags, τ . Figure 4 plots the autocorrelation function for a periodic signal.

Due to the independence assumption, sources are not expected to have lagged cross-correlations. Therefore, lagged covariance matrices for the source signals are diagonal and those computed on the mixtures are not, providing information for separation via joint diagonalization. These lagged covariance matrices are defined as the following:

$$\mathbf{R}_z^{lag}(\tau) = E\{\mathbf{z}(t + \tau)\mathbf{z}(t)^H\} , \quad (26)$$

where lag labels this as a lagged covariance matrix and τ is the time lag. The sampled version is:

$$\hat{\mathbf{R}}_z^{lag}(\tau) = \sum_{t=1}^{n-\tau} \mathbf{z}(t + \tau)\mathbf{z}(t)^H , \quad (27)$$

where n is the length of the signal. Because we are operating on the whitened sources, $\mathbf{R}_z^{lag}(0)$ is the identity matrix, \mathbf{R}_z and should not be included in the set. Another issue is

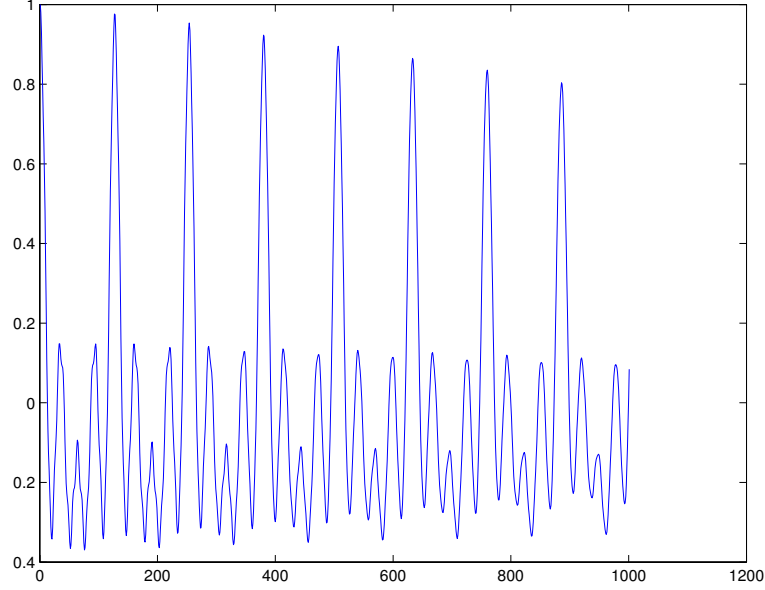


Figure 4: Autocorrelation function for a periodic signal

that different sources must have different autocorrelation functions. Otherwise, the $\mathbf{R}_{\mathbf{z}}^{lag}(\tau)$ will be scalar multiples of each other and therefore contain no distinguishing information. Diagonalizing a set of lagged covariance matrices with $\tau > 0$ identifies the unitary matrix \mathbf{U} and thereby separates independent autocorrelated sources.

3.2.3 Time-varying Energy

The previous algorithms operate on stationary sources. That is, the properties of the signal do not change over time. If a signal does change over time, this temporal structure informs separation. The first form of time-varying structure we consider is time-varying energy. Figure 5 shows the energy profile for a series of piano notes. Each note has a sharp attack followed by a smooth decay and release.

Matsuoka et al. [83] propose a neural network that attempts to decorrelate the mixture signal at every point in time. Alternatively, local correlation matrices computed for a neighborhood around time t capture this non-stationary variance [58]:

$$\mathbf{R}_{\mathbf{z}}^{loc}(t) = E_t\{\mathbf{z}(t)\mathbf{z}(t)^H\} , \quad (28)$$

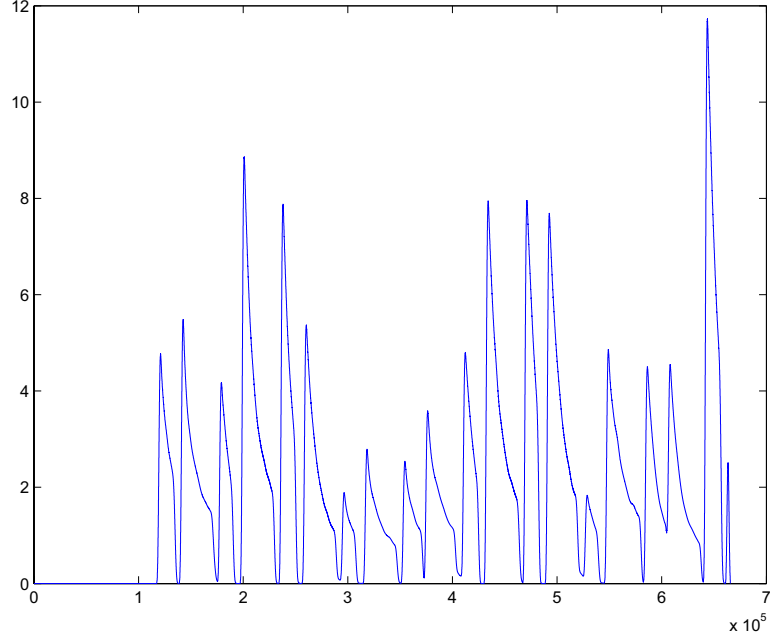


Figure 5: Time-varying energy for series of piano notes

where loc labels this as a local correlation matrix and E_t is the expectation within a local time interval around t . By assuming that the variance of each source varies slowly, the local expectation is computed as a weighted mean of nearby correlation matrices:

$$\hat{\mathbf{R}}_{\mathbf{z}}^{loc}(t) = \sum_{\tau} h(\tau) \mathbf{z}(t + \tau) \mathbf{z}^H(t + \tau), \quad (29)$$

where h is a windowing function with unit sum. The variance of each source must fluctuate differently, otherwise each $\mathbf{R}_{\mathbf{x}}^{loc}(t)$ will be a scalar multiple of the mixture covariance matrix, $\mathbf{R}_{\mathbf{x}}$. This leads to $\mathbf{R}_{\mathbf{z}}^{loc}(t)$ that are already diagonal and therefore provide no additional information. Provided that the independent sources have non-stationary variance and that they fluctuate differently, diagonalizing the set of $\hat{\mathbf{R}}_{\mathbf{z}}^{loc}(t)$ identifies \mathbf{U} and thereby separates them.

3.2.4 Time-Frequency Sparseness

Another form of non-stationarity occurs in the time-frequency domain (TFD). When sources change over time they often exhibit different frequency spectra. By converting the problem of source separation to the time-frequency domain, these changes can be isolated and leveraged for source separation. Time-frequency distributions capture this structure by representing a signal at time-frequency points. The spectrogram is often used to estimate the time-frequency energy of a single signal. However, other distributions enable the estimation of shared energy between signals, *e.g.*, the pseudo Wigner distribution [29]:

$$R_{z_1 z_2}^{tf}(t, f) = \int h(\tau) z_1(t + \frac{\tau}{2}) z_2^*(t - \frac{\tau}{2}) e^{-j2\pi f \tau} d\tau, \quad (30)$$

where tf labels this as a time-frequency correlation matrix. Written in sampled matrix form this becomes:

$$\hat{\mathbf{R}}_{\mathbf{z}}^{tf}(t, f) = \sum_{\tau} h(\tau) \mathbf{z}(t + \tau) \mathbf{z}^H(t - \tau) e^{-j2\pi f \tau}, \quad (31)$$

where $[\hat{\mathbf{R}}_{\mathbf{z}}^{tf}(t, f)]_{ij} \approx R_{z_i z_j}^{tf}(t, f)$. Figure 6 shows the time-frequency representation for a series of overlapping organ notes. Each note has a fundamental frequency and a number of harmonic frequency at integer multiples of the fundamental.

The relationship between the source and whitened time-frequency correlation matrices is preserved so that $\hat{\mathbf{R}}_{\mathbf{z}}^{tf}(t, f) = \mathbf{U} \hat{\mathbf{R}}_{\mathbf{s}}^{tf}(t, f) \mathbf{U}^H$. However, a crucial difference is that now the source correlation matrices may contain non-zero entries off the main diagonal. This is a result of the multiplication of the two signals in the time domain that results in a convolution of each source's spectra. In fact, $\mathbf{R}_{\mathbf{z}}^{tf}$ may contain non-zero entries off the main diagonal even if the diagonal is zero. Therefore, it is important to distinguish between correlation matrices that receive their energy from cross-terms and those that receive it from auto-terms.

When two or more sources have true energy concentrations at the same time-frequency point, it is very likely that there are large cross-terms [40]. Therefore, the surest way

where λ_i are the eigenvalues of $\mathbf{R}_z(\dots)$. Because eigenvalues are invariant under a unitary transformation, the rank-oneness of $\mathbf{R}_z(\dots)$ is the same as $\mathbf{R}_s(\dots)$. If a matrix has energy on the diagonal and is rank-one, it is likely to be an autoterm matrix. The joint diagonalization of a collection of autoterm matrices identifies the unitary matrix, \mathbf{U} , thereby separating signals that are non-stationary in the time-frequency domain.

3.2.5 Discussion

What all of these techniques have in common is that they diagonalize a set of matrices that capture some form of structure within the source signals. The key is that the source correlation matrices must be (nearly) diagonal with distinct eigenvalues and not proportional to the source covariance matrix, $\mathbf{R}_s^{lag}(0)$. This is always the case for the 4th-order correlation matrices, $\mathbf{R}_z^{4th}(i, j)$, computed on non-Gaussian independent sources in the JADE algorithm. There is exactly one matrix slice per source that contains a single non-zero element and it is on the diagonal [24]. In contrast, the lagged covariance matrices, $\mathbf{R}_z^{lag}(\tau)$, used by SOBI are likely to contain duplicate eigenvalues [17]. Therefore, a collection of matrices are diagonalized with the expectation that at least some of the matrices contain distinct eigenvalues and aid separation. When using local correlation matrices, $\mathbf{R}_z^{loc}(t)$, it is important that the ratio between local source variances change over time [83]. This ensures distinction from the source covariance matrix. Again, multiple local correlation matrices are diagonalized.

In time-frequency blind source separation, the diagonality of time-frequency spatial correlation matrices is brought into question. The only way to be certain that the matrices are diagonal is to choose time-frequency points with only one source contribution [55]. Because each autoterm matrix has only one non-zero diagonal entry, it reveals only one source. Therefore, multiple matrices are chosen in order to find an autoterm for each source. Next we consider the utility of repetitive structure and what can be expected from time-time correlation matrices.

3.3 Repetitive Structure

Many audio signals exhibit structure in the form of repetition. Music is the most obvious example because the structure is carefully constructed. Different combinations of instruments play at different times and the notes they play are repeated over the course of a song. Repetitive structure also exists in other audio signals such as speech and natural recordings. Words, syllables, and phonemes are repeated in a conversation. The sounds of keyboards, telephones, and printers permeate an office building. The similarity of the repetitions vary as do the patterns of repetition. For example, a bell tower chimes at regular intervals with each bell sounding the same every time it rings. A public address system replays the same announcement or variations of it at each stop on the subway. When the signal is a product of digital technology, the repetitions can be nearly identical as in a music synthesizer. Because each sound repeats in a different pattern, we expect to more easily separate it from a recording.

Music provides an excellent example of repetitive structure because the repetition is carefully constructed. Foote’s self-similarity matrix visualizes short- and long-term repetitions based on the comparison of very short audio frames [44]. The audio signal is segmented into short (*e.g.*, 50 millisecond) frames and each pair of frames is compared via a similarity metric. Figure 7 shows a self-similarity matrix for a rock song. Time runs from top-to-bottom and left-to-right. Regions of self-similarity appear as white squares along the diagonal. Repetitions appear as white rectangles off the diagonal. The diagonal is white because a frame is maximally similar to itself. Clearly there are two main parts to the song that repeat with high similarity: part A (0-15, 25-55, and 85-125 seconds) and part B (55-80 and 125-150 seconds). This type of repetitive structure informs tasks such as segmentation [47], summarization [33], and compression [61]. We propose using repetitive structure in a similar time-time representation for source separation. In addition, long-term structure has been used to identify different versions of the same song in a database [45, 7].

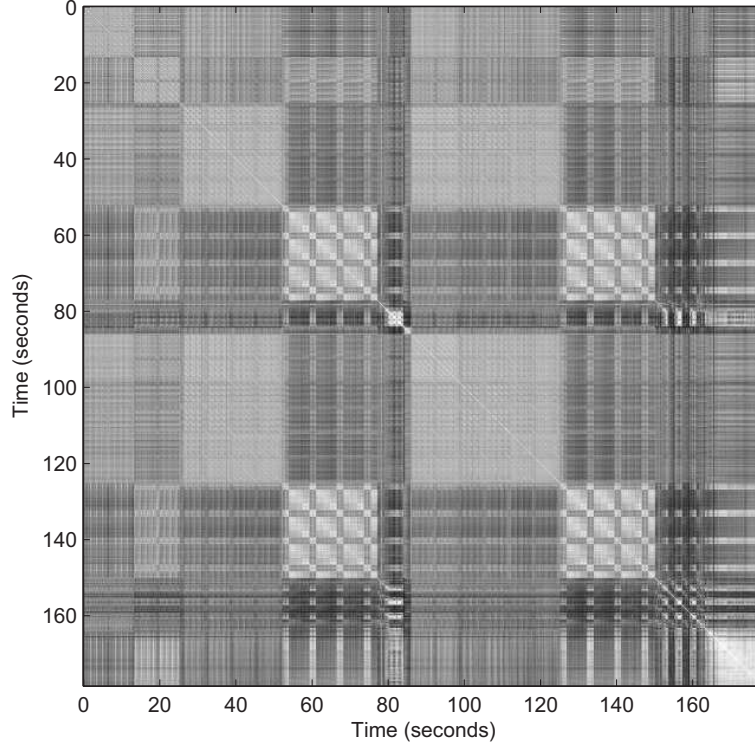


Figure 7: Self-similarity matrix for “March of the Pigs” by Nine Inch Nails

3.4 Time-Time Representations

Using the repetitive structure in source signals we propose a novel approach to blind signal separation. Using the same general approach as the above algorithms we present two ways to extend existing methods to capture repetitive structure in the time-time domain (TTD). First, repetitive structure can be thought of as a combination of local *and* lagged correlation matrices. That is, we construct correlation matrices between two *different* local time regions within the signal:

$$\vec{\mathbf{R}}_{\mathbf{z}}^{tt}(t_1, t_2) = E_{t_1 t_2} \{ \mathbf{z}(t_1) \mathbf{z}(t_2)^H \} , \quad (34)$$

where \vec{tt} labels this as a forward time-time correlation matrix. This is a lagged version of $\mathbf{R}_{\mathbf{z}}^{loc}(t)$ or a local version of $\mathbf{R}_{\mathbf{z}}^{lag}(\tau)$. In contrast to the other methods, time-time correlation matrices utilize lags that extend the entire length of the signal and are computed with a small neighborhood of samples. We estimate $\vec{\mathbf{R}}_{\mathbf{z}}^{tt}(t_1, t_2)$ using a windowing function h to

localize the computation of $\hat{\mathbf{R}}_{\mathbf{z}}^{\rightarrow}(t_1, t_2)$:

$$\hat{\mathbf{R}}_{\mathbf{z}}^{\rightarrow}(t_1, t_2) = \sum_{\tau} h(\tau) \mathbf{z}(t_1 + \tau) \mathbf{z}^H(t_2 + \tau) . \quad (35)$$

When $t_1 = t_2$, this is equivalent to the local correlation matrices in Equation 29. Otherwise, this representation captures correlations at various repetitions. The precision in the time domain depends on the size of the windowing function $h(\tau)$. As the window size increases, the time precision diminishes. That is, $\hat{\mathbf{R}}_{\mathbf{z}}^{\rightarrow}(t_1 + \delta t, t_2 + \delta t)$ changes slowly with respect to δt .

In addition, we represent repetitive structure within the framework of time-frequency distributions [88]. We manipulate the pseudo Wigner distribution (Equation 31) to operate on two points in time *without* frequency dependency (*i.e.*, setting $f = 0$):

$$\hat{\mathbf{R}}_{\mathbf{z}}^{\leftarrow}(t_1, t_2) = \sum_{\tau} h(\tau) \mathbf{z}(t_1 + \tau) \mathbf{z}^H(t_2 - \tau) , \quad (36)$$

where \leftarrow labels this as a time-time correlation matrix with the second windowed signal time-reversed. This approach benefits the precise analysis of signals in the time domain regardless of the window size. However, this precision comes at the cost of a slowly changing correlation matrix as t_1 and t_2 shift away from each other. That is, $\hat{\mathbf{R}}_{\mathbf{z}}^{\leftarrow}(t_1 - \delta t, t_2 + \delta t)$ varies slowly with respect to δt .

If we make the assumption that source signals have zero cross-correlations for every pair of time points, every time-time correlation matrix is diagonal. We can separate sources by simply diagonalizing a large set of time-time correlation matrices. However, if we make the more reasonable assumption that the sources are uncorrelated at every point in time (*i.e.*, the same as the nonstationary variance approach) not all matrices are diagonal. Instead, all matrices on the time diagonal, $\hat{\mathbf{R}}_{\mathbf{z}}^{\rightarrow}(t, t)$, are diagonal. Matrices computed at different points in time can have non-diagonal elements. This is because we are allowing a source at t_1 to be correlated to a different source at t_2 . This enables correlation matrices that have zeros on the diagonal and non-zeros off the diagonal. We use the trace of the matrix to remove matrices without sufficient diagonal energy. When there is energy on the diagonal

and the matrix is near rank-one, the matrix is likely to have only one diagonal element. Therefore, we choose autoterm time-time correlation matrices in the same way that we choose autoterm time-frequency points. That is, we apply a threshold the trace and rank-ness of the whitened time-time correlation matrices via Equation 32 and 33.

Because we are motivated by the self-similarity matrix in Figure 7 to incorporate repetitive structure for source separation, we might consider other time-time representations and their applicability to source separation. We chose the representations in Equation 35 and 36 because of their relation to existing source separation algorithms. Other forms of self-similarity are employed in Foote’s self-similarity matrices [44, 46, 47]. For example, we could use the magnitude spectrum, mel-frequency cepstral coefficients, or chroma [11] computed at each time-windowed signal instead of the time-domain signal. The key difficulty in using these and other similar signal features is that they are a non-linear function of the original signal, thereby destroying the linear relationship between sources and mixtures. For the sake of curiosity, we also implement the time-time algorithm using magnitude spectra and achieve some separation. However, for our test cases, the linear representations perform better. Perhaps other linear representations could be tailored to a particular set of sources.

3.5 Application to Blind Source Separation

In this section, we show the relevance of time-time representations for blind source separation by comparing it to the other algorithms described in this chapter on a variety of simulated and real source signals.

3.5.1 Simulated Sample-based Repetition

In its purest form, repetitive structure is evidenced by the exact repetition of a signal. In order to separate such a signal, $\hat{\mathbf{R}}_{\mathbf{z}}^{\rightarrow tt}(t_1, t_2)$ is clearly best suited. Consider two signals that

are drawn from a Gaussian distribution and one signal repeats:

$$s_1(t) = N(0, 1)$$

$$s_2(t) = \begin{cases} N(0, 1), & t \leq t_a \\ s_2(t - \tau_a), & t_a < t < t_b \\ N(0, 1), & t \geq t_b \end{cases},$$

where $\tau > t_b - t_a$. We construct two 1000 sample signals drawn from a Gaussian distribution. The second signal repeats samples 1-200 at samples 201-400, with $t_a = 200$ and $t_b = 400$. We use the following parameters for the algorithms:

$$\tau \in [1, 200]$$

$$t \in \{nK/2\}$$

$$t_1 \in \{nK/2\}$$

$$t_2 \in \{K/2 + i\},$$

where n is a positive integer, $K = 64$ is the frame size, and $i \in [0, 1000 - K]$. In addition, $h(\tau)$ is a Hamming window of size $K - 1$ centered at $\tau = 0$, and $t = t_1$ for the local correlation matrices, $\hat{\mathbf{R}}_z^{loc}(t)$. The second time point, t_2 must be evaluated at every sample in order to isolate the exact offset where the repetition occurs.

For time-time and time-frequency autoterms we choose the correlation matrices that exceed a rank-oneness of $\varepsilon_r = 0.8$ and are among the top 50 in terms of magnitude trace. We run 1000 trials drawing the real mixing matrix \mathbf{A} from a Gaussian distribution. To evaluate our approach with respect to how precisely the signal repeats, we add noise to each source and vary the signal-to-noise ratio (SNR). We measure the success of each algorithm based on the maximum interference-to-signal ratio (ISR) computed on the estimated $\hat{\mathbf{A}}$:

$$\text{ISR}(\hat{\mathbf{A}}, \mathbf{A}) = \max_p \sqrt{\frac{\sum_q |(\hat{\mathbf{A}}^\# \mathbf{A})_{pq}|^2}{\max_q |(\hat{\mathbf{A}}^\# \mathbf{A})_{pq}|^2}} - 1. \quad (37)$$

If $\hat{\mathbf{A}}$ is a good estimate of \mathbf{A} , $\hat{\mathbf{A}}^\# \mathbf{A}$ is close to a permutation matrix and the ISR is near zero. Table 1 summarizes our results. Our time-time representation, $\vec{\hat{\mathbf{R}}}_z^{tt}(t_1, t_2)$, outperforms the

Table 1: Average maximum ISR for each algorithm in decibels as a function of repetition similarity as SNR in decibels (*i.e.*, $10 \log_{10}(\text{ISR})$)

SNR	<i>4th</i>	<i>lag</i>	<i>loc</i>	<i>tf</i>	\vec{tt}	\overleftarrow{tt}
$+\infty$	-4.85	-8.94	-4.43	-4.62	-13.50	-4.80
20	-5.07	-8.71	-4.60	-4.78	-13.39	-5.02
15	-5.24	-8.50	-4.84	-4.64	-13.19	-4.99
10	-4.94	-8.27	-4.85	-4.66	-12.93	-4.82
5	-5.11	-7.07	-4.57	-4.98	-10.91	-4.69
0	-4.49	-7.65	-4.35	-4.80	-7.20	-4.76

others. This is to be expected because the sources are Gaussian with stationary variance and the same TFD. The only other method with marginal success is SOBI which is informed by the repetition in the correlation matrix at lag 200. We also see that the repetition need not be identical. An SNR of 5 dB provides enough similarity for time-time separation.

3.5.2 Simulated Spectrum-based Repetition

Signals often exhibit a less restrictive form of self-similarity. Although they do not repeat sample-for-sample, statistical properties of the signal repeat. For example, the frequency spectrum of a signal may repeat over time. Figure 7 shows regions of similarity where the spectral content is similar. To compare the various joint diagonalization algorithms, we construct source signals that have different repetition patterns with frequency-based similarity. To make different segments of the signal highly correlated to other parts, we draw each source from a Gaussian distribution and filter it with a conjugate pair filter. Each source has a different center frequency, f_i :

$$\begin{aligned}
 r_i(t) &= N(0, 1) \\
 z_i &= p e^{j2\pi f_i} \\
 a_i &= [1, -2\Re\{z_i\}, z_i z_i^*] \\
 s_i(t) &= r_i(t) - a_i(2)s_i(t-1) - a_i(3)s_i(t-2),
 \end{aligned} \tag{38}$$

where $p = 0.85$, $f_1 = 0.25 - \Delta f$, $f_2 = 0.25$, and $f_3 = 0.25 + \Delta f$. We create the repetition pattern by replacing sections of each signal with white Gaussian noise. Figure 8 shows the TTD computed on the sources with $\Delta f = 0.2$. Source 1 is filtered by f_1 for the first 60 frames. Source 2 is filtered by f_2 for the first and last 30 frames. Source 3 is filtered by f_3 for the last 60 frames. The repetition is characterized by the dark regions in the three matrices on the diagonal, $\vec{\mathbf{R}}_{s_i, s_i}^{tt}$. To aid the analysis of this figure, it can also be viewed as one large self-similarity matrix for one signal constructed as the concatenation of the three sources. The off-diagonal matrices represent cross-correlations between sources. When a source is being actively filtered, it is highly similar to itself (dark gray regions in matrices on the diagonal) and dissimilar to the other active sources (light gray and white regions in the off-diagonal matrices). The Gaussian noise is somewhat correlated to itself and everything else (medium gray regions in all matrices). The source correlation matrices that we whiten and then diagonalize are formed by taking the element (t_1, t_2) of each of these 9 matrices to construct $\vec{\mathbf{R}}_s^{tt}(t_1, t_2)$. The key observation is that any such matrix will likely have more energy on the diagonal than off-diagonal. Therefore, attempting to diagonalize $\vec{\mathbf{R}}_z^{tt}(t_1, t_2)$ will rotate the whitened mixtures closer to the original sources.

Figure 8 shows an example of sources that are well separated and provides a very good case for when time-time distribution source separation should work well. However, the sources in this example exhibit multiple types of structure. Because of the way the signals are filtered, each signal is a function of lagged versions of itself. The SOBI algorithm was designed especially for this type of signal and succeeds with only one time-lag. In addition, it is likely that the variance of each source fluctuates somewhat differently, in addition to the clearly separated time-frequency structure. Therefore, in its present state these sources should be easy to separate. To test how well the algorithms perform as the sources become more similar we evaluate the degree of separation for the different algorithms while varying Δf . We construct signals that are 6000 samples long. The first 4000, first and last 2000, and last 4000 samples are filtered using f_1 , f_2 , and f_3 , on sources 1, 2, and 3, respectively.

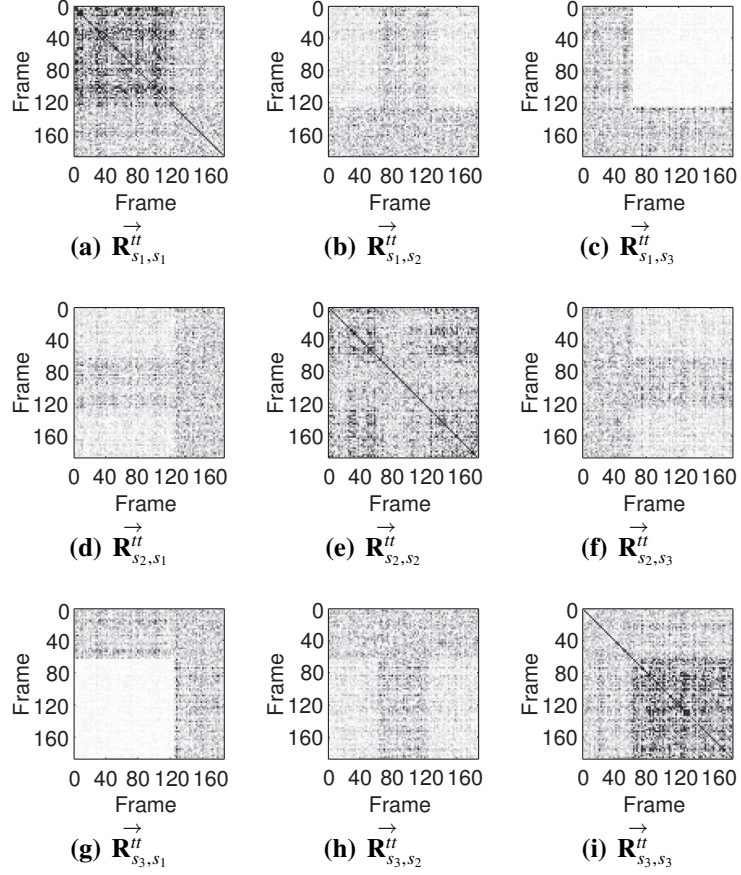


Figure 8: Time-time distribution matrices for each pair of sources

We run 1000 trials varying $\Delta f \in \{.2, .05, .01, .002, 0\}$. We use the same parameters as example 1, except $t = t_1 = t_2 \in \{nK/2\}$. We choose time-frequency and time-time matrices with above average trace and a rank-oneness above $\varepsilon_r \in \{0.1, 0.2, \dots, 0.9\}$. Table 2 summarizes the results using $\varepsilon_r = 0.7, 0.4$, and 0.4 for tf , \vec{tt} , and \overleftarrow{tt} , respectively.

Because the sources all have Gaussian distributions, there are no 4th-order correlations to aid separation. Otherwise, when the sources are well separated in frequency all of the algorithms perform well. Perhaps using local correlation matrices (*loc*) performs worse because there are no explicit changes in the variance. In fact, the signals are normalized to have unit variance. The other noticeable difference between the algorithms is that as the sources become more similar, time-time separation performs relatively better. This is due to the repetitive structure in the sources that is captured by time-time correlation matrices.

Table 2: Average maximum ISR for each algorithm in decibels as a function of difference in center frequency, Δf (i.e., $10 \log_{10}(\text{ISR})$)

Δf	<i>4th</i>	<i>lag</i>	<i>loc</i>	<i>tf</i>	\vec{tt}	\overleftarrow{tt}
0.200	-2.04	-18.75	-14.05	-17.27	-18.02	-17.95
0.050	-1.45	-15.57	-10.31	-12.21	-15.07	-15.07
0.010	-1.45	-7.66	-6.27	-9.75	-11.37	-11.43
0.002	-1.41	-4.04	-5.69	-9.46	-10.35	-10.46
0.000	-1.45	-3.78	-5.81	-9.37	-10.64	-10.64

3.5.3 Separation of Clarinet Recordings with Repetitive Structure

In the previous examples, we ran the experiments with multiple rank-oneness threshold values, ε_r , and chose the one that gave the best separation. In a blind separation task, this parameter must be chosen *a priori*. For the time-frequency algorithm, a number of methods for selecting auto-terms and even cross-terms have been proposed [18, 16, 40], involving the trace and possibly the rank of the correlation matrices. Because the trace and rank of a matrix is invariant under unitary transformation, the trace and rank of the whitened mixture correlation matrices is the same as that of the source correlation matrices. Perhaps the most convincing argument is that we can only be sure to find a diagonal source correlation matrix when there is only one source active at that time-frequency point [40]. In this case, the rank one source correlation matrix will have relatively high trace. To identify these points, the trace is thresholded against the average trace of all correlation matrices and the rank-oneness is measured as a ratio between the largest eigenvalue and the sum of the eigenvalues. A rank-oneness ratio near one indicates a nearly rank-one matrix.

When applied to time-time correlation matrices, a larger than average trace indicates that at least one source is active at both time-points, and the rank of the matrix indicates how many sources are active at both time-points. We consider a rank-one time-time correlation matrix with large enough trace to indicate the repetition of exactly one source. This type of structure reveals itself in sources that have different activation patterns or repetitive

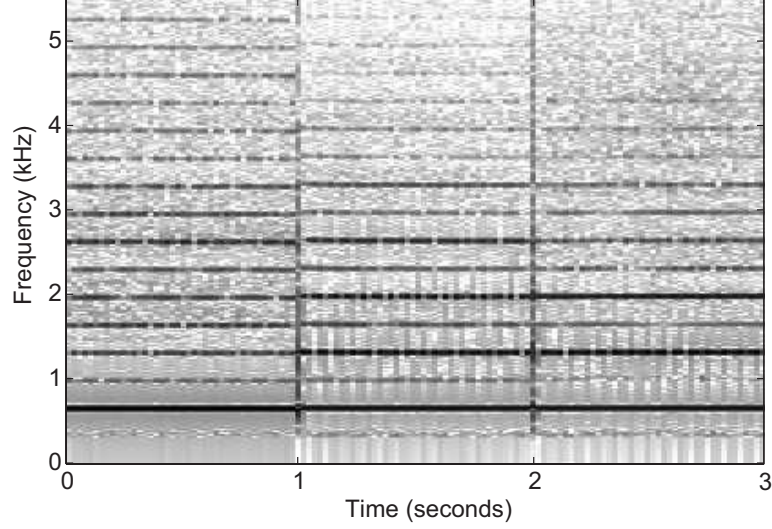


Figure 9: Time-frequency distribution for three clarinets

structure.

Using repetitive structure, we consider the separation of highly similar musical audio from the Iowa Musical Instrument Samples Database [48]. We extract one-second examples of the same note played on bass clarinet, Bb clarinet, and Eb clarinet. These instruments produce quite similar frequency spectra as shown by the log of their time-frequency distributions in Figure 9. The range from light to dark indicates mean energy to max energy. The horizontal lines are harmonics that overlap nearly perfectly. The self-similarity or time-time distribution of the bass clarinet ($\hat{\mathbf{R}}_{s_1 s_1}^{tt}$), Bb clarinet ($\hat{\mathbf{R}}_{s_2 s_2}^{tt}$), and Eb clarinet ($\hat{\mathbf{R}}_{s_3 s_3}^{tt}$) are shown in Figure 10(a), 10(e), and 10(i), respectively. The cross-correlations are contained in the off-diagonal matrices of Figure 10. The matrix formed by connecting the matrices in Figure 10 is the time-time distribution of a recording containing the three instruments played consecutively. If the sources were not correlated the off-diagonal matrices would be white (*i.e.*, no correlation). Instead, these sources are highly correlated at different points in time.

We use a threshold on the trace, ε_{Tr} , equal to the average trace of all correlation matrices, and a threshold on rank-oneness, ε_r , of 0.9 for all correlation matrices. In addition,

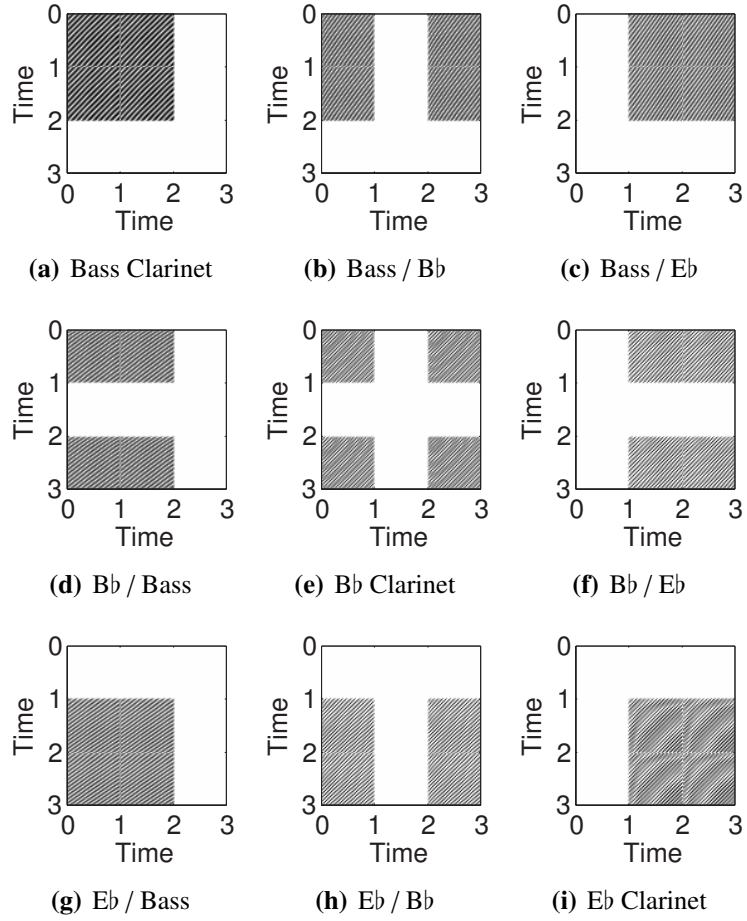


Figure 10: Time-time distribution matrices between and within clarinets

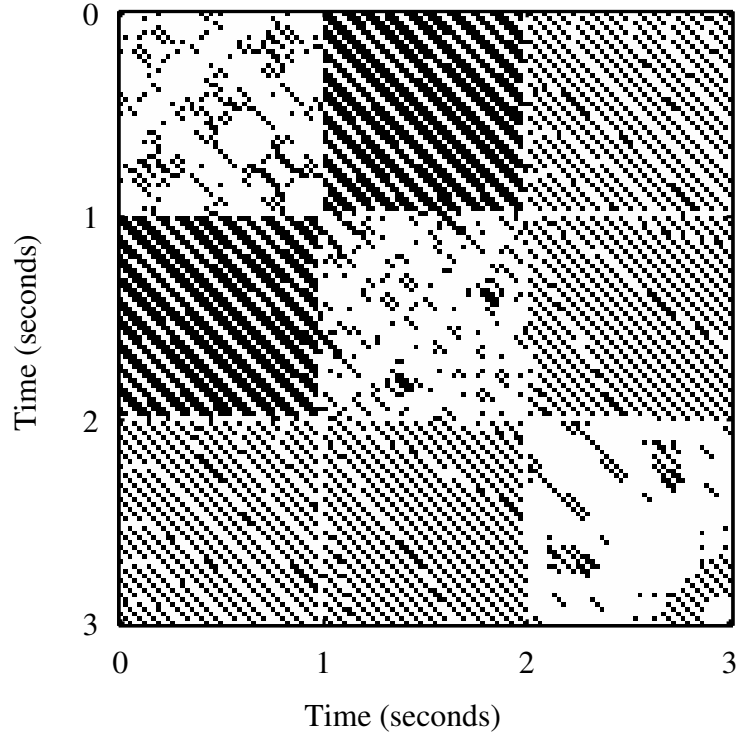


Figure 11: Time-time (\vec{tt}) autoterms (in black) for clarinets example

we use $K = 1024$ and a 50% overlap for frames in time-time, time-frequency, and local correlation approaches.

The \vec{tt} and \overleftarrow{tt} autoterms selected for this example are shown in Figure 11 and Figure 12, respectively. In spite of the similarity of the instruments, many time-time autoterms are identified. The alternating black and white lines for \vec{tt} (\overleftarrow{tt}) parallel (perpendicular) to the main diagonal indicate the fluctuating energy pattern in the clarinet sources. Each color change identifies when the energy crosses the energy threshold. This is also an example of how \overleftarrow{tt} and \vec{tt} differ. The \vec{tt} representation is more precise in the lag domain and less precise in the time domain, whereas the \overleftarrow{tt} representation is more precise in the time domain and less precise in the lag domain. The \vec{tt} and \overleftarrow{tt} algorithms accomplish an ISR of -12.54 dB and -12.40 dB, respectively.

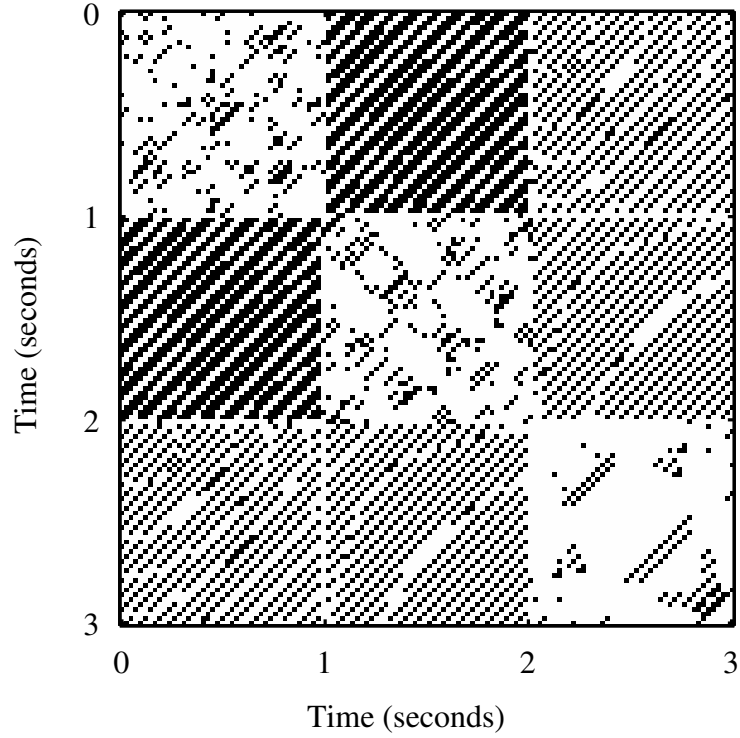


Figure 12: Time-time (\hat{tt}) autoterms (in black) for clarinets example

Figure 13 shows the time-frequency autoterms selected for the clarinet example. Because the time-frequency representations for each source overlap significantly, the time-frequency algorithm fails to find time-frequency autoterms. This results in an ISR of -3.74 dB.

The JADE algorithm operating on 4th-order correlation fails as well. The sources are non-Gaussian with kurtoses of 1.7, 2.0, and 2.7, respectively, and therefore contain 4th-order cumulants. However, sources 2 and 3 exhibit 4th-order cross-correlations. That is, $Q_s(2, 2, 2, 3)$, $Q_s(2, 3, 3, 3)$, and all permutations are non-zero. In short, the sources are *not* independent. Because JADE attempts to remove these higher-order correlations between sources it attains only an ISR of -6.17 dB.

Lagged autocovariance matrices capture the spectral shape of the sources. Figure 14 shows the structure in auto- and cross-correlation functions. The signals on the diagonal

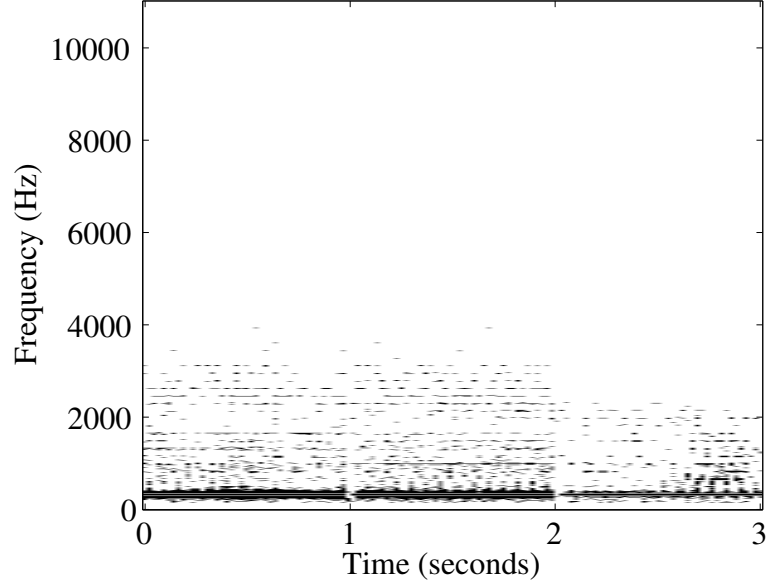


Figure 13: Time-frequency autoterms (in black) for clarinets

represent the autocorrelation function for each source and the off-diagonal signals represent the crosscorrelation between sources. Clearly, there is not much correlation between sources, so each matrix will be diagonal. However, each autocorrelation signal is very similar due to the spectral similarity of the sources. Each autocorrelation signal has the same fundamental frequency and slightly different shape. Figure 15 shows the overlap of the three autocorrelation functions for the first 125 time-lags. Because the sources are so similar spectrally, this approach does not perform quite as well, accomplishing an ISR of -11.31 dB using $\tau = 1000$.

Finally, the local correlation matrices perform the best because each source has a very distinct energy profile. Figure 16 illustrates this structure. The fact that each source is inactive at different times means that each 3×3 local source correlation matrix contains five zeros (one on the diagonal) and four non-zeros (two on the diagonal). Therefore, attempting to eliminate the off-diagonal entries is a good strategy on average. This approach achieves an ISR of -14.24 dB for this example.

Because the instruments are non-stationary with highly overlapping spectral shape and

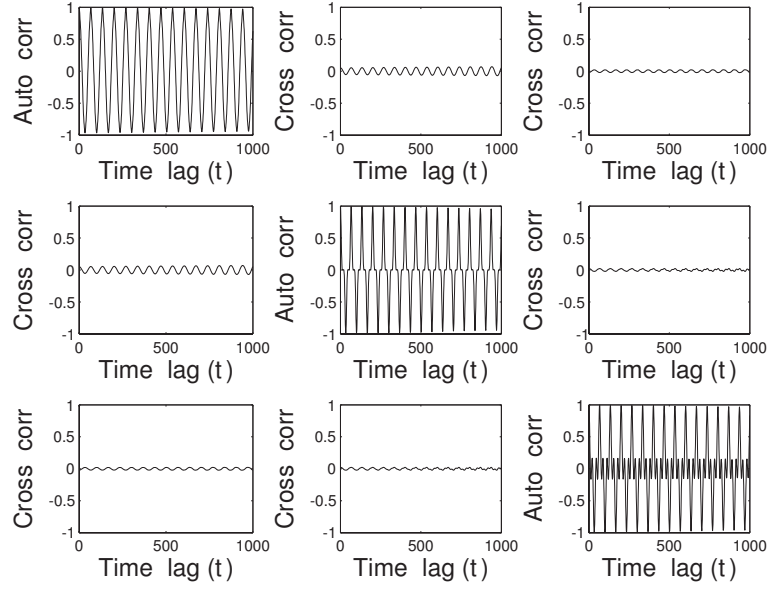


Figure 14: Lagged autocorrelation structure for clarinets

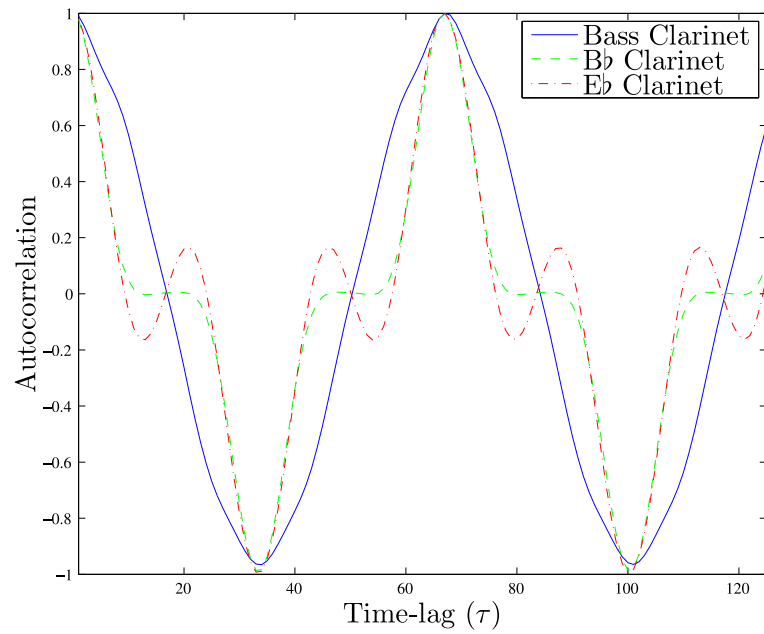


Figure 15: Similarity between each clarinet's autocorrelation

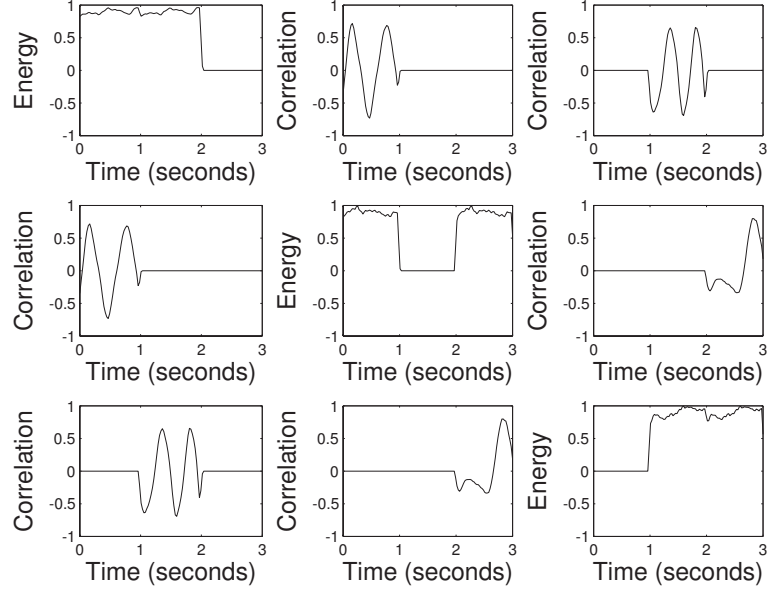


Figure 16: Local autocorrelation structure for clarinets

Table 3: ISR for each algorithm in decibels for clarinet example

<i>4th</i>	<i>lag</i>	<i>loc</i>	<i>tf</i>	\vec{tt}	\overleftarrow{tt}
-6.17	-11.31	-14.24	-3.74	-12.54	-12.40

contain 4th-order crosscorrelations, time-time separation outperforms time-frequency separation, JADE, and SOBI. Local autocorrelation matrices capture the activation pattern in the energy profile of the sources and improves on time-time separation by about two decibels ISR. Table 3 summarizes the results.

We began this analysis with the idea that single-source time-frequency and time-time points reveal the necessary structure for separation. However, we have learned that it might not be necessary to limit the time-frequency analysis to single-source time-frequency points. One motivating factor for choosing single-source points is that multiple sources at the same point might introduce crossterms. For this clarinet example it appears that these crossterms do not adversely affect separation. In fact, by considering a larger number of correlation matrices by lowering the rank-oneness threshold, we can improve the results.

Time-time separation reaches an ISR of -12.70 dB and -13.24 dB for \vec{tt} and \overleftarrow{tt} , respectively, when all time-time correlation matrices are used. Time-frequency separation outperforms all others with an ISR of -14.62 dB when all time-frequency correlation matrices are used. In general, the best ISR will not be attained by including all time-time or time-frequency points (as we saw in the last example). Therefore, blindly choosing the best rank-oneness threshold remains an unsolved problem.

3.6 Application to Source Detection

We have shown the relevance of repetitive structure for blind source separation of instantaneous mixtures when the number of source signals equals the number of mixture signals. It is straightforward under a white noise assumption to apply this to fewer sources than mixtures. For the case of more source signals than mixtures, it is enticing to think that repetitive structure might reveal the source locations or detect when a particular source is active.

In the time-frequency domain, rank-one time-frequency correlation matrices reveal when exactly one source is active. Because the source correlation matrix is diagonal and contains exactly one non-zero entry, it uniquely reveals one column of the whitened mixing matrix (*i.e.*, the pseudo-unitary matrix \mathbf{U}). The matrix \mathbf{U} is now $M \times N$ and pseudo-unitary in that $\mathbf{U}\mathbf{U}^H = \mathbf{I}_M$. As long as each source is the sole contributor to at least one time-frequency point, it is possible to detect all of the columns of \mathbf{U} , even if the number of sources, N , is greater than the number of mixtures, M [62, 85, 99].

It is clear that time-frequency autoterms reveal source positions. However, time-time autoterms are not as helpful. Because time-frequency correlation matrices are computed at a single point in time, the correlation matrices are symmetric. That is, $R_{z_i z_j}^{tf} = R_{z_j z_i}^{tf}$. However, time-time correlation matrices are not symmetric unless $t_1 = t_2$ because by switching the order of the arguments you are also switching the points in time, that is,

$$R_{z_i z_j}^{tt}(t_1, t_2) = R_{z_j z_i}^{tt}(t_2, t_1).$$

Even though the time-time correlation matrices are not symmetric, the time-time correlation matrices of the sources, $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}$, contain all the information required for source detection. If the ij -th element of $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}(t_1, t_2)$ (i.e., $\overset{\leftrightarrow}{R}_{s_i s_j}^{\text{tt}}(t_1, t_2)$) is nonzero, source i at t_1 is correlated to source j at t_2 . However, because $N > M$, we cannot simply invert \mathbf{U} to find $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}$. Instead, we must isolate time pairs that reveal parts of $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}$. For example, if source i is the only active source at t_1 and t_2 , the time-time correlation matrix is rank-one and reveals the whitened position of source i (a column of \mathbf{U}):

$$\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2) = \overset{\leftrightarrow}{R}_{s_i s_i}^{\text{tt}}(t_1, t_2) \mathbf{u}_i \mathbf{u}_i^H, \quad (39)$$

where \mathbf{u}_i is the i th column of \mathbf{U} . In this case, \mathbf{u}_i can be estimated up to a scale factor by the most significant eigenvector of $\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2)$, thus detecting source i at t_1 and t_2 . This is a special case because $\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2)$ happens to be rank-one and symmetric. In the general case, $\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2)$ is a linear combination of the product of all pairs of whitened mixing parameters:

$$\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2) = \sum_{ij} \overset{\leftrightarrow}{R}_{s_i s_j}^{\text{tt}}(t_1, t_2) \mathbf{u}_i \mathbf{u}_j^H. \quad (40)$$

Although reconstructing $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}$ from $\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}$ is generally not possible, we can hope to estimate one element of $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}(t_1, t_2)$, revealing one source active at t_1 and one source active at t_2 . If one element of $\overset{\leftrightarrow}{\mathbf{R}}_{\text{ss}}^{\text{tt}}(t_1, t_2)$ dominates the rest (i.e., $|\overset{\leftrightarrow}{R}_{s_q s_r}^{\text{tt}}(t_1, t_2)| \gg |\overset{\leftrightarrow}{R}_{s_i s_j}^{\text{tt}}(t_1, t_2)| \forall i \neq q$ or $j \neq r$), $\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2)$ is approximately a rank-one matrix:

$$\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2) \approx \overset{\leftrightarrow}{R}_{s_q s_r}^{\text{tt}}(t_1, t_2) \mathbf{u}_q \mathbf{u}_r^H. \quad (41)$$

We estimate this rank-one matrix using singular value decomposition:

$$\overset{\leftrightarrow}{\mathbf{R}}_{\text{zz}}^{\text{tt}}(t_1, t_2) \approx d(t_1, t_2) \mathbf{v}_1 \mathbf{v}_2^H, \quad (42)$$

where $d(t_1, t_2)$ is the largest magnitude singular value and \mathbf{v}_1 and \mathbf{v}_2 are the corresponding left and right singular vector, respectively ($\|\mathbf{v}_i\| = 1$). We estimate $d(t_1, t_2)$ and \mathbf{v}_1 for all t_1 and t_2 constructing the matrix on the left side of Figure 19. Then, we assign each $d(t_1, t_2)$,

to a “collection” function for one of the sources. We use the normalized inner product to determine the whitened position most similar to \mathbf{v}_1 :

$$\hat{k} = \underset{k}{\operatorname{argmax}} \frac{\mathbf{v}_1^H \mathbf{u}_k}{\|\mathbf{u}_k\|}. \quad (43)$$

If \mathbf{v}_1 is most similar to the whitened position of source \hat{k} , the value of the collection function $c_{\hat{k}}(t_1, t_2)$ is assigned to the value of $d(t_1, t_2)$. The other collection functions are set to zero for that time-time point. The three matrices on the right side of Figure 19 represent the collection functions.

The function $c_n(t_1, t_2)$ contains the evidence that source n is active at time t_1 given $\mathbf{R}_{\mathbf{z}\mathbf{z}}^{\leftrightarrow}(t_1, t_2)$. Each row contains all the activation information collected for that point in time. Therefore, we construct the activation function for source n , by integrating across the rows of c_n :





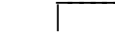
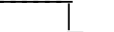

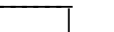
$$g_n(t) = \int c_n(t, \tau) d\tau. \quad (44)$$

Applying a threshold classifier to a smoothed version of this function could then provide the source activations.

We explore the following algorithm for source detection:

1. Compute the whitened time-time representation of the mixture signals from Equation 35 or Equation 36.
2. For each t_1 and t_2
 - (a) Compute the rank-one approximation according to Equation 42.
 - (b) Classify each left principal singular vector according to Equation 43 to find the source \hat{k} associated with t_1 .
 - (c) Assign $c_{\hat{k}}(t_1, t_2)$ to the largest singular value, $d(t_1, t_2)$.
3. Construct the activation function, g_n , by summing across the rows of matrix c_n .

Table 4: Activation sequence of sources

Time	0	1	2	3	4	5	6	7
Source 1	<i>on</i> <i>off</i>							
Source 2	<i>on</i> <i>off</i>							
Source 3	<i>on</i> <i>off</i>							

3.6.1 Detection of Spectrally Similar Sources

To demonstrate the relevance of our algorithm for source detection, we analyze a two channel mixture of three sources with overlapping frequency content. The sources are drawn from a Gaussian distribution with zero mean and unit variance and then filtered by a conjugate pair filter according to Equation 38 with $p = 0.85$, $f_1 = 0.20$, $f_2 = 0.25$, and $f_3 = 0.30$. Figure 17 shows the frequency content of each of the sources. The distributions show considerable overlap in frequency. We construct the repetitive structure by activating each source in a different pattern, shown in Table 4. Each activation from the same source is randomly generated using the same distribution and filter. Thus, the repetitions are not identical, only highly correlated.

We generate the mixtures, $\mathbf{x}(t)$, via Equation 15 using the following mixing matrix (Figure 18):

$$A = \begin{bmatrix} 0.4403 & 0.5499 & 0.9068 \\ -0.8978 & 0.8352 & 0.4215 \end{bmatrix}.$$

We compute the time-time representation of the whitened mixtures using Equation 36 and fill in the collection function, c_n , for each source. Figure 19 shows the collection function for source 1. Each row, t_1 , contains the evidence for source 1 being active at time index t_1 . The darker squares indicate that $\mathbf{R}_{zz}^{\leftarrow}(t_1, t_2)$ provides more evidence for source 1 activity when source 1 is present at both t_1 and t_2 . Figure 20 shows the activation function for each of the sources. As expected, when one source is active, only the correct source receives evidence of activation. For example, only source 3 is active from 3-4 seconds

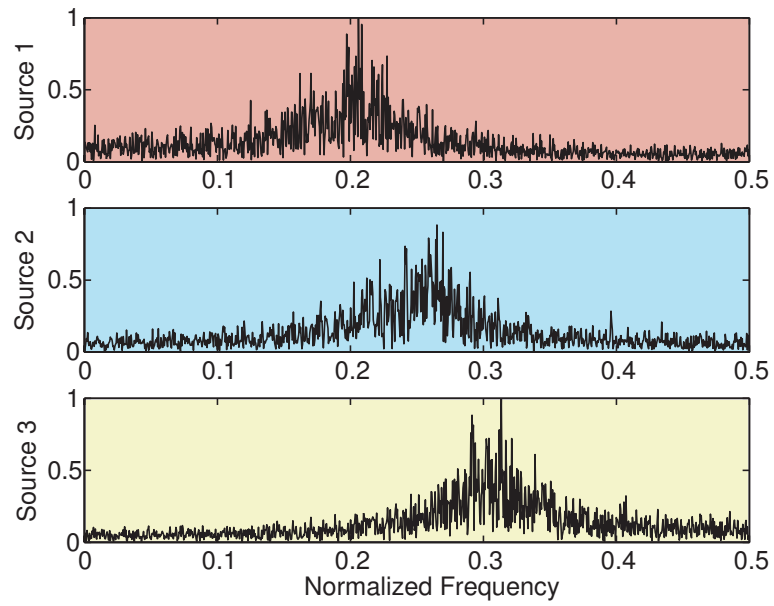


Figure 17: Normalized frequency of the sources

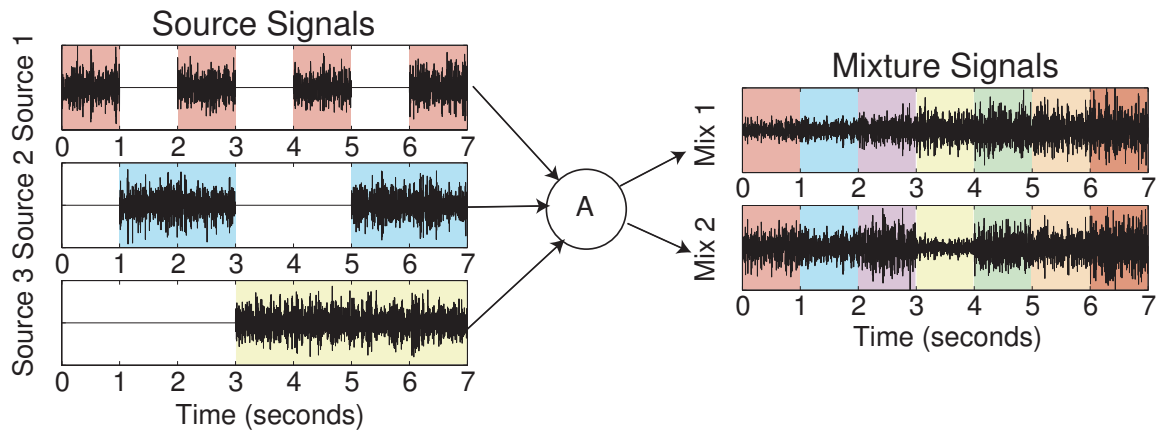


Figure 18: Generating the mixture of sources

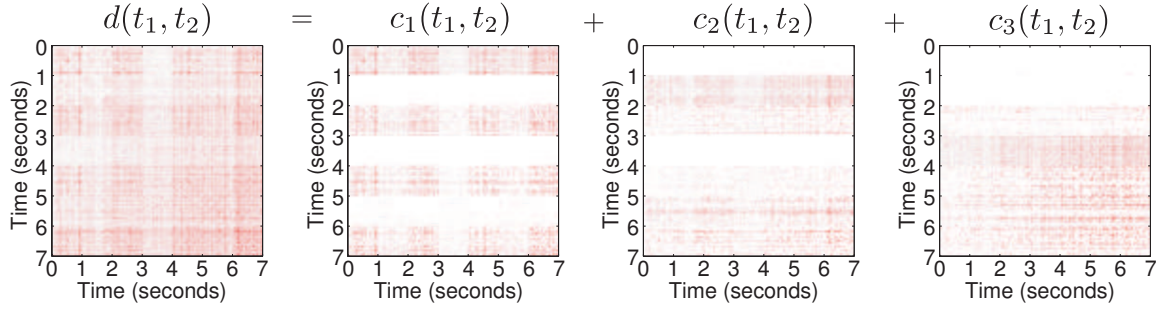


Figure 19: Computing the collection functions

in Figure 20. When two sources are active, they sometimes combine to approximate the remaining inactive source. For example some activation energy is shown for source 1 between 5-6 seconds, source 2 between 4-5 seconds, and source 3 between 2-3 seconds. When all three sources are active, the activation function is high for all three sources.

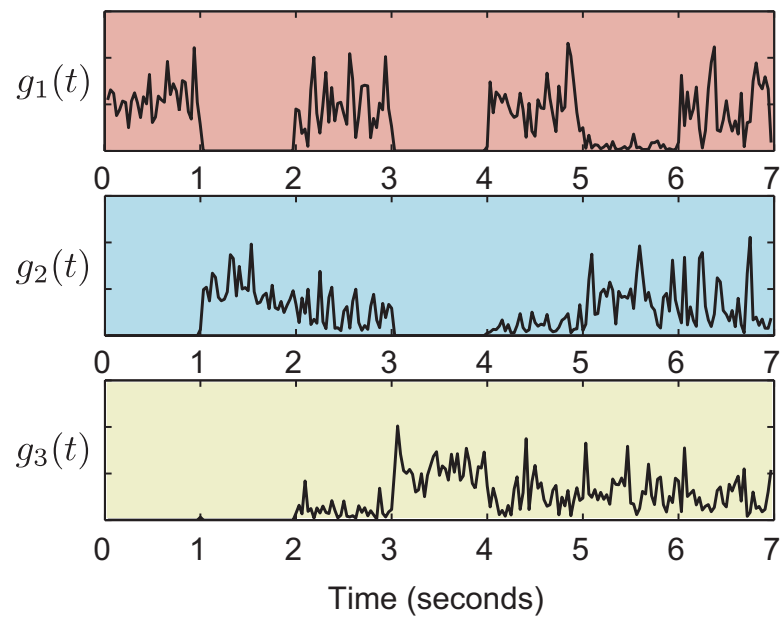


Figure 20: Activation function for each source

CHAPTER IV

SEPARATING MORE SOURCES THAN MIXTURES BY NON-NEGATIVE SPECTROGRAM FACTORIZATION*

Traditional approaches to source separation using independent component analysis including those we propose in Chapter 3 require that the number of sources does not exceed the number of mixture signals. This is rather restrictive considering that it may not be possible or affordable to record from as many microphones as there are instruments and because the majority of existing audio recordings are in mono (one channel) or stereo (two channels). For the case when only one mixture signal is available, this problem is particularly difficult.

One way to apply standard ICA algorithms to a single mixture signal is to transform it into a time-frequency representation such as the short-time Fourier transform (STFT). Because of phase-invariant aspects of human hearing and the sparseness of the resulting representation, the phase information in the STFT is removed yielding the magnitude, power, or log-magnitude spectrogram [25, 110]. By treating each frequency channel as a different input mixture signal, ICA extracts spectral components. The difference is that instead

*This chapter contains parts of the following copyrighted material:

PARRY, R. M. and ESSA, I., “Estimating the spatial position of spectral components in audio,” in *Independent Component Analysis and Blind Signal Separation*, vol. 3889 of *Lecture Notes in Computer Science (LNCS)*, (Charleston, SC), pp. 666–673, Springer, March 2006.
©Springer-Verlag Berlin Heidelberg 2006. With kind permission of Springer Science and Business Media.

PARRY, R. M. and ESSA, I., “Incorporating phase information for source separation via spectrogram factorization,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Honolulu, HI), April 2007.
©2007 IEEE. Reprinted with permission.

PARRY, R. M. and ESSA, I., “Phase-aware non-negative spectrogram factorization,” in *Independent Component Analysis and Signal Separation*, vol. 4666 of *Lecture Notes in Computer Science (LNCS)*, (London), pp. 536–543, Springer, September 2007.
©Springer-Verlag Berlin Heidelberg 2007. With kind permission of Springer Science and Business Media.

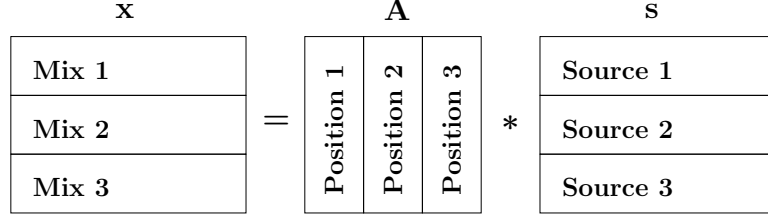


Figure 21: ICA in the time domain

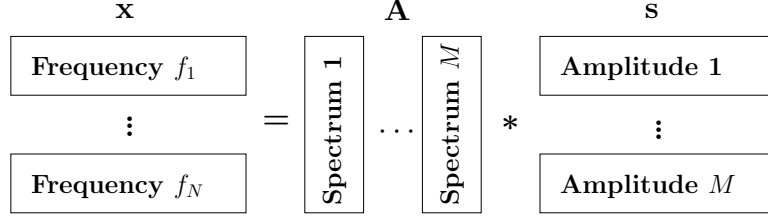


Figure 22: ICA in the frequency domain

of extracting sources with a static spatial position, the spectral components have a static spectral shape that has time-varying energy. These components of a signal represent one small part of a complex source and roughly correspond to a musical note or steady-state portions of speech. Because each component lacks the expressiveness of a complex source signal, multiple components are combined to form each source spectrogram. Finally, phase information is estimated or copied from the original mixture to construct an STFT for each source. The STFTs are inverted to estimate the source signals. Figure 21 and 22 depicts ICA in the time domain and frequency domain, respectively.

Applying ICA to the mixture spectrogram attempts to make the spectral components as independent as possible. However, an inherent mismatch exists between the ICA algorithm and the magnitude or power spectrogram data. Although the magnitude spectrogram is always positive, ICA is unconstrained and often produces negative frequency components or amplitudes. This is not a problem for reconstructing the sources because a negative magnitude simply rotates the original phase 180 degrees. However, it makes physical and visual interpretation of the components more difficult. As an alternative, non-negative matrix factorization (NMF) [72, 73] extracts spectral components and constrains the solution to be

non-negative [2, 112, 118, 119]. NMF has been applied to a variety of other problems in speech and audio (*e.g.*, [9, 97]). Here we use it to estimate the magnitude or power spectrum and amplitude envelope for each component while removing the independence criterion.

In this chapter, we address two aspects of this approach. First, we address the difficulty in determining which components belong to each source, and propose using an additional mixture signal to learn the spatial position of each component. Because different sources are at different spatial positions, we propose clustering the components in the spatial domain. Second, we examine the effect of removing the phase information in the mixture before analysis. The usual assumption is that the mixture magnitude or power spectrogram is the sum of the component spectrograms. However, this relationship additionally depends on the unknown phase of the sources. We show how this uncertainty can be incorporated by a cost function for NMF that improves the estimation of component spectrograms. We start with a brief review of fundamental technologies.

Although spectrogram factorization techniques have been extended to incorporate sparseness, convolution, and shifted spectra [2, 115, 43, 56, 57, 86, 106, 111, 116], we focus on improving the fundamental technique knowing that these extensions still apply with multiple mixtures and a different cost function.

4.1 Fundamental Technologies

4.1.1 Non-negative Spectrograms

We start with the standard instantaneous mixing model used in Chapter 3:

$$x_m(t) = \sum_{n=1}^N \mathbf{A}_{mn} s_n(t) , \quad (45)$$

where each of the N sources has a unique spatial position in the columns of the mixing matrix \mathbf{A} . In order to represent source components that capture note-like portions of the source signals, we model $x_m(t)$ as the weighted sum of R source components, $c_r(t)$:

$$x_m(t) = \sum_{r=1}^R \mathbf{A}_{mr} c_r(t) , \quad (46)$$

where $R \gg N$ source components. Now the mixing matrix \mathbf{A} contains multiple copies of each source position in its columns. Multiple instances of the same position indicate that multiple source components emit from the same spatial position and therefore the same source signal in Equation 45. We convert these signals into a time-frequency representation using the short-time Fourier transform (STFT):

$$\mathcal{F}_x(k, t) = \int x(\tau)h(\tau - t)e^{-j2\pi k\tau}d\tau, \quad (47)$$

where h is a short time window that localizes the Fourier transform. This preserves the linear relationship in Equation 46:

$$\mathcal{F}_{x_m}(k, t) = \sum_{r=1}^R \mathbf{A}_{mr} \mathcal{F}_{c_r}(k, t). \quad (48)$$

Because of phase-invariant aspects of human hearing and the sparseness of the resulting representation, the phase information in the STFT is removed yielding the magnitude or power spectrogram matrix [110]. The $K \times T$ magnitude spectrogram is the absolute value of the complex-valued STFT:

$$(\mathbf{X}_m)_{kt} = |\mathcal{F}_{x_m}(k, t)| \quad (\mathbf{C}_r)_{kt} = |\mathcal{F}_{c_r}(k, t)|. \quad (49)$$

The original STFT contains additional phase information which is not typically utilized:

$$\mathcal{F}_{x_m}(k, t) = (\mathbf{X}_m)_{kt}(\cos \Theta_{kt} + i \sin \Theta_{kt}) = \sum_{r=1}^R (\mathbf{C}_r)_{kt}(\cos (\Theta_r)_{kt} + i \sin (\Theta_r)_{kt}). \quad (50)$$

4.1.2 Non-negative Spectrogram Factorization

For a single mixture signal, non-negative spectrogram factorization (NSF) techniques including ICA and NMF decompose a single $K \times T$ mixture non-negative spectrogram, \mathbf{X} , into the product of a $K \times R$ matrix, \mathbf{B} , and an $R \times T$ matrix, \mathbf{H} :

$$\mathbf{X} \simeq \mathbf{B}\mathbf{H}, \quad (51)$$

where K is the number of frequency bins, T is the number of time samples, and R is the number of components. This factorization constrains each source component to have a

rank-one spectrogram. For now, because we are dealing with only one mixture signal, we omit the channel index m used in Equation 49. The r th (rank-one) component spectrogram is the product of the r th column of \mathbf{B} and the r th row of \mathbf{H} :

$$(\mathbf{C}_r)_{kt} = \mathbf{B}_{kr} \mathbf{H}_{rt} . \quad (52)$$

The columns of \mathbf{B} contain the spectral shapes and the rows of \mathbf{H} contain the amplitude envelopes for the components. The different NSF algorithms vary in how they estimate \mathbf{B} and \mathbf{H} given only \mathbf{X} .

4.1.3 NMF-based Non-negative Spectrogram Factorization

Non-negative matrix factorization (NMF) estimates \mathbf{B} and \mathbf{H} by minimizing a distance function between the single mixture spectrogram, \mathbf{X} , and \mathbf{BH} . The two common distance functions are the squared Euclidean distance:

$$\|\mathbf{X} - \mathbf{BH}\|^2 = \sum_{kt} (\mathbf{X}_{kt} - (\mathbf{BH})_{kt})^2 \quad (53)$$

and a generalized version of the Kullback-Leibler divergence:

$$D(\mathbf{X} \parallel \mathbf{BH}) = \sum_{kt} \left(\mathbf{X}_{kt} \log \frac{\mathbf{X}_{kt}}{(\mathbf{BH})_{kt}} - \mathbf{X}_{kt} + (\mathbf{BH})_{kt} \right) . \quad (54)$$

A gradient descent algorithm starts with a random initialization of \mathbf{B} and \mathbf{H} and follows the negative gradient until a local minimum is found:

$$\mathbf{B}_{kr} \leftarrow \mathbf{B}_{kr} - \beta_{kr} \frac{\partial D}{\partial \mathbf{B}_{kr}} \quad (55)$$

$$\mathbf{H}_{rt} \leftarrow \mathbf{H}_{rt} - \eta_{rt} \frac{\partial D}{\partial \mathbf{H}_{rt}} , \quad (56)$$

where D is a distance function.

4.1.3.1 Single Channel Euclidean Updates

The gradient for the Euclidean distance is proportional to the following:

$$\frac{\partial}{\partial \mathbf{B}_{kr}} \|\mathbf{X} - \mathbf{BH}\|^2 \propto (\mathbf{BHH}^T)_{kr} - (\mathbf{XH}^T)_{kr} \quad (57)$$

$$\frac{\partial}{\partial \mathbf{H}_{rt}} \|\mathbf{X} - \mathbf{BH}\|^2 \propto (\mathbf{B}^T \mathbf{BH})_{rt} - (\mathbf{B}^T \mathbf{X})_{rt} . \quad (58)$$

Choosing the following learning rates:

$$\beta_{kr} = \mathbf{B}_{kr} / (\mathbf{B} \mathbf{H} \mathbf{H}^T)_{kr} \quad (59)$$

$$\eta_{rt} = \mathbf{H}_{rt} / (\mathbf{B}^T \mathbf{B} \mathbf{H})_{rt} , \quad (60)$$

leads to the following multiplicative updates [73]:

$$\mathbf{B}_{kr} \leftarrow \mathbf{B}_{kr} \frac{(\mathbf{X} \mathbf{H}^T)_{kr}}{(\mathbf{B} \mathbf{H} \mathbf{H}^T)_{kr}} \quad (61)$$

$$\mathbf{H}_{rt} \leftarrow \mathbf{H}_{rt} \frac{(\mathbf{B}^T \mathbf{X})_{rt}}{(\mathbf{B}^T \mathbf{B} \mathbf{H})_{rt}} . \quad (62)$$

4.1.3.2 Single Channel KL-Divergence Updates

The gradient for the generalized Kullback-Leibler divergence is the following:

$$\frac{\partial}{\partial \mathbf{B}_{kr}} = \sum_t \mathbf{H}_{rt} - \sum_t \mathbf{H}_{rt} \frac{\mathbf{X}_{kt}}{(\mathbf{B} \mathbf{H})_{kt}} \quad (63)$$

$$\frac{\partial}{\partial \mathbf{H}_{rt}} = \sum_k \mathbf{B}_{kr} - \sum_k \mathbf{B}_{kr} \frac{\mathbf{X}_{kt}}{(\mathbf{B} \mathbf{H})_{kt}} . \quad (64)$$

Choosing the following learning rates:

$$\beta_{kr} = \mathbf{B}_{kr} / \sum_t \mathbf{H}_{rt} \quad (65)$$

$$\eta_{rt} = \mathbf{H}_{rt} / \sum_k \mathbf{B}_{kr} , \quad (66)$$

provides the following multiplicative updates [73]:

$$\mathbf{B}_{kr} \leftarrow \mathbf{B}_{kr} \frac{\sum_t \mathbf{H}_{rt} \mathbf{X}_{kt} / (\mathbf{B} \mathbf{H})_{kt}}{\sum_t \mathbf{H}_{rt}} \quad (67)$$

$$\mathbf{H}_{rt} \leftarrow \mathbf{H}_{rt} \frac{\sum_k \mathbf{B}_{kr} \mathbf{X}_{kt} / (\mathbf{B} \mathbf{H})_{kt}}{\sum_k \mathbf{B}_{kr}} . \quad (68)$$

4.1.4 ICA-based Non-negative Spectrogram Factorization

Instead of estimating \mathbf{B} and \mathbf{H} directly, ICA-based approaches start with the original mixture, \mathbf{X} , and attempt to transform it into a set of statistically independent amplitude envelopes in \mathbf{H} . After removing the mean so that the rows of \mathbf{X} sum to zero, the first step is

often a whitening transform, \mathbf{D} . Whitening the mixtures creates uncorrelated signals with unit variance. Optionally, smaller magnitude principal components may be removed to perform a dimensionality reduction:

$$\mathbf{Z} = \mathbf{D}\mathbf{X} , \quad (69)$$

where \mathbf{Z} is the $P \times T$ whitened mixture spectrogram and \mathbf{D} is the $P \times K$ whitening matrix and $P \leq K$. When no whitening is applied, we use $\mathbf{D} = \mathbf{I}_K$.

ICA is performed using a classic algorithm such as Bell and Sejnowski's information maximization algorithm [13]. Instead of estimating the spectral mixing matrix, \mathbf{B} , we estimate the $R \times P$ spectral unmixing matrix \mathbf{W} that maximizes the independence of the amplitude envelopes in \mathbf{H} :

$$\mathbf{H} = \mathbf{W}\mathbf{Z} . \quad (70)$$

Bell and Sejnowski's algorithm estimates a square \mathbf{W} ($R = P$) by maximizing the entropy of a nonlinear function of the estimated signals [13]:

$$H(\mathbf{Y}) = H(\mathbf{Z}) + \ln |\det \mathbf{W}| + F(\mathbf{Y}) \quad (71)$$

$$F(\mathbf{Y}) = \frac{1}{T} \sum_{t=1}^T \sum_{r=1}^R \ln |1 - \mathbf{Y}_{rt}^2| , \quad (72)$$

where $H(\cdot)$ is the entropy and we use $\mathbf{Y} = \tanh(\mathbf{W}\mathbf{Z})$ as the nonlinear function. We use gradient ascent to find a local maximum in H using an additive update rule:

$$\mathbf{W} \leftarrow \mathbf{W} + \omega \frac{\partial H}{\partial \mathbf{W}} , \quad (73)$$

where ω is a small constant and the gradient of H with respect to \mathbf{W} is the following (See Appendix B for our derivation) [13]:

$$\frac{\partial H}{\partial \mathbf{W}} \propto \mathbf{W}^{-T} - \frac{2}{T} \mathbf{Y}\mathbf{Z}^T . \quad (74)$$

Because the number of frequency bins, K , will often exceed the number of desired spectral components, a dimensionality reduction must be performed. In order to use standard

ICA algorithms that estimate a square unmixing (or mixing) matrix including Bell and Sejnowski's, the dimensionality reduction must be performed during the whitening stage via principal component analysis. However, this might lead to a loss of important information, for example, in the case of a source signal with relatively low energy. This signal's subspace is likely to be defined by relatively small magnitude principal components and therefore will be lost during the dimensionality reduction. Instead, we propose using undercomplete independent component analysis [113, 28, 4] to perform a dimensionality reduction concurrent to the estimation of the source signals. This requires the estimation of a non-square unmixing matrix. Stone estimates the entropy in Equation 71 using a non-square unmixing matrix \mathbf{W} [113]:

$$H(\mathbf{Y}) \approx \frac{1}{2} \ln |\det \mathbf{W} \mathbf{R}_Z \mathbf{W}^T| + F(\mathbf{Y}), \quad (75)$$

where $\mathbf{R}_Z = \mathbf{Z} \mathbf{Z}^T / (T - 1)$ is the covariance of the rows of \mathbf{Z} . The gradient of this approximate entropy with respect to \mathbf{W} does not require a square unmixing matrix (See Appendix B for our derivation) [113]:

$$\frac{\partial H}{\partial \mathbf{W}} = (\mathbf{W} \mathbf{R}_Z \mathbf{W}^T)^{-1} (\mathbf{W} \mathbf{R}_Z) - \frac{2}{T} \mathbf{Y} \mathbf{Z}^T. \quad (76)$$

We find a local maximum of the estimated entropy using the gradient ascent in Equation 73.

4.2 Multichannel Extensions

4.2.1 Extending NMF-based NSF to Multiple Channels

When multiple mixture signals for a recording are available (*i.e.*, a multichannel recording), different instruments occupy different spatial positions in the mixture. FitzGerald et al. extended non-negative matrix factorization of a single mixture to non-negative tensor factorization of multiple mixtures using Kullback-Leibler divergence [42]. We present matrix factorizations for NMF- and ICA-based non-negative spectrogram factorization by concurrently learning the spatial positions of independent spectral components. Our underlying assumption is that instruments maintain their spatial position and spectral components

maintain their shape across channels. Therefore, a single component may be modeled as a single spectral shape, spatial position, and amplitude envelope.

To accommodate multiple mixtures we reintroduce the $M \times R$ spatial mixing matrix \mathbf{A} in Equation 46. Each column of \mathbf{A} contains the spatial position of the spectral component represented by the corresponding column in \mathbf{B} and row in \mathbf{H} . In order to apply a factorization on magnitude (or power) spectra from multiple recordings, \mathbf{X}_m ($1 \leq m \leq M$), we construct $\bar{\mathbf{X}} \approx \bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}$, where $\bar{\mathbf{B}}$ is the multichannel spectral mixing matrix and $\bar{\mathbf{A}}$ is the multichannel spatial mixing matrix. For $M = 2$,

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} \approx \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \mathbf{H}. \quad (77)$$

The diagonal matrix \mathbf{A}_m contains the m -th row of \mathbf{A} on the diagonal, whereas \mathbf{A}_{ij} is the element in the i th row and j th column of matrix \mathbf{A} . Figure 23 illustrates this factorization highlighting one component with $K = 5$, $M = 2$, $R = 3$, and $T = 7$. Each spectral shape in \mathbf{B} is modulated by an amplitude envelope in \mathbf{H} spread across the the M mixture channels by $\bar{\mathbf{A}}$. We use a gradient descent algorithm with an additive update for \mathbf{A} analogous to Equation 55 and 56:

$$\mathbf{A}_{mr} \leftarrow \mathbf{A}_{mr} - \alpha_{mr} \frac{\partial D}{\partial \mathbf{A}_{mr}}. \quad (78)$$

4.2.1.1 Multichannel Euclidean Updates

We minimize the squared Euclidean distance between $\bar{\mathbf{X}}$ and $\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}$:

$$\|\bar{\mathbf{X}} - \bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}\|^2 = \sum_{mkt} ((\mathbf{X}_m)_{kt} - (\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt})^2, \quad (79)$$

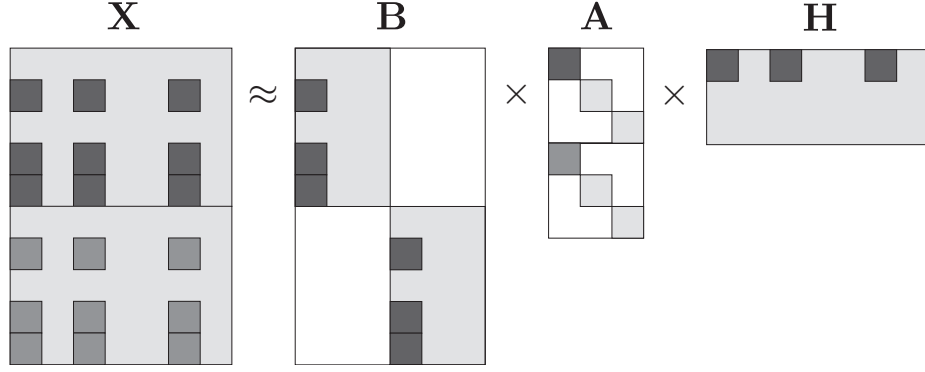


Figure 23: Multichannel factorization for NMF-based NSF

The gradient is proportional to the following:

$$\frac{\partial}{\partial \mathbf{B}_{kr}} \|\bar{\mathbf{X}} - \bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}\|^2 \propto \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} - \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt} \quad (80)$$

$$\frac{\partial}{\partial \mathbf{A}_{mr}} \|\bar{\mathbf{X}} - \bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}\|^2 \propto \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} - \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt} \quad (81)$$

$$\frac{\partial}{\partial \mathbf{H}_{rt}} \|\bar{\mathbf{X}} - \bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}\|^2 \propto \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} - \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{X}_m)_{kt} . \quad (82)$$

We choose the learning rates as follows:

$$\beta_{kr} = \mathbf{B}_{kr} / \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} \quad (83)$$

$$\alpha_{mr} = \mathbf{A}_{mr} / \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} \quad (84)$$

$$\eta_{rt} = \mathbf{H}_{rt} / \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt} , \quad (85)$$

and derive the following multiplicative updates:

$$\mathbf{B}_{kr} \leftarrow \mathbf{B}_{kr} \frac{\sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt}}{\sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt}} \quad (86)$$

$$\mathbf{A}_{mr} \leftarrow \mathbf{A}_{mr} \frac{\sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt}}{\sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt}} \quad (87)$$

$$\mathbf{H}_{rt} \leftarrow \mathbf{H}_{rt} \frac{\sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{X}_m)_{kt}}{\sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{B} \mathbf{A}_m \mathbf{H})_{kt}} . \quad (88)$$

4.2.1.2 Multichannel KL-Divergence Updates

We minimize the Kullback-Leibler divergence between $\bar{\mathbf{X}}$ and $\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}$:

$$D(\bar{\mathbf{X}}\|\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}) = \sum_{mkt} \left((\mathbf{X}_m)_{kt} \log \frac{(\mathbf{X}_m)_{kt}}{(\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}} - [\mathbf{X}_m]_{kt} + (\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt} \right). \quad (89)$$

The gradient is proportional to the following:

$$\frac{\partial}{\partial \mathbf{B}_{kr}} D(\bar{\mathbf{X}}\|\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}) = \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} - \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} \frac{(\mathbf{X}_m)_{kt}}{(\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}} \quad (90)$$

$$\frac{\partial}{\partial \mathbf{A}_{mr}} D(\bar{\mathbf{X}}\|\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}) = \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} - \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} \frac{(\mathbf{X}_m)_{kt}}{(\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}} \quad (91)$$

$$\frac{\partial}{\partial \mathbf{H}_{rt}} D(\bar{\mathbf{X}}\|\bar{\mathbf{B}}\bar{\mathbf{A}}\mathbf{H}) = \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} - \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} \frac{(\mathbf{X}_m)_{kt}}{(\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}}. \quad (92)$$

We choose the following learning rates:

$$\beta_{kr} = \mathbf{B}_{kr} / \sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} \quad (93)$$

$$\alpha_{mr} = \mathbf{A}_{mr} / \sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} \quad (94)$$

$$\eta_{rt} = \mathbf{H}_{rt} / \sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr}, \quad (95)$$

and derive the following multiplicative updates:

$$\mathbf{B}_{kr} \leftarrow \mathbf{B}_{kr} \frac{\sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt} / (\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}}{\sum_{mt} \mathbf{A}_{mr} \mathbf{H}_{rt}} \quad (96)$$

$$\mathbf{A}_{mr} \leftarrow \mathbf{A}_{mr} \frac{\sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt} (\mathbf{X}_m)_{kt} / (\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}}{\sum_{kt} \mathbf{B}_{kr} \mathbf{H}_{rt}} \quad (97)$$

$$\mathbf{H}_{rt} \leftarrow \mathbf{H}_{rt} \frac{\sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr} (\mathbf{X}_m)_{kt} / (\mathbf{B}\mathbf{A}_m\mathbf{H})_{kt}}{\sum_{mk} \mathbf{B}_{kr} \mathbf{A}_{mr}}. \quad (98)$$

4.2.2 Extending ICA-based NSF to Multiple Channels

For multichannel ICA-based non-negative spectrogram factorization, we introduce an $M \times P$ matrix \mathbf{V} containing the spatial unmixing parameters for each component in its columns.

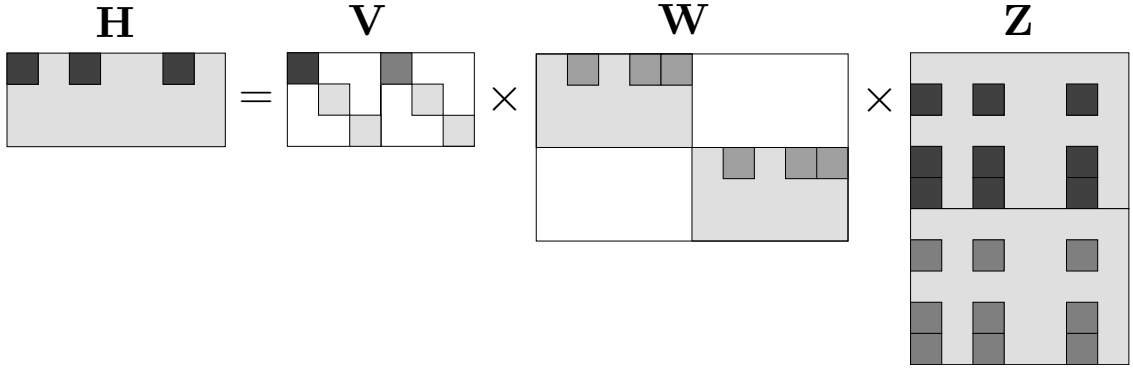


Figure 24: Multichannel factorization for ICA-based NSF

We factorize the unmixing system as $\mathbf{H} = \tilde{\mathbf{V}}\tilde{\mathbf{W}}\tilde{\mathbf{D}}\tilde{\mathbf{X}}$, where $\tilde{\mathbf{V}}$ is the multichannel spatial unmixing matrix, $\tilde{\mathbf{W}}$ is the multichannel spectral unmixing matrix, and $\tilde{\mathbf{D}}$ is the multichannel whitening matrix:

$$\mathbf{H} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}, \quad (99)$$

where \mathbf{V}_m is a diagonal matrix containing the m -th row of \mathbf{V} . Figure 24 shows the multichannel ICA-based factorization using $P = 5$, $M = 2$, $R = 3$, and $T = 7$. The $K : P$ dimensionality reduction via whitening has already been applied to the original mixtures to form \mathbf{Z} .

We incorporate this new factorization into the estimated entropy used for undercomplete ICA in Equation 75 and now estimate entropy as the following:

$$H(\mathbf{Y}) \approx \frac{1}{2} \ln |\det \tilde{\mathbf{V}}\tilde{\mathbf{W}}\mathbf{R}_{\tilde{\mathbf{Z}}}\tilde{\mathbf{W}}^T\tilde{\mathbf{V}}^T| + F(\mathbf{Y}), \quad (100)$$

where $\mathbf{Y} = \tanh(\tilde{\mathbf{V}}\tilde{\mathbf{W}}\tilde{\mathbf{Z}})$.

4.2.2.1 Multichannel ICA-based NSF Updates

We find a local maximum in the estimated entropy using gradient ascent with the additive update for \mathbf{W} (also in Equation 73) and \mathbf{V} :

$$\mathbf{W} \leftarrow \mathbf{W} + \omega \frac{\partial H}{\partial \mathbf{W}} \quad (101)$$

$$\mathbf{V} \leftarrow \mathbf{V} + \nu \frac{\partial H}{\partial \mathbf{V}} . \quad (102)$$

We derive the following gradient for $H(\mathbf{Y})$ with respect to \mathbf{W} and \mathbf{V} in Appendix B:

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = \sum_m \left(\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} - \frac{2}{T} \mathbf{V}_m \mathbf{Y} \mathbf{Z}_m^T \right)_{ij} \quad (103)$$

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{V}_{ij}} = \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T - \frac{2}{T} \mathbf{Y} \mathbf{Z}_i^T \mathbf{W}^T \right)_{jj} , \quad (104)$$

where

$$\begin{aligned} \mathbf{R}_{\hat{\mathbf{H}}} &= \bar{\mathbf{V}} \bar{\mathbf{W}} \mathbf{R}_{\bar{\mathbf{Z}}} \bar{\mathbf{W}}^T \bar{\mathbf{V}}^T \\ &= \sum_m \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T . \end{aligned} \quad (105)$$

4.2.3 Experiments

We show the relevance of our derivation to the estimation of spatial positions in addition to estimating spectral shapes and amplitude envelopes for non-negative spectrogram factorization.

4.2.3.1 Piano and Drum Mixture

We demonstrate our multichannel extensions to NMF- and ICA-based non-negative spectrogram factorization on mixtures of drum and piano music sampled at 11025 Hz. We artificially mix the tracks in the time domain via Equation 45. Then, we extract magnitude spectra from the short-time Fourier transform of the mixture signals using a Hanning window of 512 samples with 50% overlap and a fast Fourier transform of 1024 samples.

We generate the mixture signals using the first 20 seconds of the instrument tracks, panning the piano to the left and drum to the right with the mixing matrix:

$$\mathbf{A} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}. \quad (106)$$

The first column of \mathbf{A} distributes most of the piano to the first mixture channel (left). The second column of \mathbf{A} applies the reverse distribution to the drums.

We chose drum and piano music because note spectra from both are well modeled as the sum of stationary spectral components and have visibly different magnitude spectra and amplitude envelopes. We validate our approach by showing the correspondence between these visual attributes and the estimated spatial positions.

For multichannel NMF-based NSF, we apply a gradient descent algorithm to the drum and piano mixture. To initialize \mathbf{B} and \mathbf{H} , we apply successive updates of the single-channel Euclidean multiplicative updates in Equations 61 and 62 on the average magnitude spectrogram of the mixtures. After convergence, we set the minimum value in \mathbf{B} and \mathbf{H} to a small factor to avoid clamping at zero with the multiplicative updates. Finally, we alternately apply the multichannel Euclidean multiplicative Equations 86, 87, and 88 to extract $R = 7$ components. Throughout the estimation, we maintain unit norm columns of \mathbf{B} and \mathbf{H} . For multichannel ICA-based NSF, we apply a block whitening matrix $\hat{\mathbf{D}}$ that provides the $K = 513$ to $P = 50$ dimensionality reduction before alternate updates using Equations 101 and 102 to extract $R = 7$ components.

The whitening transform preserves 99.99987% of the variance in the mixture magnitude spectrograms.

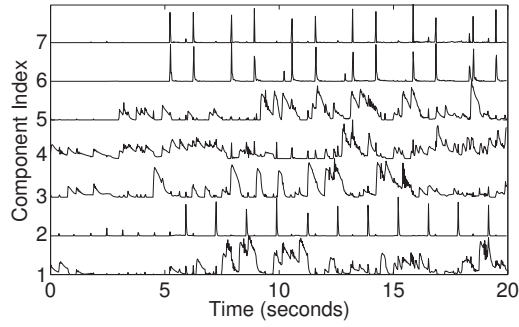
The left and right side of Figure 25 shows the extracted components using multichannel NMF and multichannel ISA, respectively. Figure 25(a) and 25(b) plots the time envelope of the components. The envelopes show that components 2, 6, and 7 from NMF and components 1 and 2 from ICA represent the short spiked attacks of the drums. The other components are from the piano. Because the NMF components contain only non-negative values,

they are generally easier to interpret than the ICA components. For example, the piano components in Figure 25(a) have sharp attacks and smooth decay illustrated by roughly right-triangular onsets. This detail is less prevalent in the ICA components especially at lower energy levels.

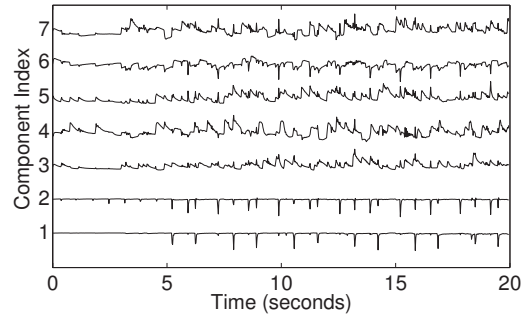
The component spectra in Figure 25(c) show the harmonic content of the piano and the noisy or low-frequency content of the drums. The larger peaks in the piano components occur at roughly linearly spaced frequencies indicating a harmonic relationship between them. This structure is more apparent in NMF components 3, 4, and 5. The noisy frequency content in component 2, and low-frequency concentration in components 6 and 7 are characteristic of the drums. This structure is difficult to see in the ICA components in Figure 25(d). Figure 25(e) and 25(f) show the component positions. These positions verify what we can see in the temporal envelopes and frequency content of the components. The drum components cluster on the left and the piano components cluster on the right.

4.2.3.2 *Estimating Component Positions with a Variety of Mixing Parameters*

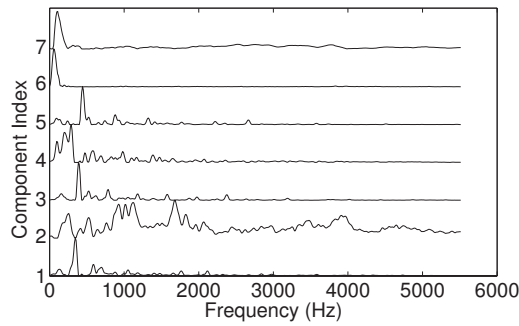
In the above example, we chose well-separated instrument positions. To test our algorithm's performance on a variety of mixing parameters, we apply 100 Monte-Carlo runs to extract seven components with uniformly distributed random mixing matrices. We estimate the utility of each of the extracted positions using the interference-to-signal ratio (ISR). We use the interference-to-signal ratio (ISR) of the spatial positions to encapsulate this information. For the two instrument case, an ISR of 1 indicates that a component is placed evenly between both instruments. An ISR of 0 is perfectly matched to its true position. Table 5 summarizes the distribution of ISRs for all 700 extracted components. More than half of all components are positioned within an ISR of 0.001, while only 3% appear closer to the wrong instrument position. For comparison, an ISR less than 0.001 using the mixing matrix in Equation 106 corresponds to a range of 0.18-0.22 for piano components and 0.78-0.82 for drum components.



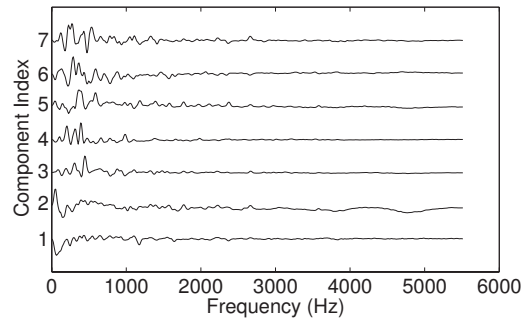
(a) NMF envelopes



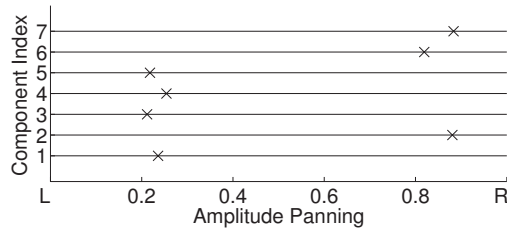
(b) ISA envelopes



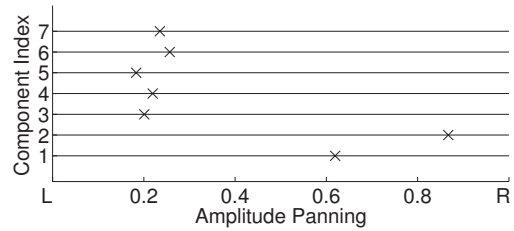
(c) NMF spectra



(d) ISA spectra



(e) NMF positions



(f) ISA positions

Figure 25: Components extracted from drums and piano using multichannel NMF- (*left*) and ICA-based NSF (*right*)

Table 5: Distribution of Component Position Error (ISR)

ISR	% of components
<0.001	52
<0.01	80
<0.1	89
<1	97
>1	3

We found three causes for ISRs greater than 1. First, when the random mixing matrix has nearly identical instrument positions (*i.e.*, an approximately rank one mixing matrix), it is unreasonable to expect well separated components. Second, sometimes two components learn the same spectral shape. This detection error affects the position of the components because moving one component to the right can be compensated by the other component moving to the left. Third, sometimes components learn parts of both instruments. For example, spikes in components 5 and 6 of Figure 25(a) can be seen in component 7. When large portions of both instruments are contained within one component, its position is somewhere in between the two source positions. Generally, the better a component represents exactly one instrument, the closer its position to that instrument.

4.2.3.3 *Three Pianos Playing Same Four Notes*

When applied to more difficult examples, multichannel ICA-based NSF was less predictable and visually less informative than multichannel NMF-based NSF. For example, sources that contain highly similar spectra are difficult for the ICA-based approach to handle. When applied to magnitude spectrograms, ICA generates linearly independent spectral shapes. Therefore, it is impossible for two components to represent the same spectra. In contrast, multichannel NMF only requires the non-negativity of source components.

We apply multichannel NMF to three pianos playing the same four notes in different orders. Piano 1, 2 and 3, are positioned to the left, center, and right in the stereo mixture,

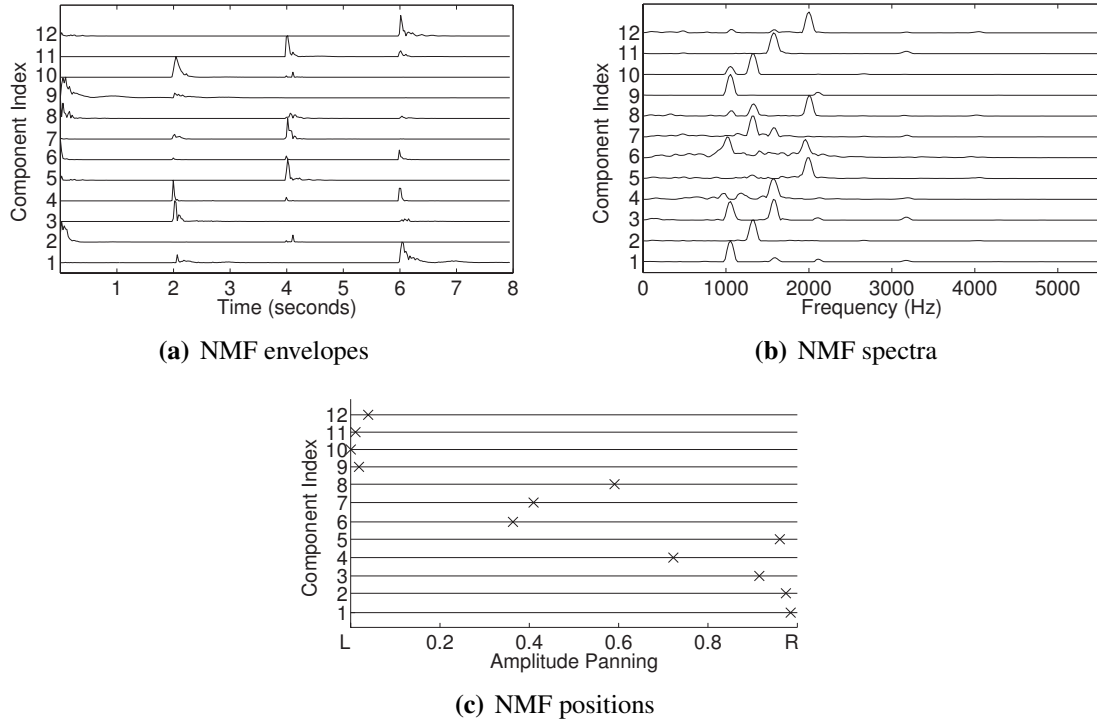


Figure 26: Extracted component envelopes, spectra, and positions for multichannel NMF-based NSF and three piano sources

respectively. Figure 26 shows $R = 12$ extracted components. Components 9–12 clearly represent piano 1 playing the notes in order from low to high. Each component is roughly one note represented by a temporal spike in Figure 26(a), one dominating frequency in Figure 26(b), and cluster together on the left side of Figure 26(c). In a similar way, components 1, 2, 3, and 5 represent piano 3, except component 3 contains two frequency peaks instead of one. The remaining components capture parts of piano 2. However, each contain multiple frequency concentrations and are generally less distinct. In spite of this, each source can be distinguished by its stereo position in Figure 26(c).

4.3 Incorporating Phase Information

As presented in the previous sections, NSF methods commonly assume that the mixture magnitude or power spectrogram is well approximated by the sum of source components.

ICA forces this relationship while maximizing the independence of the spectral components [25], whereas NMF minimizes a cost function between the mixture spectrogram and the sum of spectral components [73]. However, because of the nonlinearity of the absolute value function a mixture spectrogram is not the sum of the component spectrograms. That is, even though the components are mixed linearly in Equation 46:

$$x_m(t) = \sum_{r=1}^R \mathbf{A}_{mr} c_r(t) ,$$

and their STFTs are mixed linearly in Equation 48:

$$\mathcal{F}_{x_m}(k, t) = \sum_{r=1}^R \mathbf{A}_{mr} \mathcal{F}_{c_r}(k, t) , \quad (107)$$

discarding the phase to form the magnitude or power spectrogram removes the linearity of the relationship. The mixture non-negative spectrogram is *not* the sum of the component non-negative spectrograms:

$$\mathbf{X}_m \neq \sum_{r=1}^R \mathbf{A}_{mr} \mathbf{C}_r . \quad (108)$$

This is a problem even when there is only one mixture, \mathbf{X} , and R components:

$$\mathbf{X} \neq \sum_{r=1}^R \mathbf{C}_r , \quad (109)$$

where the scalar weight for each component is incorporated into its spectrogram, \mathbf{C}_r . Instead, the mixture spectrogram depends on the component spectrograms and their phases. For this single-channel case, we derive a cost function suitable for NSF by treating the phase as a uniform random variable and maximizing the likelihood of the mixture spectrogram.

4.3.1 Probabilistic Representation of the Non-negative Mixture Spectrogram

Both ICA- and NMF-based techniques implicitly assume that the mixture non-negative spectrogram, \mathbf{X} , is well approximated by the sum of the spectral components, \mathbf{C}_r . However,

by incorporating the phase of the components, Θ_r , we make this relationship precise:

$$\begin{aligned}
\mathbf{X}_{kt} &= |\mathcal{F}_x(k, t)| \\
&= |\mathbf{X}_{kt}(\cos \Theta_{kt} + i \sin \Theta_{kt})| \\
&= \left| \sum_{r=1}^R (\mathbf{C}_r)_{kt} (\cos (\Theta_r)_{kt} + i \sin (\Theta_r)_{kt}) \right| \\
&= \left| \sum_{r=1}^R (\mathbf{C}_r)_{kt} \cos (\Theta_r)_{kt} + i \sum_{r=1}^R (\mathbf{C}_r)_{kt} \sin (\Theta_r)_{kt} \right| \\
&= \sqrt{\left(\sum_{r=1}^R (\mathbf{C}_r)_{kt} \cos (\Theta_r)_{kt} \right)^2 + \left(\sum_{r=1}^R (\mathbf{C}_r)_{kt} \sin (\Theta_r)_{kt} \right)^2} \\
&= \sqrt{\sum_{qr} (\mathbf{C}_q)_{kt} (\mathbf{C}_r)_{kt} \cos (\Theta_q)_{kt} \cos (\Theta_r)_{kt} + \sum_{qr} (\mathbf{C}_q)_{kt} (\mathbf{C}_r)_{kt} \sin (\Theta_q)_{kt} \sin (\Theta_r)_{kt}} \\
&= \sqrt{\sum_{qr} (\mathbf{C}_q)_{kt} (\mathbf{C}_r)_{kt} (\cos (\Theta_q)_{kt} \cos (\Theta_r)_{kt} + \sin (\Theta_q)_{kt} \sin (\Theta_r)_{kt})} \\
&= \sqrt{\sum_{qr} (\mathbf{C}_q)_{kt} (\mathbf{C}_r)_{kt} \cos((\Theta_q)_{kt} - (\Theta_r)_{kt})} . \tag{110}
\end{aligned}$$

The mixture magnitude spectrogram does not equal the sum of component magnitude spectrograms unless at most *one* component is active at a time or all active components have the *same* phase.

In spite of the importance of phase information for determining the mixture magnitude spectrogram, the phase has not been utilized to estimate the component spectrograms in the aforementioned NSF techniques. Perhaps the simplest way to introduce information about the phase without knowing the specific values is to leverage its probability density function. If we plot a histogram of phase values for a music or speech source signal we find a uniform distribution between $-\pi$ and π . This represents the simplest information about phase we can utilize for component estimation. Without knowledge of the phase at any other time-frequency point, the phase at point (t, f) is equally likely to be anywhere in the range $-\pi$ to π . Because we know the probability density function of the phase, we use Equation 110 to derive the probability density function of the magnitude mixture spectrogram.

4.3.2 Two Components

For the case of two components, we simplify the notation so that $x = \mathbf{X}_{kt}$, $c_1 = (\mathbf{C}_1)_{kt}$, $c_2 = (\mathbf{C}_2)_{kt}$, and x is the function of a single random variable, $\theta = \Theta_1 - \Theta_2$:

$$x(\theta) = \sqrt{c_1^2 + c_2^2 + 2c_1c_2 \cos(\theta)} . \quad (111)$$

Because of the circularity of phase, the difference in two uniformly distributed random phases is also a uniformly distributed random variable, $\theta = U(-\pi, \pi)$. Because x is a function of θ , x is also a random variable. We derive the probability density function for x given c_1 and c_2 .

The phase difference, θ , is equally likely in the domain $-\pi$ to π . However, because the cosine function is unaffected by sign, we choose to map it to the non-positive domain, $\theta = U(-\pi, 0)$:

$$p_\theta(\theta) = \frac{1}{\pi} , \quad -\pi \leq \theta \leq 0 . \quad (112)$$

Because $x(\theta)$ is a monotonically increasing function on the domain $-\pi \leq \theta \leq 0$, the probability density function of x is the following [53]:

$$p_x(x) = p_\theta(\theta(x)) \left| \frac{d\theta(x)}{dx} \right| . \quad (113)$$

We solve for θ in terms of x :

$$\theta(x) = \cos^{-1} \left(\frac{x^2 - c_1^2 - c_2^2}{2c_1c_2} \right) , \quad (114)$$

and differentiate w.r.t. x :

$$\frac{d\theta(x)}{dx} = \frac{-x}{c_1c_2 \sqrt{1 - \left(\frac{x^2 - c_1^2 - c_2^2}{2c_1c_2} \right)^2}} . \quad (115)$$

The probability density function of θ in terms of x is the following:

$$p_\theta(\theta(x)) = \frac{1}{\pi} , \quad |c_1 - c_2| \leq x \leq c_1 + c_2 . \quad (116)$$

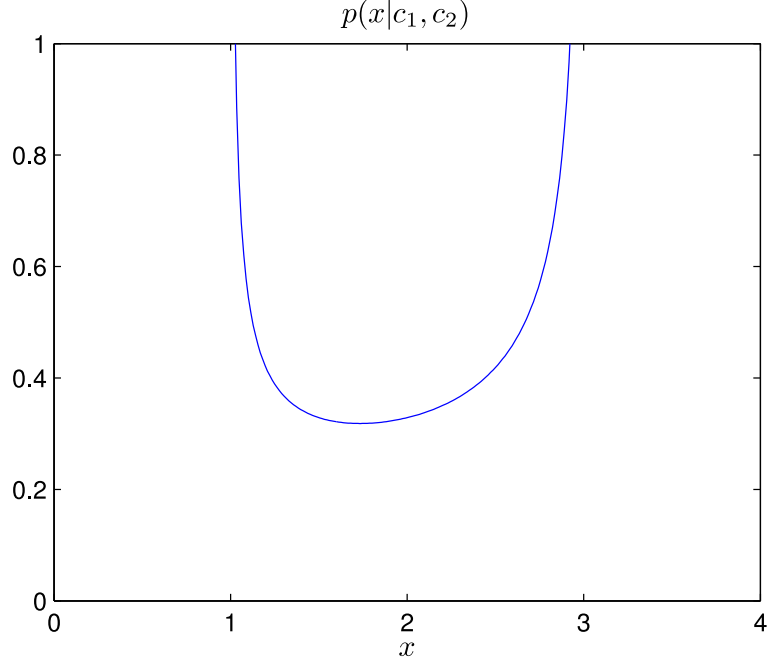


Figure 27: Likelihood function for x when $c_1 = 2$ and $c_2 = 1$

Substituting Equations 115 and 116 into Equation 113, dropping the subscript, and making the dependence on c_1 and c_2 explicit yields the likelihood of x :

$$\begin{aligned}
 p(x|c_1, c_2) &= \frac{x}{\pi c_1 c_2 \sqrt{1 - \left(\frac{x^2 - c_1^2 - c_2^2}{2c_1 c_2} \right)^2}}, \quad |c_1 - c_2| \leq x \leq c_1 + c_2 \\
 &= \frac{2x}{\pi \sqrt{-(x + c_1 + c_2)(x + c_1 - c_2)(x - c_1 + c_2)(x - c_1 - c_2)}}. \quad (117)
 \end{aligned}$$

The roots of the polynomial inside the square root are $x = \pm c_1 \pm c_2$. The function is defined in the domain $|c_1 - c_2| < x < c_1 + c_2$ and approaches infinity as x approaches $|c_1 - c_2|$ and $c_1 + c_2$. Figure 27 plots $p_x(x)$ with $c_1 = 2$ and $c_2 = 1$.

In our problem, the mixture spectrogram is known, and the component spectrograms need to be estimated. Therefore, we wish to maximize the likelihood in Equation 117 as a function of c_1 and c_2 . We could incorporate priors on c_1 and c_2 in a maximum *a posteriori* approach: $p(c_1, c_2|x) \propto p(x|c_1, c_2)p(c_1, c_2)$. However, we are already constraining each component to have a rank-one spectrogram and do not want to impose additional bias.

It is worth noting that many spectrogram factorization techniques incorporate a prior

distribution on the components. This is usually in the form of a prior that emphasizes the sparseness of the amplitude envelopes [2, 115, 43, 56, 57, 86, 106, 111, 116]. The difficulty with sparse priors is that they require an additional tuning parameter that represents the expected level of sparseness. The quality of the overall solution often depends on the choice of this parameter. We view the sparse prior as one of many ways that might improve the basic algorithm. However, in this thesis, the additional parameter might obscure the role of the underlying cost functions. Instead, we isolate the performance of the various cost functions, and leave the various extensions to future work.

Taken as a function of c_1 and c_2 , the likelihood of x is difficult function to optimize. For positive c_1 and c_2 , the function is only defined within the rectangular region originating at the line segment $x = c_1 + c_2$ and extending diagonally for positive c_1 and c_2 . Figure 28 shows the surface of $p(x|c_1, c_2)$ with $x = 1$. The dark lines on the $c_1 c_2$ -plane represent the boundaries of the defined region. These boundaries appear as roots of the denominator of Equation 117. In addition, there is a fourth root that corresponds to a line that runs parallel to $x = c_1 + c_2$ but never enters the positive quadrant, namely $x = -c_1 - c_2$. Figure 29 shows the four boundary lines and a contour plot in the defined region.

In order to simplify the optimization, we make three simplifications that make it more suitable for NMF-based optimization. First, we take the absolute square of $p(x|c_1, c_2)$ so that it takes a positive real value for all values of c_1 and c_2 and approaches infinity from both sides of the boundaries. By doing this, we can randomly initialize \mathbf{B} and \mathbf{H} and then make iterative improvements to these estimates. Many of the time-frequency points will start outside the defined region but during estimation will be drawn toward the boundaries. Figure 30(a) shows the original likelihood function, and Figure 30(b) shows the squared version removing the constant terms:

$$D_{\text{sqr}} = \frac{x^2}{(x + c_1 + c_2) |(x + c_1 - c_2)(x - c_1 + c_2)(x - c_1 - c_2)|} . \quad (118)$$

The second simplification involves the term $x + c_1 + c_2$ in the denominator. This corresponds to the root $x = -c_1 - c_2$ and the line in the lower-left of Figure 29. It is farthest away

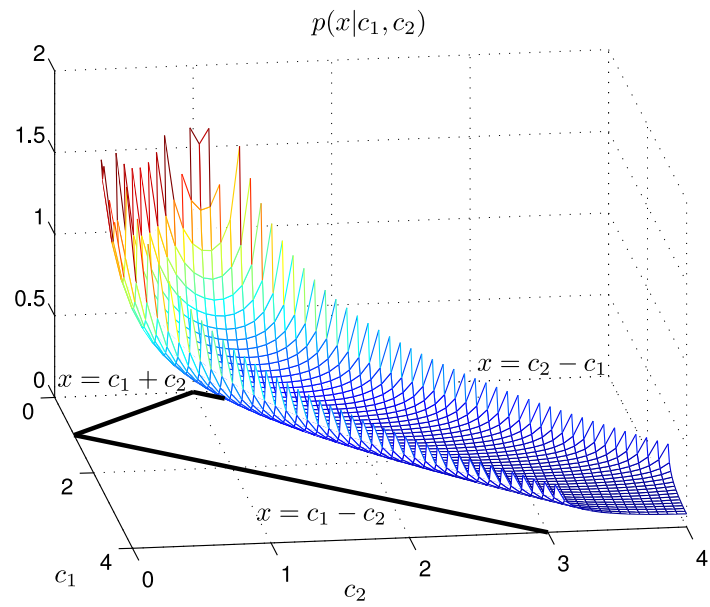


Figure 28: Surface of $p(x|c_1, c_2)$ as a function of c_1 and c_2 when $x = 1$

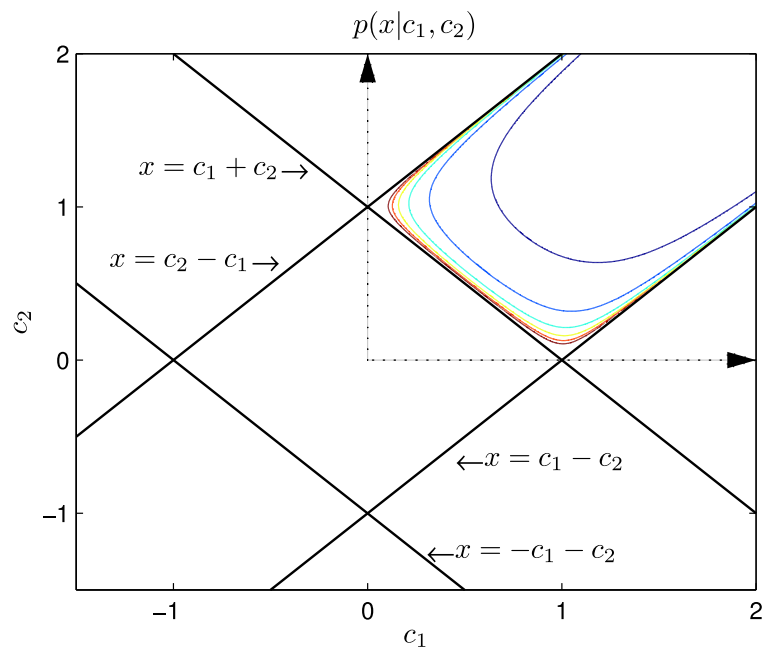


Figure 29: Surface of $p(x|c_1, c_2)$ as a function of c_1 and c_2 with $x = 1$

from the defined region that the others border, and therefore has a relatively small effect on the function. In addition, it impedes the progress of points near the origin from moving toward the defined region. Because we will have points near the origin (at least initially), and because its affect on the function is small, we simply remove the term $x + c_1 + c_2$ from the optimization function:

$$D_{\text{rem}} = \frac{x^2}{|(x + c_1 - c_2)(x - c_1 + c_2)(x - c_1 - c_2)|} . \quad (119)$$

Figure 30(c) shows the contour plot for D_{rem} . Notice that now the function is symmetric around $x = c_1 + c_2$. However, the function is still undefined on the region boundary. In order to make the function defined everywhere, we instead minimize its reciprocal:

$$D_{\text{rec}} = \frac{|(x + c_1 - c_2)(x - c_1 + c_2)(x - c_1 - c_2)|}{x^2} . \quad (120)$$

Figure 30(d) shows the contour plot for this function that reaches a minimum of zero on the boundary. Unfortunately, the function does not have a smooth gradient for a gradient-based optimization. Figure 30(e) plots the function with $x = 1$ and $c_2 = .5$. In order to make the gradient zero on the boundary, we square D_{rec} . This results in the final function that we optimize across all time-frequency points:

$$D_{\text{smooth}} = (x + c_1 - c_2)^2(x - c_1 + c_2)^2(x - c_1 - c_2)^2/x^4 . \quad (121)$$

Figure 30(f) plots D_{smooth} with $x = 1$ and $c_2 = 0.5$. Figure 31 shows the contour plot.

In a maximum likelihood optimization, the product of $p(x|c_1, c_2)$ across all time-frequency points would provide the likelihood of \mathbf{X} (as long as the time-frequency points are independent):

$$p(\mathbf{X}|\{\mathbf{C}_r\}) = \prod_{kt} p(\mathbf{X}_{kt}|\{(\mathbf{C}_r)_{kt}\}) . \quad (122)$$

If one point hits the boundary its likelihood goes to infinity and so does the product, halting the learning process. The same problem is true for our function D_{smooth} , except that it would reach a minimum of zero as soon as one point hits a boundary. Already we have diverged

from a true maximum likelihood approach by simplifying the optimization function. Now we take the sum of this function across all time-frequency points instead of the product to avoid the problem of halting when one point reaches the boundary:

$$D = \sum_{kt} (\mathbf{X}_{kt} + (\mathbf{C}_r)_{kt} - (\mathbf{C}_r)_{kt})^2 (\mathbf{X}_{kt} - (\mathbf{C}_r)_{kt} + (\mathbf{C}_r)_{kt})^2 (\mathbf{X}_{kt} - (\mathbf{C}_r)_{kt} - (\mathbf{C}_r)_{kt})^2 / \mathbf{X}_{kt}^4 . \quad (123)$$

This emphasizes a solution in which all points are near the boundaries (not just one).

4.3.2.1 Update Rules

We minimize the function D , which is proportional to the sum of D_{smooth} across all time-frequency points:

$$D = \frac{1}{2} \sum_{kt} \mathbf{P}_{kt}^2 \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}^2 / \mathbf{X}_{kt}^4 , \quad (124)$$

where

$$\mathbf{P} = \mathbf{X} + \mathbf{C}_1 - \mathbf{C}_2 , \quad (125)$$

$$\mathbf{Q} = \mathbf{X} - \mathbf{C}_1 + \mathbf{C}_2 , \quad (126)$$

$$\mathbf{R} = \mathbf{X} - \mathbf{C}_1 - \mathbf{C}_2 , \quad (127)$$

and all the operations are element-wise. Taking the derivative of D with respect to each of the columns of \mathbf{B} and rows of \mathbf{H} (from Equation 52) yields the following:

$$\frac{\partial D}{\partial \mathbf{B}_{k1}} = \sum_t \mathbf{H}_{1t} (\mathbf{P}_{kt} \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt} \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}) / \mathbf{X}_{kt}^4 \quad (128)$$

$$\frac{\partial D}{\partial \mathbf{H}_{1t}} = \sum_k \mathbf{B}_{k1} (\mathbf{P}_{kt} \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt} \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}) / \mathbf{X}_{kt}^4 \quad (129)$$

$$\frac{\partial D}{\partial \mathbf{B}_{k2}} = \sum_t \mathbf{H}_{2t} (-\mathbf{P}_{kt} \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}^2 + \mathbf{P}_{kt}^2 \mathbf{Q}_{kt} \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}) / \mathbf{X}_{kt}^4 \quad (130)$$

$$\frac{\partial D}{\partial \mathbf{H}_{2t}} = \sum_k \mathbf{B}_{k2} (-\mathbf{P}_{kt} \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}^2 + \mathbf{P}_{kt}^2 \mathbf{Q}_{kt} \mathbf{R}_{kt}^2 - \mathbf{P}_{kt}^2 \mathbf{Q}_{kt}^2 \mathbf{R}_{kt}) / \mathbf{X}_{kt}^4 . \quad (131)$$

We randomly initialize \mathbf{B} and \mathbf{H} , and minimize D using gradient descent with additive updates.

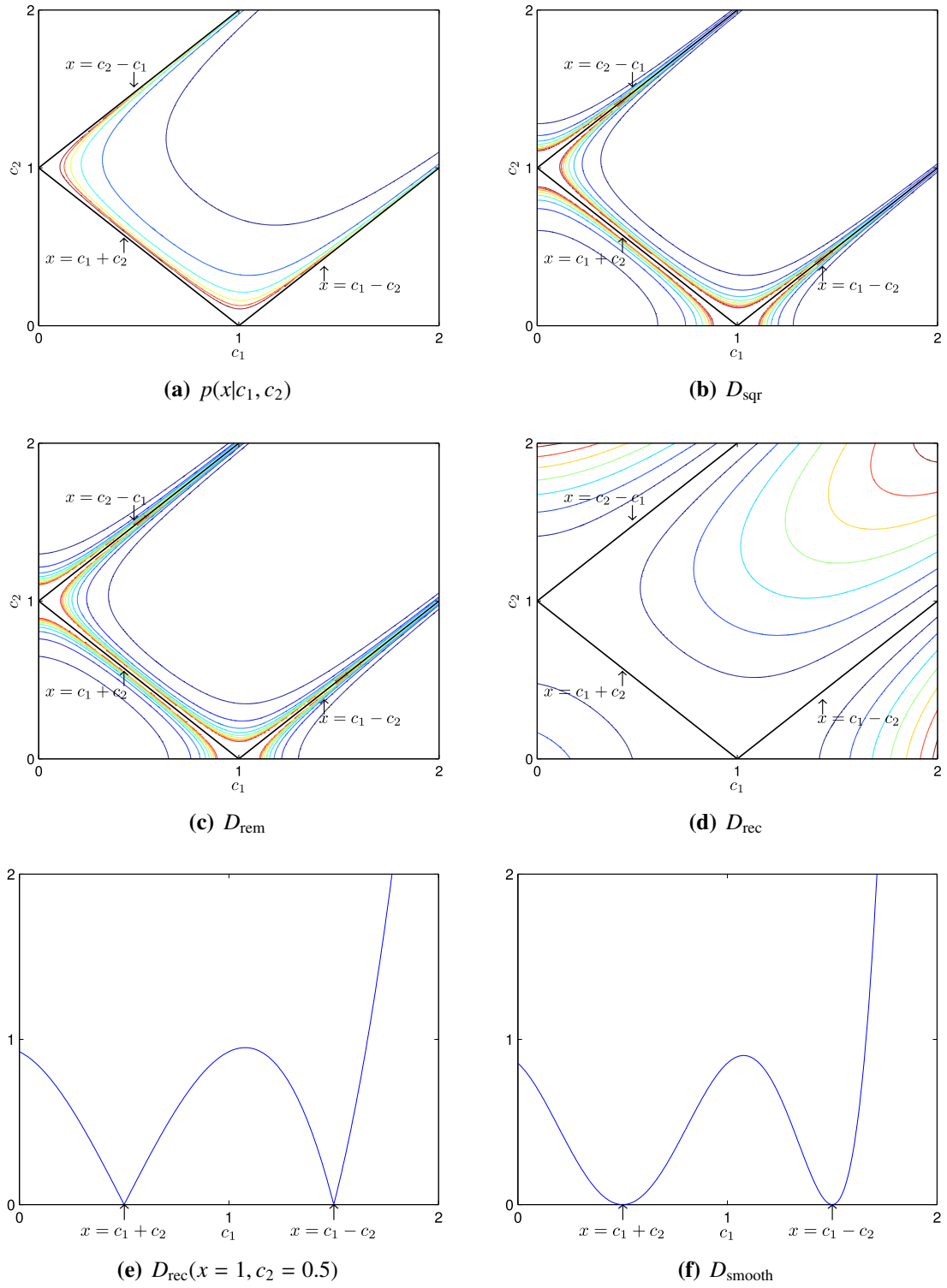


Figure 30: Plots of the intermediate functions between $p(x|c_1, c_2)$ and the optimization function

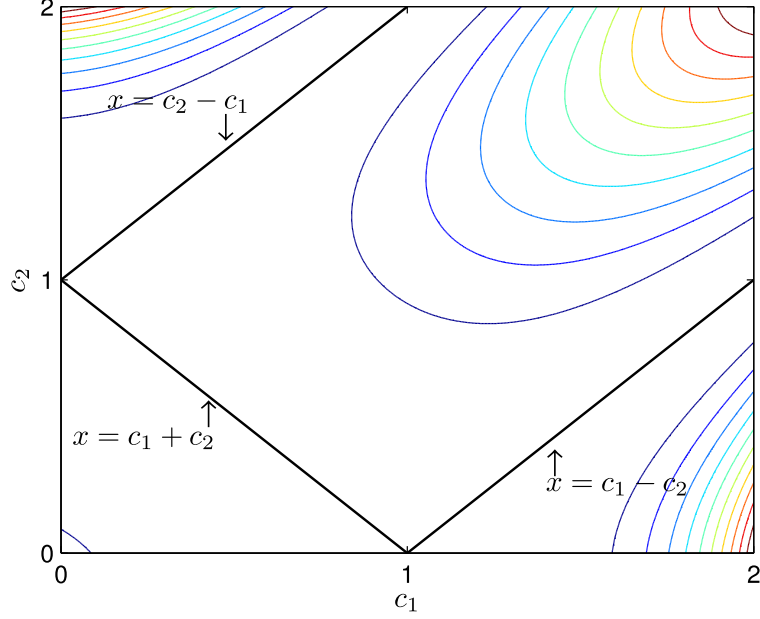


Figure 31: Contour plot of the cost function D_{smooth}

4.3.2.2 Improved Component Estimation Using Phase Information

In order to compare our probabilistic phase algorithm against standard non-negative matrix factorization we construct source and mixture spectrograms as follows:

$$\mathbf{B}_{kr} = |N(0, 1)| \quad \mathbf{H}_{rt} = |N(0, 1)| \quad (132)$$

$$(\boldsymbol{\Theta}_1)_{kt} = U(-\pi, \pi) \quad (\boldsymbol{\Theta}_2)_{kt} = U(-\pi, \pi) \quad (133)$$

$$(\mathbf{C}_1)_{kt} = \mathbf{B}_{k1}\mathbf{H}_{1t} \quad (\mathbf{C}_2)_{kt} = \mathbf{B}_{k2}\mathbf{H}_{2t} \quad (134)$$

$$\mathcal{F}_{c_1}(k, t) = (\mathbf{C}_1)_{kt}e^{i(\boldsymbol{\Theta}_1)_{kt}} \quad \mathcal{F}_{c_2}(k, t) = (\mathbf{C}_2)_{kt}e^{i(\boldsymbol{\Theta}_2)_{kt}} \quad (135)$$

$$\mathcal{F}_x(k, t) = \mathcal{F}_{c_1}(k, t) + \mathcal{F}_{c_2}(k, t) \quad \mathbf{X} = |\mathcal{F}_x(k, t)|. \quad (136)$$

We choose $K = T = 100$, $R = 2$, and run both algorithms for 1000 trials, each time drawing new source spectrograms and initializing NMF with new random matrices. We initialize our approach with the NMF solution. The scatter plot of time-frequency bins from one representative trial is shown in Figure 32. Each point represents one time-frequency point of the component spectrograms. For illustrative purposes, we normalize the position of

each point to conform to the $x = 1$ scale. That is, we place each time-frequency bin at the point $(c_1/x, c_2/x)$. Notice that our approach in Figure 32(d) more closely resembles the correct scatter plot in Figure 32(a) than traditional NMF in Figure 32(c). Standard NMF minimizes the distance of each bin to the line $x = c_1 + c_2$ (*i.e.*, the line between $(0, 1)$ and $(1, 0)$ in the normalized space). Our approach additionally minimizes the distance to the lines $x = c_1 - c_2$ and $x = c_2 - c_1$ (*i.e.*, the parallel diagonal lines in the figure).

Because our cost function makes it difficult for bins to cross boundary lines, we use the NMF solution for initialization. During the NMF phase, bins move freely toward the boundary $x = c_1 + c_2$. We believe that this allows bins to orient themselves toward the top or bottom parallel boundary line without restriction. We then use our criterion function to favor solutions that minimize the distance to all three boundaries.

Figure 33 shows the combined histograms for all trials. The histogram for the correct solutions in Figure 33(a) resembles the function in Figure 28 and has long tails along the $x = c_1 - c_2$ and $x = c_2 - c_1$ boundary lines. Figure 33(b) shows the initial solution drawn from a positive Gaussian distribution. Notice that our approach in Figure 33(d) has visible tails similar to the correct histogram, whereas NMF in Figure 33(c) does not.

The visual difference between the methods accounts for an improvement in the mean square error between the actual and estimated components. We first normalize the columns of \mathbf{B} and the rows of \mathbf{H} to unit L_2 norm and compute the mean square error as follows:

$$MSE = \frac{1}{KR} \sum_{kr} (\hat{\mathbf{B}}_{kr} - \mathbf{B}_{kr})^2 + \frac{1}{RT} \sum_{rt} (\hat{\mathbf{H}}_{rt} - \mathbf{H}_{rt})^2. \quad (137)$$

Over the 1000 trials, the mean square error for NMF is 3.37×10^{-4} , whereas our approach attains a mean square error of 2.43×10^{-4} for an improvement of 28%.

It is important to note that although our method improves on the estimates of the components, neither approach produces an estimate within the defined region of the original likelihood function. More sophisticated learning algorithms could constrain the solution to this region and potentially improve estimates.

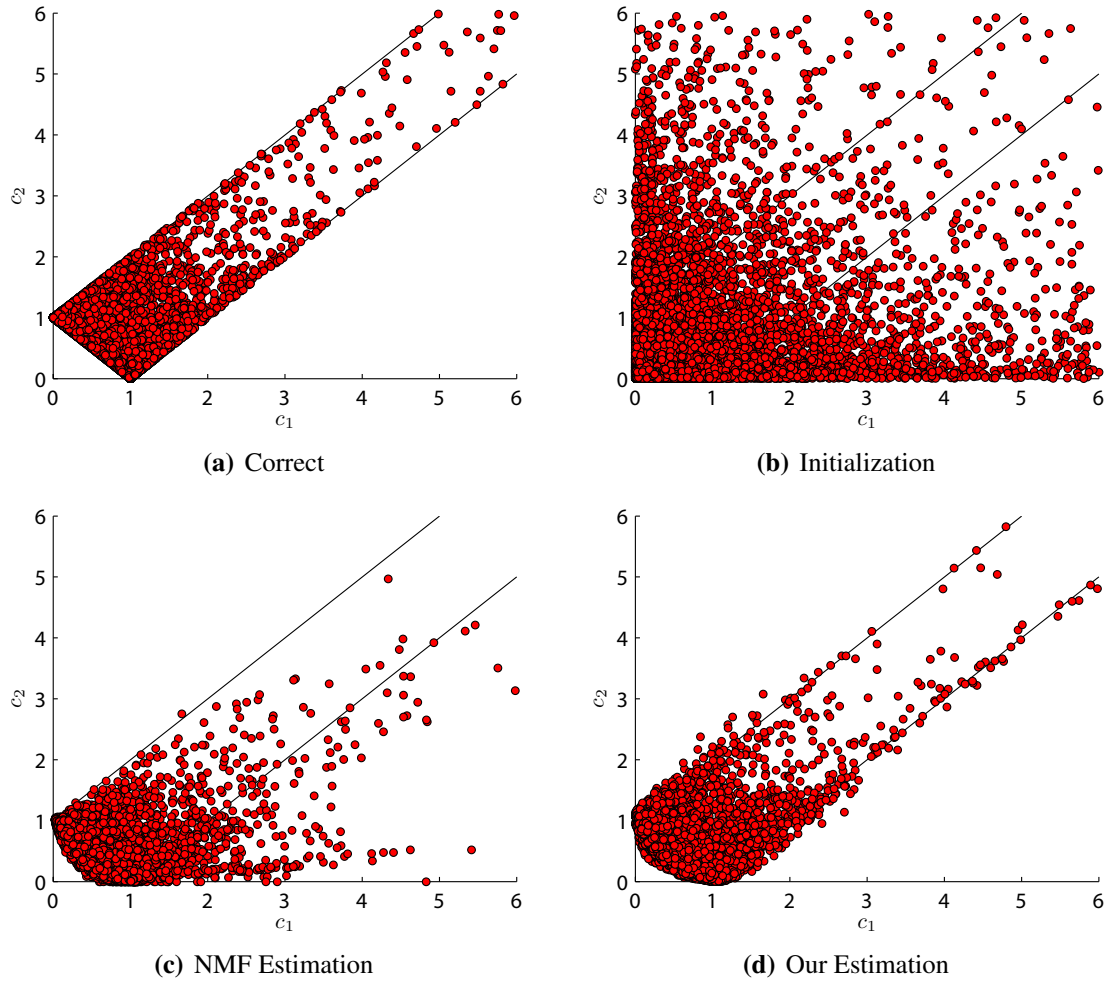
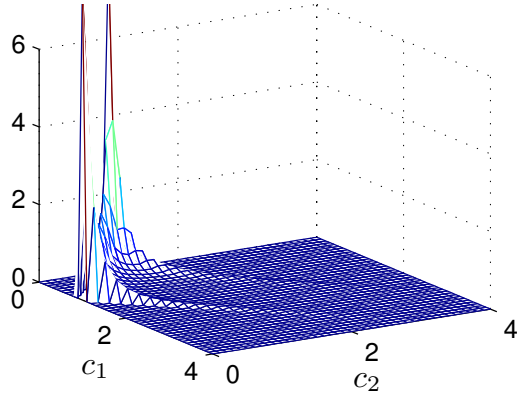
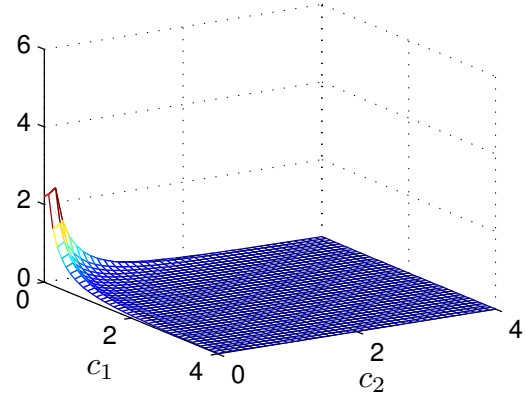


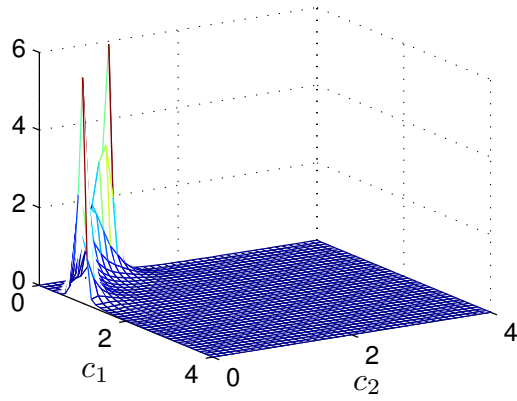
Figure 32: Scatter plot of bins for one representative trial



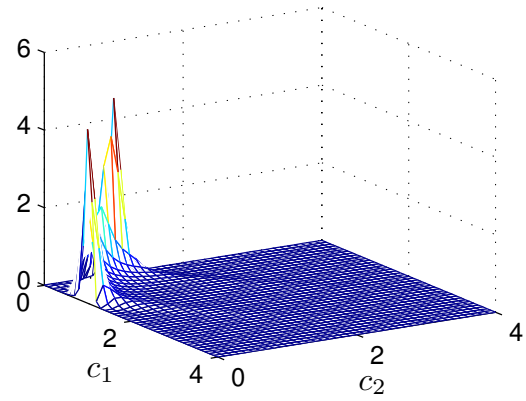
(a) Correct



(b) Initialization



(c) NMF Estimation



(d) Our Estimation

Figure 33: Histogram for all trials in units of 10^5

4.3.3 Extension to More Than Two Components

In this section, we incorporate phase information to improve non-negative spectrogram factorization for the case of more than two components. Deriving the explicit likelihood of \mathbf{X} for the case of more than two components (analogous to Equation 117 for two components) has proven exceedingly difficult. Instead, we estimate the likelihood using the central limit theorem to capture the shape of the distribution for a large number of components. We also make the simplifying assumption that the phase is independent at different time-frequency points. To some degree, this is true. However, the unwrapped phase of a steady state signal can be approximated from the previous two time-steps [15, 14]. Although this violates the independence assumption, we have found that the resulting approach works well in practice.

The probability density function for a complex random variable with magnitude c_r and uniform random phase has a mean of zero and a variance of c_r^2 . According to the Lindeberg-Feller central limit theorem [38], the sum of many such variables tends toward a complex Gaussian with zero mean and a variance of $\sum_r c_r^2$. This theorem is valid under the Lindeberg condition, which states that the component variances, c_r^2 , are small relative to their sum [38]. Applied to magnitude spectrograms we have the following:

$$p(\mathcal{F}_x | \mathbf{C}_1, \dots, \mathbf{C}_R) = \prod_{kt} \frac{1}{\pi \Lambda_{kt}} \exp\left(-\frac{\mathbf{X}_{kt}^2}{\Lambda_{kt}}\right), \quad (138)$$

where $\Lambda_{kt} = \sum_r (\mathbf{C}_r)_{kt}^2$. We find the likelihood of \mathbf{X} by integrating with respect to phase, resulting in a Rayleigh distribution:

$$p(\mathbf{X} | \mathbf{C}_1, \dots, \mathbf{C}_R) = \prod_{kt} \frac{2\mathbf{X}_{kt}}{\Lambda_{kt}} \exp\left(-\frac{\mathbf{X}_{kt}^2}{\Lambda_{kt}}\right). \quad (139)$$

Figure 34 shows the histogram of samples of \mathbf{X} drawn from uniformly distributed component magnitudes and phases. As the number of components increases, they approach a Rayleigh distribution indicated by the red (dark gray in grayscale) line.

An interesting result is the histogram for three components. It looks similar to the two component likelihood in Figure 27 except that it has two tails on either side of the

bimodal distribution. For convenience, let the components have the following ordering: $c_1 \geq c_2 \geq c_3$. When x is the sum of two components, the peaks in Figure 27 represent the increased likelihood that x is within a small region near $c_1 + c_2$ or $c_1 - c_2$. This is due to the slow change in x when the magnitude of the phase difference is near zero or π in Equation 111. In Figure 34(a), c_1 happens to be greater than the sum $c_2 + c_3$, leading to tails that end abruptly at the boundaries $x = c_1 - c_2 - c_3$ and $x = c_1 + c_2 + c_3$. The region between the two peaks represents the domain where for all values of θ_1 and θ_2 , there exists a θ_3 that produces x . The tails represent the diminished likelihood that x takes a value where θ_1 and θ_2 must be constrained to produce a particular value of x . For example, at the boundaries, $\theta_1 = \theta_2$ or $\theta_1 = -\theta_2$. The peaks represent the increased likelihood that x is within a small region near $c_1 + c_2 - c_3$ or $c_1 - c_2 + c_3$. This is due to the slow change in x when all pairs of components are either nearly in-phase or π radians out-of-phase. For each additional component, the number of these “peaks” doubles until the peaks are indistinguishable from the valleys and it approaches a Rayleigh distribution.

4.3.3.1 Maximum Likelihood

In order to estimate \mathbf{C}_r , we propose minimizing the negative log likelihood of \mathbf{X} :

$$-\log p(\mathbf{X}|\mathbf{C}_1, \dots, \mathbf{C}_R) = -\sum_{kt} \left[\log \left(\frac{2\mathbf{X}_{kt}}{\Lambda_{kt}} \right) - \frac{\mathbf{X}_{kt}^2}{\Lambda_{kt}} \right]. \quad (140)$$

For comparison, we frame our maximum likelihood approach in terms of a cost function. The minimum of Equation 140 is $1 - \log(2/\mathbf{X}_{kt})$ at $\Lambda_{kt} = \mathbf{X}_{kt}^2$. By subtracting this value we find a cost function that is non-negative reaching zero only when all $\Lambda_{kt} = \mathbf{X}_{kt}^2$:

$$D_s = \sum_{kt} \frac{\mathbf{X}_{kt}^2}{\Lambda_{kt}} - 1 + \log \left(\frac{\Lambda_{kt}}{\mathbf{X}_{kt}^2} \right), \quad (141)$$

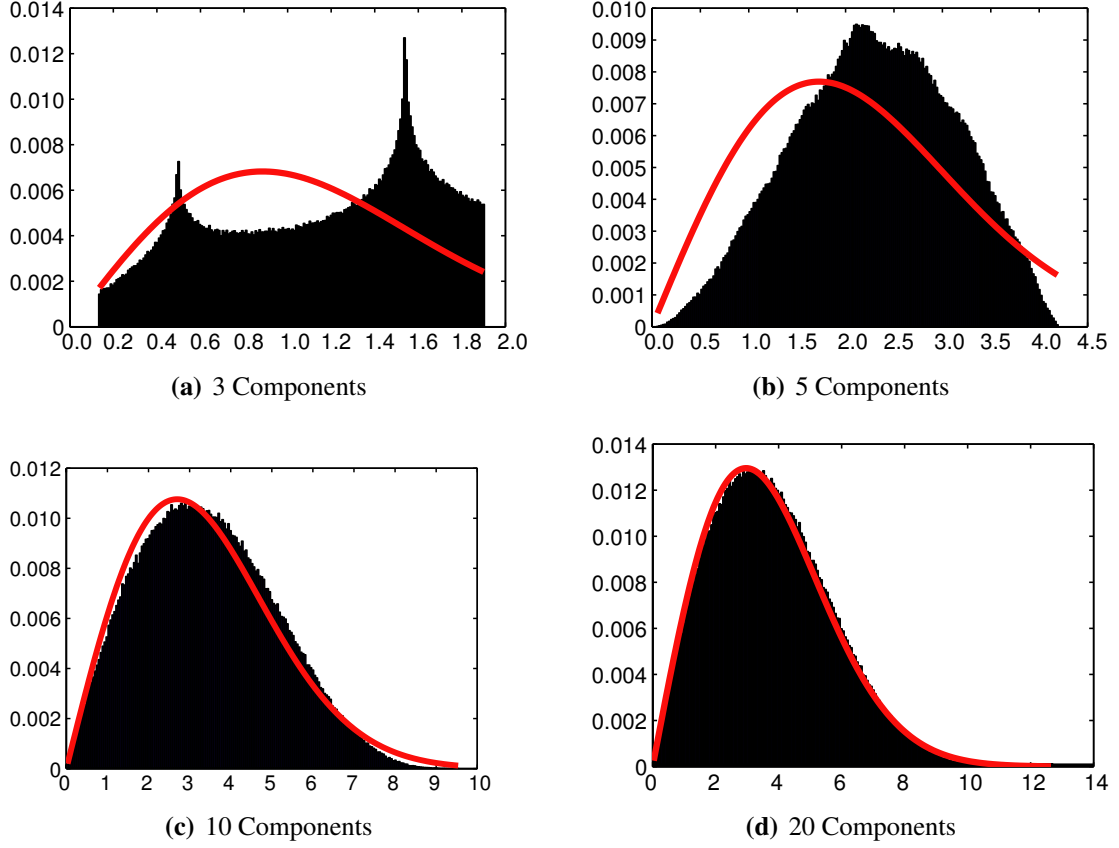


Figure 34: As the number of components increases, \mathbf{X} approaches a Rayleigh distribution

which is equivalent to Equation 8 in Abdallah and Plumbley [2]. We derive the gradient for

D_s with respect to \mathbf{B}_{kr}^2 and \mathbf{H}_{rt}^2 :

$$\frac{\partial D_s}{\partial (\mathbf{B}_{kr}^2)} = \sum_t \mathbf{H}_{rt}^2 \left(\frac{\Lambda_{kt} - \mathbf{X}_{kt}^2}{\Lambda_{kt}^2} \right) \quad (142)$$

$$\frac{\partial D_s}{\partial (\mathbf{H}_{rt}^2)} = \sum_k \mathbf{B}_{kr}^2 \left(\frac{\Lambda_{kt} - \mathbf{X}_{kt}^2}{\Lambda_{kt}^2} \right), \quad (143)$$

where $\Lambda_{kt} = \sum_r \mathbf{B}_{kr}^2 \mathbf{H}_{rt}^2$. Although D_s is not convex with respect to \mathbf{B}_{kr}^2 or H_{rt}^2 , we find local minima using the following multiplicative update rules:

$$\mathbf{B}_{kr}^2 \leftarrow \mathbf{B}_{kr}^2 \frac{\sum_t \mathbf{H}_{rt}^2 \mathbf{X}_{kt}^2 / \Lambda_{kt}^2}{\sum_t \mathbf{H}_{rt}^2 / \Lambda_{kt}} \quad (144)$$

$$\mathbf{H}_{rt}^2 \leftarrow \mathbf{H}_{rt}^2 \frac{\sum_k \mathbf{B}_{kr}^2 \mathbf{X}_{kt}^2 / \Lambda_{kt}^2}{\sum_k \mathbf{B}_{kr}^2 / \Lambda_{kt}}. \quad (145)$$

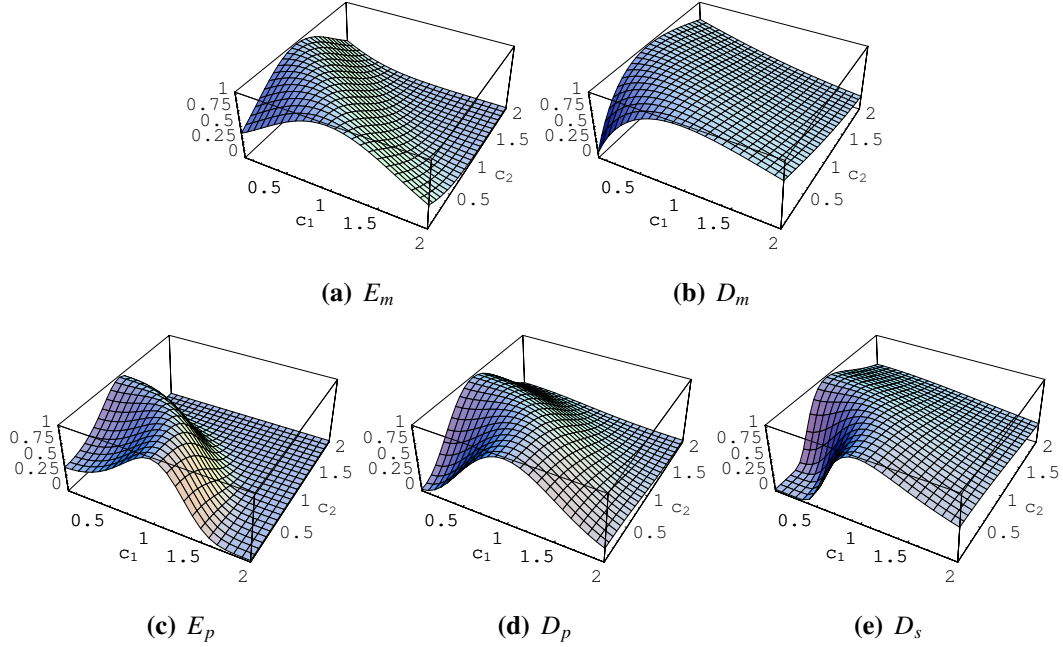


Figure 35: The shape of the likelihood functions derived from the 5 labeled cost functions for the case of two components and $x = 1$

4.3.3.2 Comparison to Other Cost Functions

We compare the phase-aware cost function, D_s , to four other cost functions based on Euclidean or generalized Kullback-Leibler divergence for magnitude or power spectrograms. Figure 35 plots the shape of the likelihood functions for each of the cost functions with $x = 1$. Magnitude spectrogram methods ($E_m = \|\mathbf{X} - \mathbf{BH}\|^2$ and $D_m = D(\mathbf{X}||\mathbf{BH})$) reach a maximum on the line $c_1 + c_2 = x$. Power spectrogram methods ($E_p = \|\mathbf{X}^2 - \mathbf{\Lambda}\|^2$, $D_p = D(\mathbf{X}^2||\mathbf{\Lambda})$, and D_s) reach a maximum on the circle $c_1^2 + c_2^2 = x^2$. When $x = 1$, the sum of c_1 and c_2 must be greater than one. D_s encourages this result by penalizing solutions near the origin more than the other cost functions.

4.3.3.3 Experimental Results

We evaluate the performance of the cost functions for a variety of spectrogram sizes, numbers of components, and component distributions. Specifically, we construct square spectrograms and vary their size with $K = T \in [32, 64, 128, 256, 512, 1024]$, $R \in [1, \dots, 30]$,

and \mathbf{B} and \mathbf{H} drawn from the uniform, positive normal, or exponential distribution. After drawing \mathbf{B} and \mathbf{H} from the specified distribution, we construct \mathbf{X} using Equations 133–136 in the previous section. We then estimate \mathbf{B} and \mathbf{H} using the multiplicative update rules for each cost function (Equations 61 and 62 for E_m and E_p , Equations 67 and 68 for D_m and D_p , and Equations 144 and 145 for D_s). Because scaling \mathbf{B} by α and \mathbf{H} by $1/\alpha$ produces the same cost, we normalize the rows of H to unit L_2 norm after every update.

We evaluate each cost function according to the mean square error between the original and estimated $\{C_r\}$. Because the factorization technique is permutation invariant, we must determine the mapping between each estimated and original C_r . For this purpose, we use a greedy algorithm that matches the two most similar components (one original and one estimated) and then removes them from consideration. The process repeats until the mapping is complete.

Figure 36 plots the average performance over five trials for each configuration of parameters with uniformly distributed components. For clarity, we only show $1 \leq R \leq 10$. Each of the 60 $[R, K]$ pairs are sorted along the x-axis in order of increasing minimum error among the five cost functions. Clearly, the problem becomes more difficult as R increases or as K decreases.

The bottom of Figure 36 plots the mean square estimation error. For simpler versions of the problem, D_s outperforms the rest. However, toward the right of the plot the performance becomes markedly worse and E_m and D_m perform better. This inversion of performance is linked to the detection rate.

The top of Figure 36 plots the detection rate. When each estimated component uniquely matches a real component, the detection rate is 100%. However, when none of the estimated components match one of the real components, that component is not detected. We compute the detection rate as the fraction of real components that are the closest match (in the mean square sense) for at least one estimated component. At $[R, K] = [4, 32]$, the detection rate for D_s drops below 100% for the first time and this corresponds to the first

large increase in estimation error. After that, the estimation rate for D_s accelerates until it is the worst of the group.

The detection rate is another indication of the difficulty of each factorization. The magnitude spectrogram methods, E_m and D_m perform better than the power spectrogram methods for the more difficult problems in spite of detection errors. Interestingly, even if we initialize the power spectrogram methods with the E_m solution the results are qualitatively the same (Figure 37); D_s performs better than the rest until detection becomes a problem (near $[R, K] = [4, 32]$), after which the D_s error accelerates until it is the worst in the group. If we initialize with the true solution (Figure 38), the detection rates improve and D_s maintains its advantage for more difficult problems. However, across the seven most difficult problems it accelerates from nearly the best to nearly the worst performance. It appears that in the extremely difficult cases there is simply not enough data to leverage the phase-aware model.

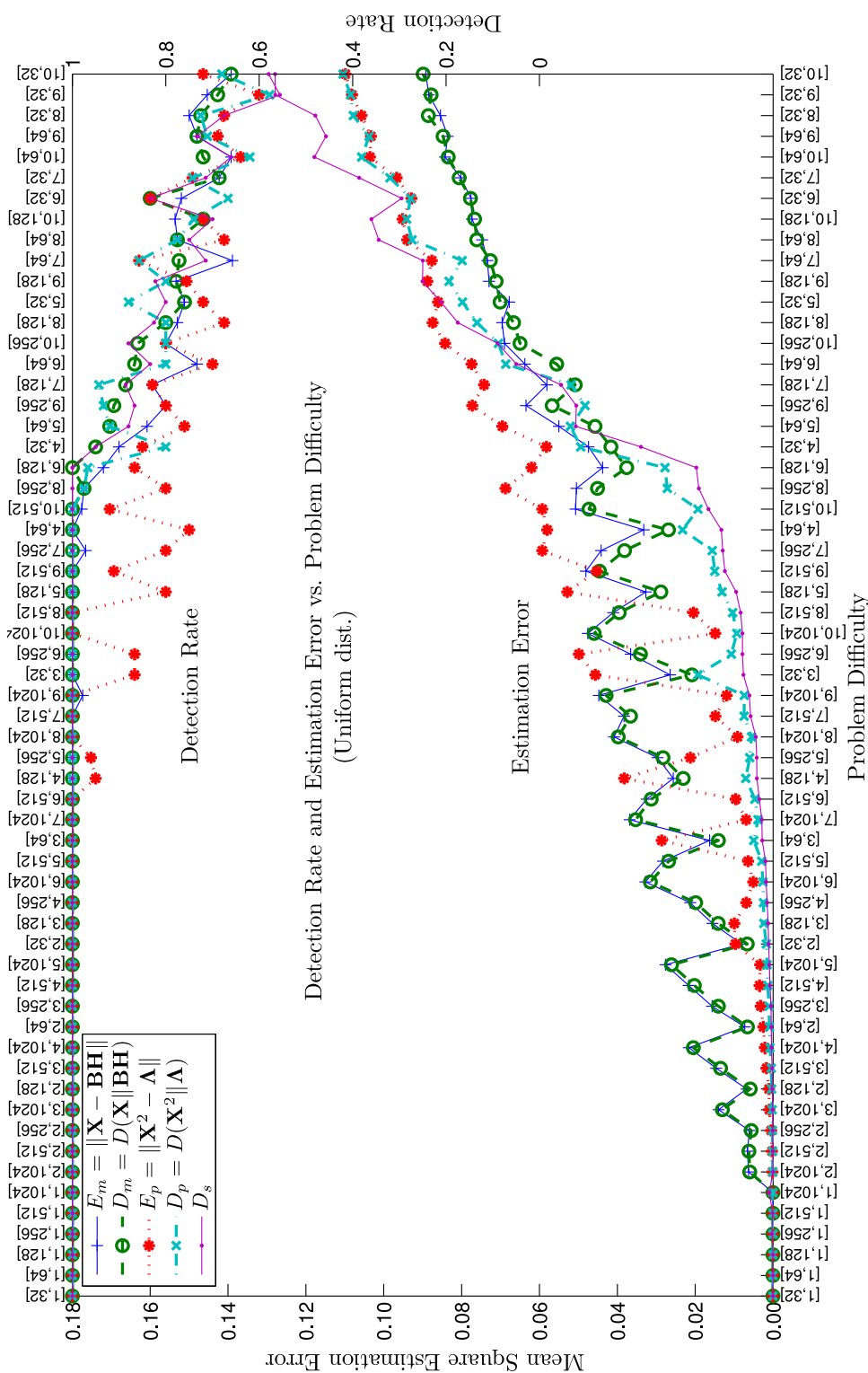


Figure 36: Estimation error and detection rate for components drawn from a uniform distribution

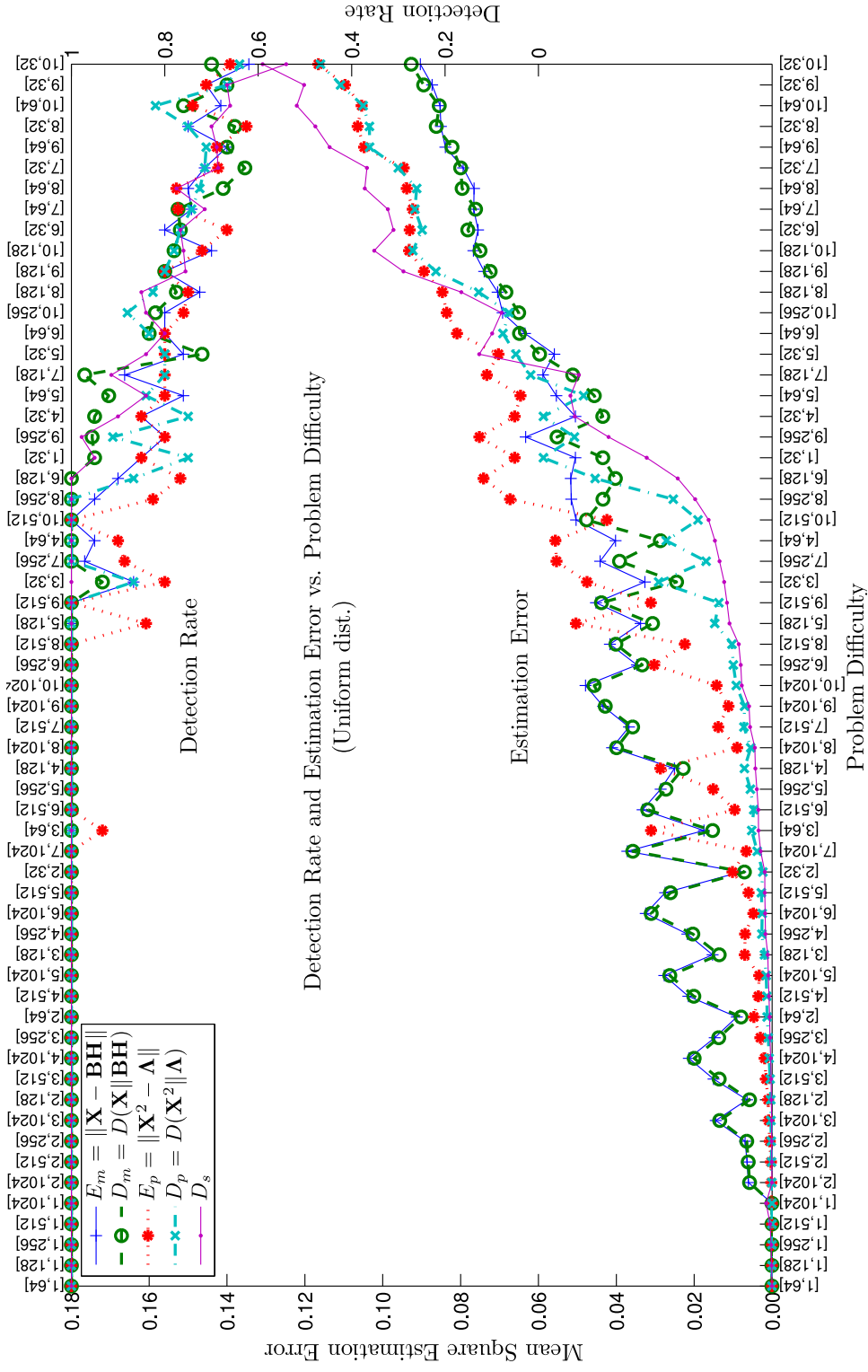


Figure 37: Estimation error and detection rate initializing E_p , D_p , and D_s with the E_m solution.

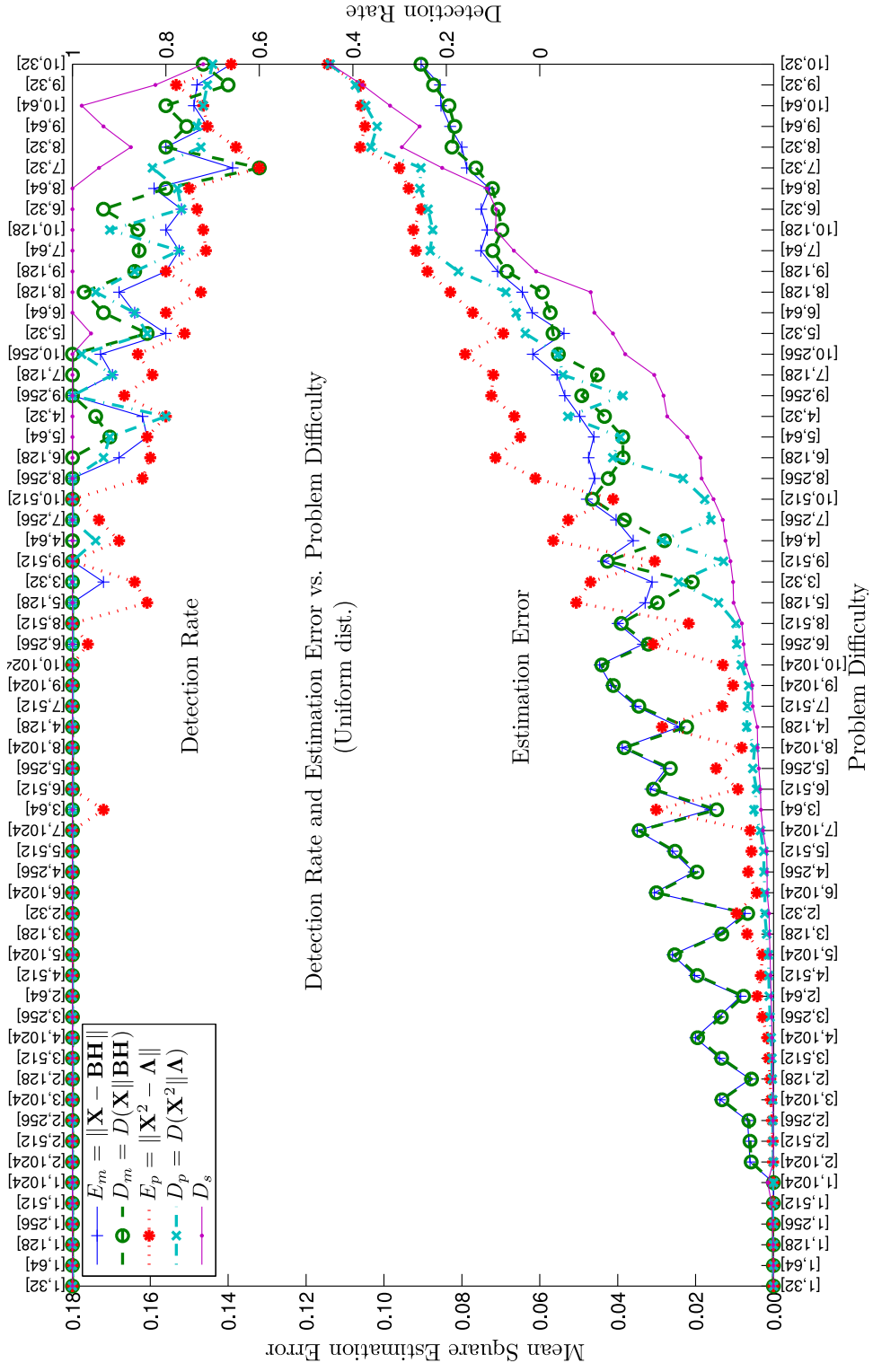


Figure 38: Estimation error and detection rate initializing all methods with the true solution

The underlying distribution of \mathbf{B} and \mathbf{H} also affects estimation and detection. Figure 39 and 40 plots the error and detection rate for positive normally and exponentially distributed components, respectively. As presented, the cost functions implicitly assume a uniform prior distribution on \mathbf{B} and \mathbf{H} in the maximum likelihood framework. Therefore, as the component distributions diverge from the uniform distribution (*e.g.*, become more sparse) the maximum likelihood approach becomes less realistic. The aggregated mean square error for the uniform, positive normal (more sparse), and exponential (most sparse) distribution is 0.036, 0.19, and 0.44, respectively. However, sparseness has the opposite effect on detection. All of the cost functions attain 100% detection for more problems as sparseness increases. Table 6 lists the number of problems that resulted in 100% detection and the number of times each algorithm provides the best estimation error for each of the distributions and R between 2 and 10. However, for more difficult problems with poor detection rate, the magnitude spectrogram methods perform better. Figure 41-43 show the difficulty of power spectrogram methods as the detection rate decreases for all trials ($1 \leq R \leq 30$).

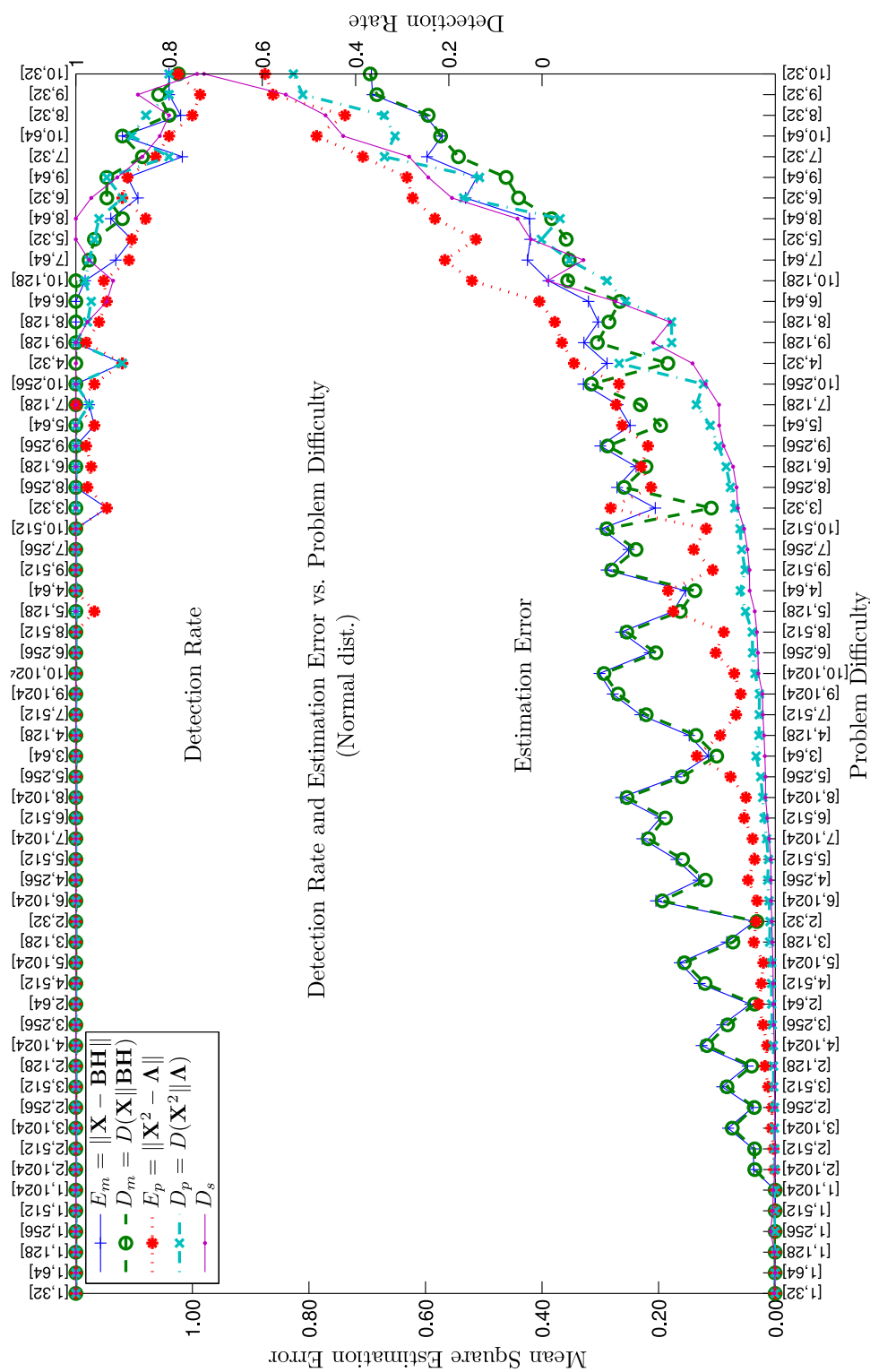


Figure 39: Estimation error and detection rate for components drawn from a positive normal distribution

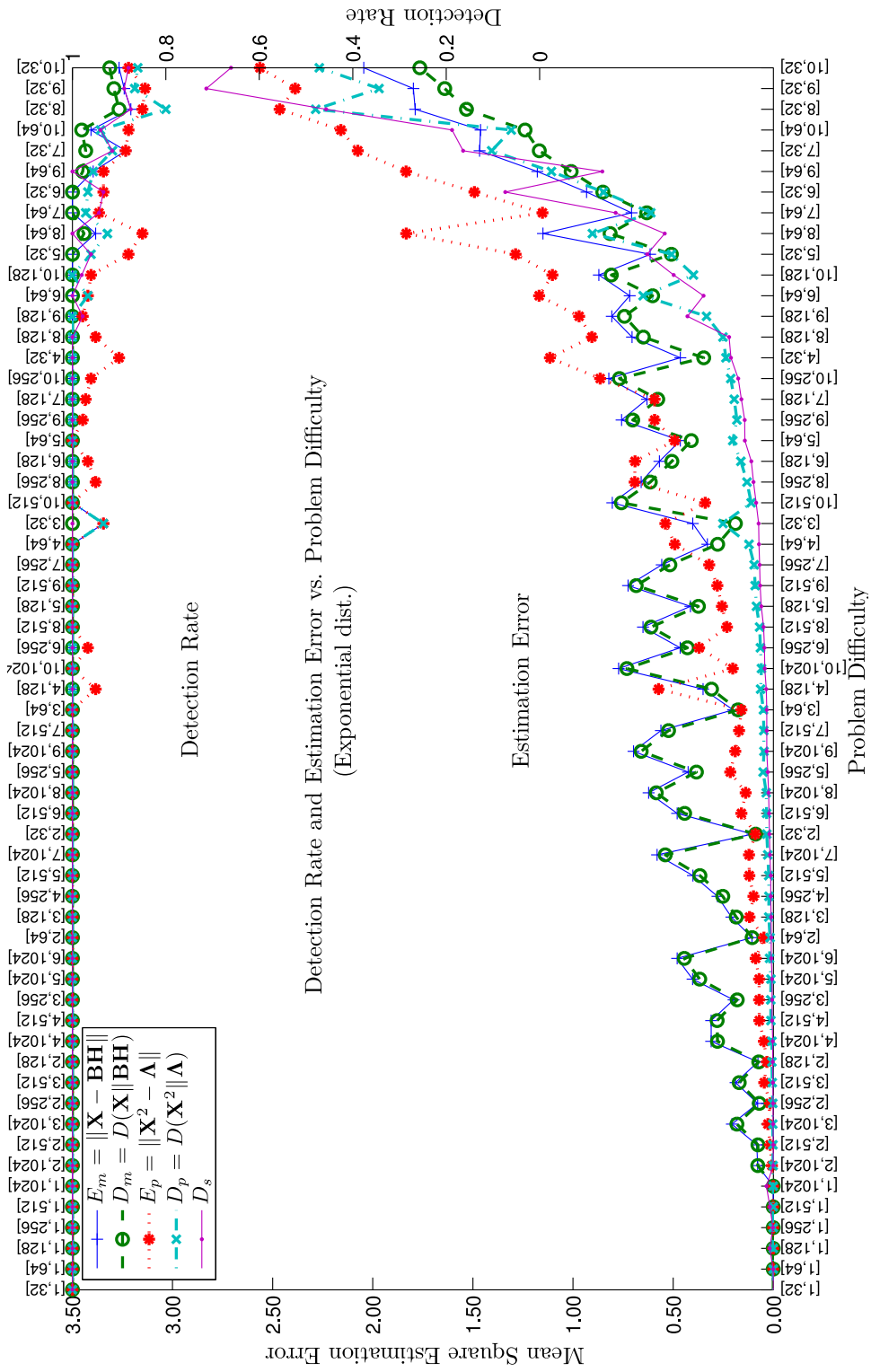


Figure 40: Estimation error and detection rate for components drawn from an exponential distribution

Table 6: Summary of detection rate and lowest estimation error for $R = [2, 10]$

Distribution:	Uniform		Positive Normal		Exponential	
Cost func.	100% det.	Best est.	100% det.	Best est.	100% det.	Best est.
E_m	27	9	37	3	44	0
D_m	34	8	43	6	47	6
E_p	23	0	29	0	30	0
D_p	33	0	38	4	41	3
D_s	35	37	40	41	42	45
Total	152	54	187	54	204	54

We speculate that if detection could be improved, D_s would maintain its advantage for more difficult problems. To test this, we repeated the experiment providing each algorithm with the correct \mathbf{B} matrix and estimated only \mathbf{H} . This simplification of the problem increases the detection rate and improves the estimation performance for all methods, especially D_p and D_s . Figures 44-46 show the improvement of all cost functions for this test. However, the power spectrogram methods improve the most. In particular, D_s maintains its advantage for more difficult problems particularly for the more sparse distributions.

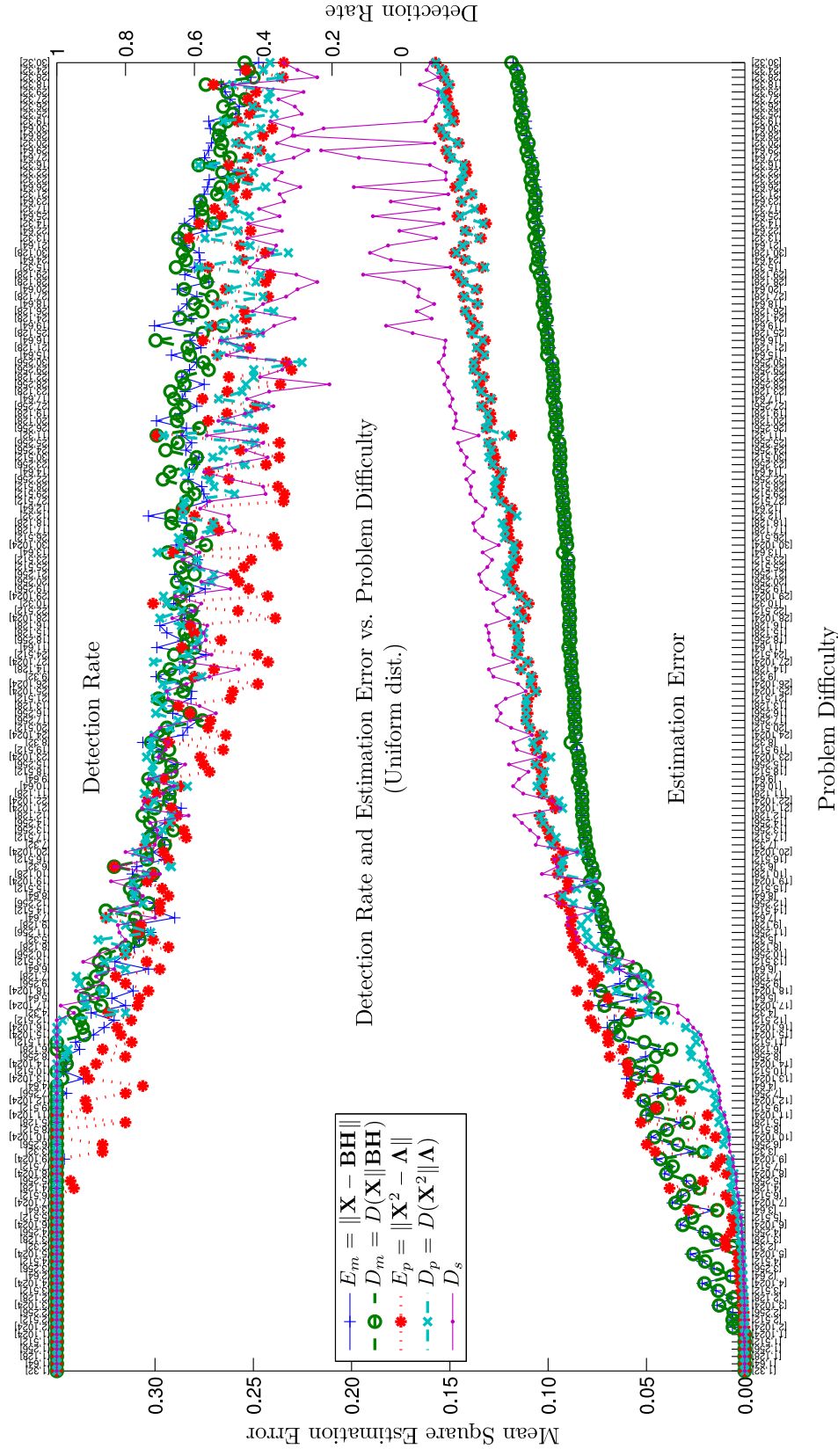


Figure 41: Estimation error and detection rate for components drawn from a uniform distribution

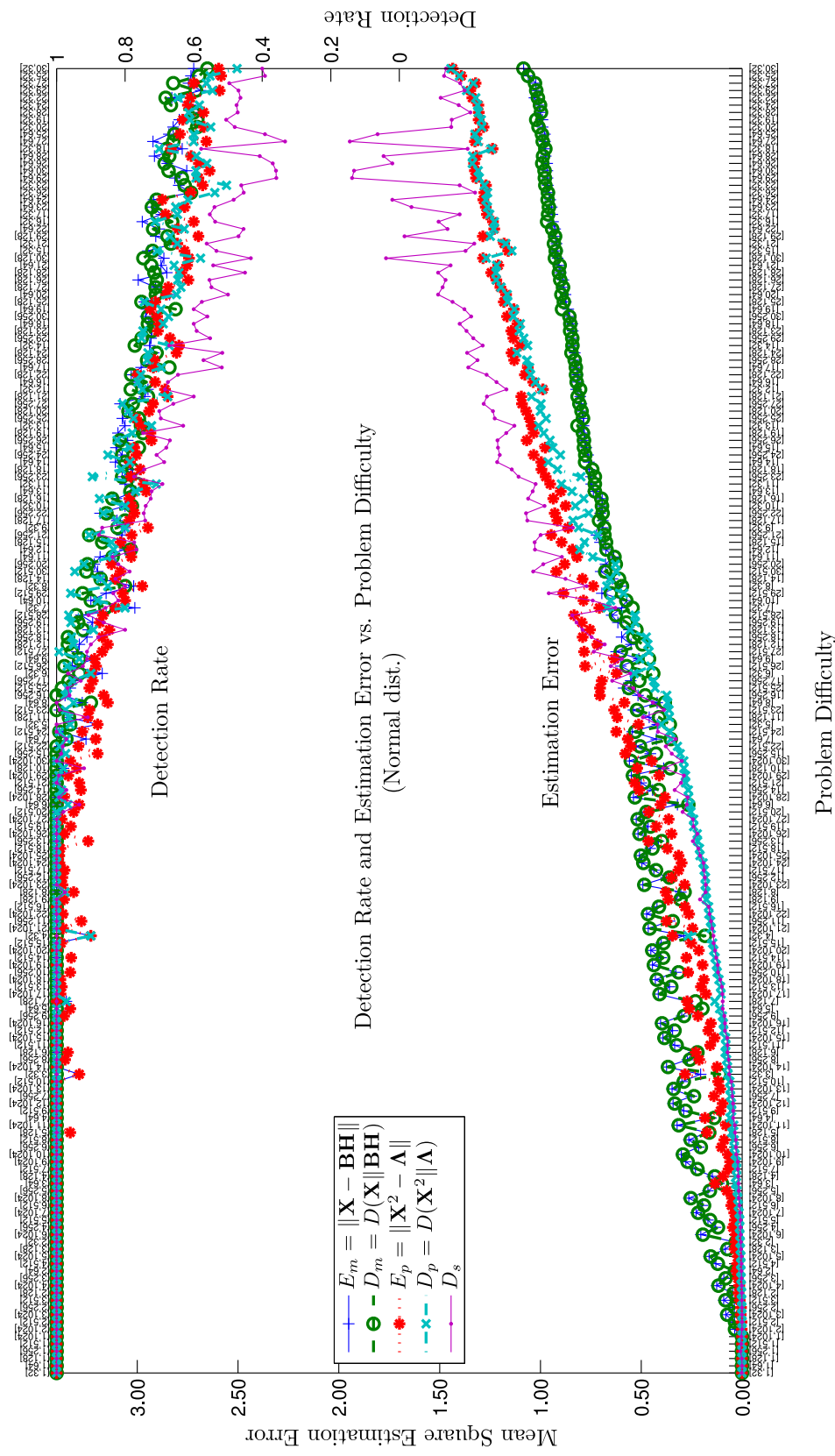


Figure 42: Estimation error and detection rate for components drawn from a positive normal distribution

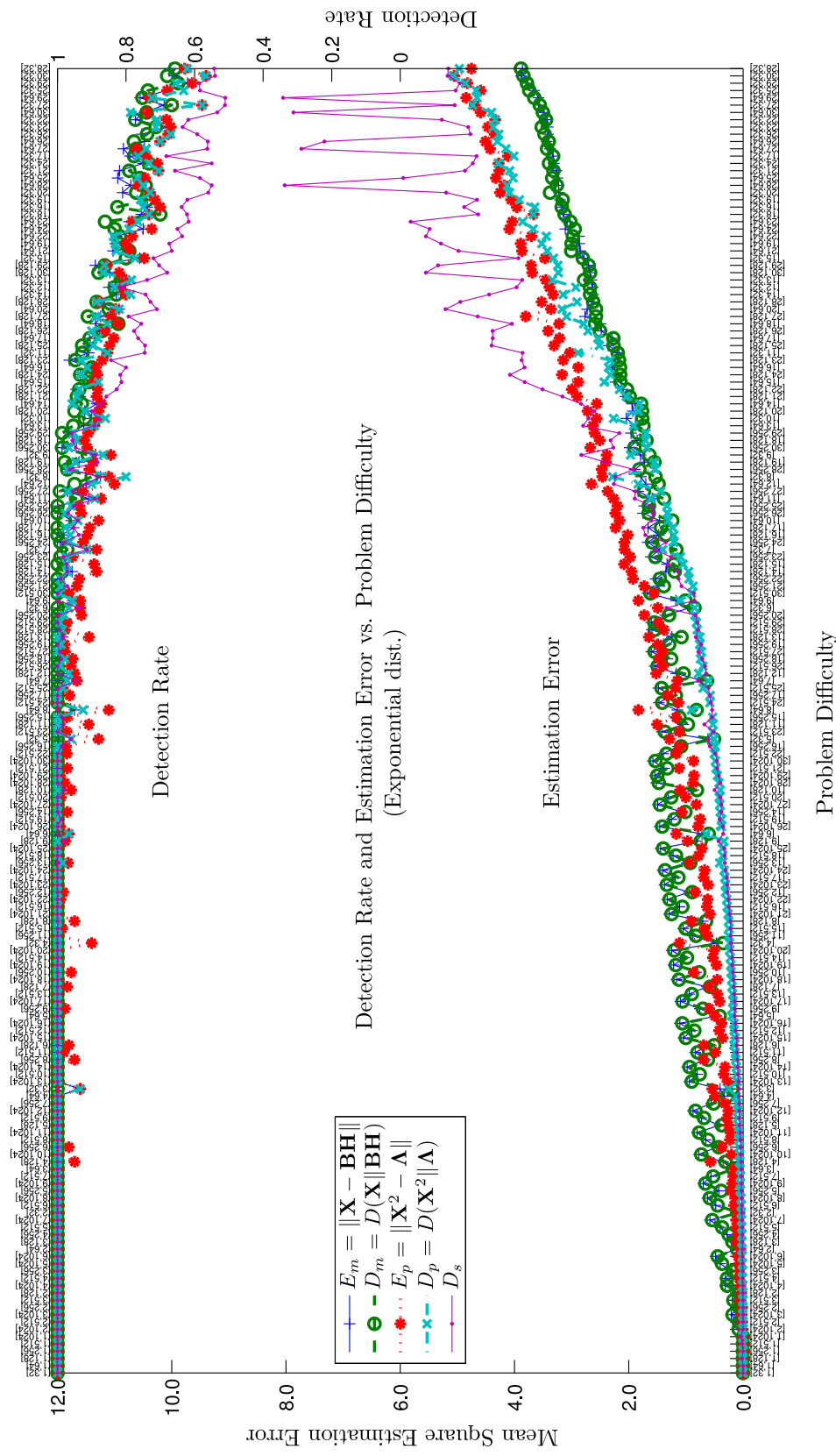


Figure 43: Estimation error and detection rate for components drawn from an exponential distribution

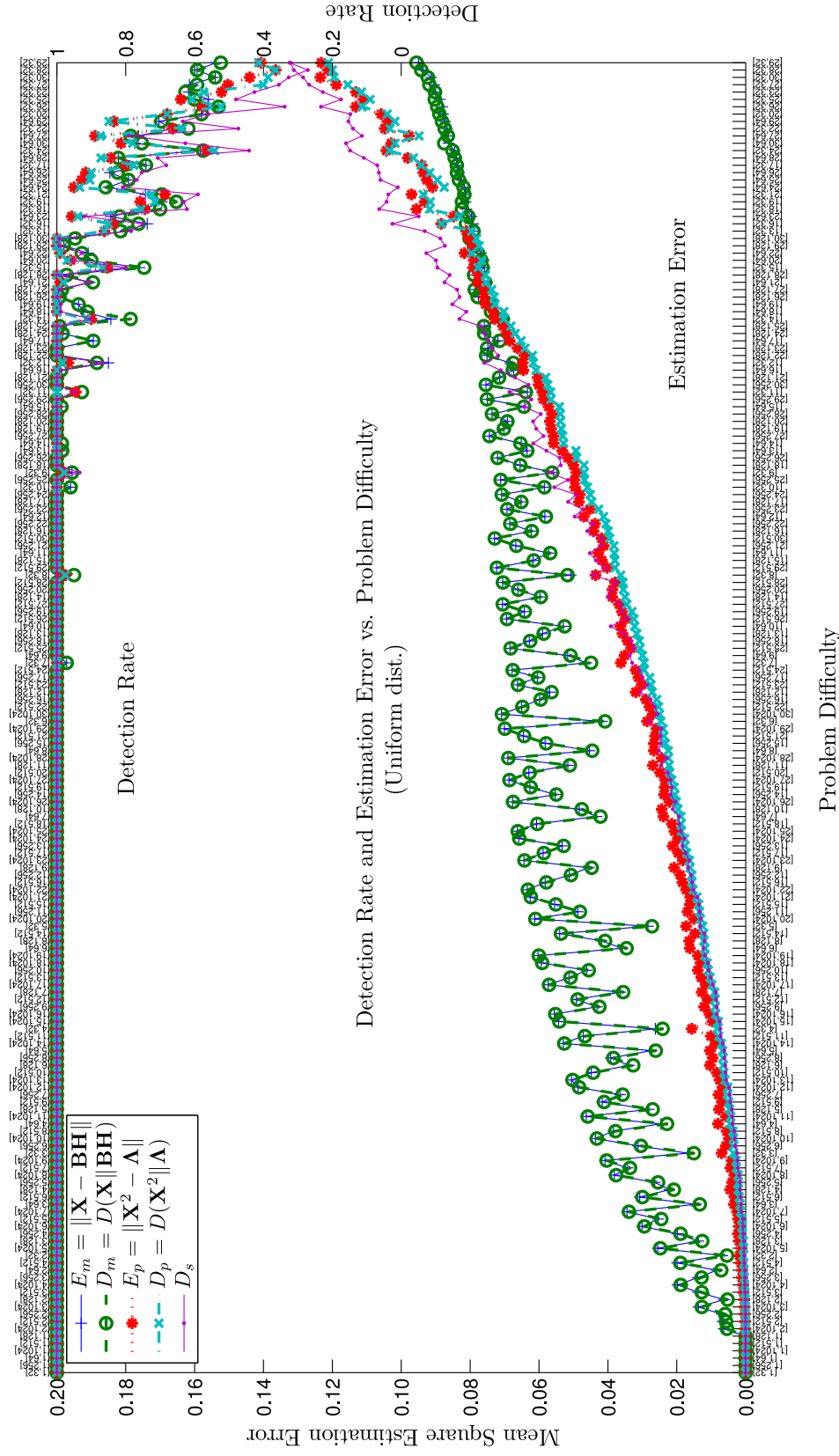


Figure 44: Estimation error and detection rate for components initialized with true **B** drawn from uniform distribution

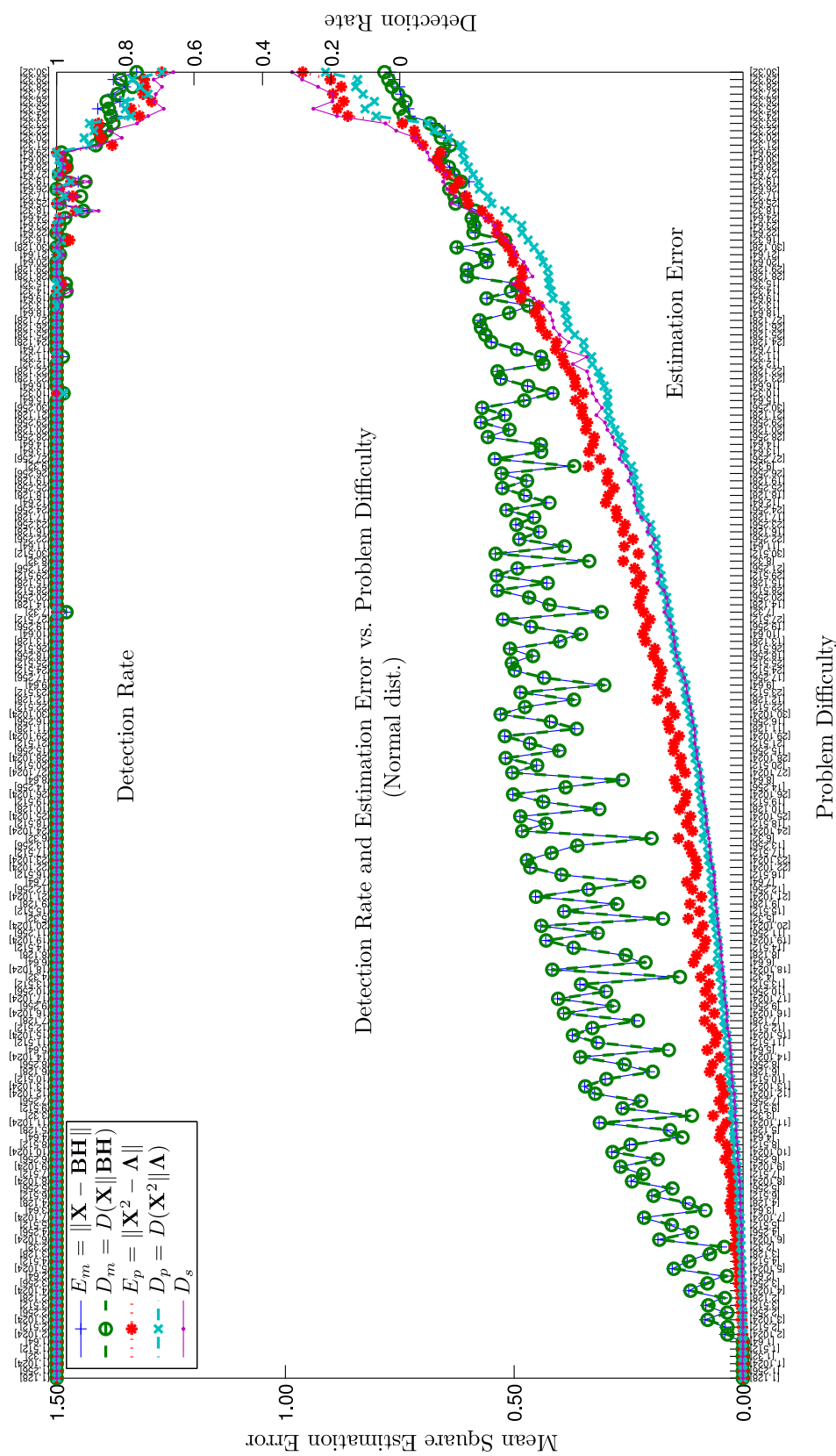


Figure 45: Estimation error and detection rate for components initialized with true **B** drawn from normal distribution

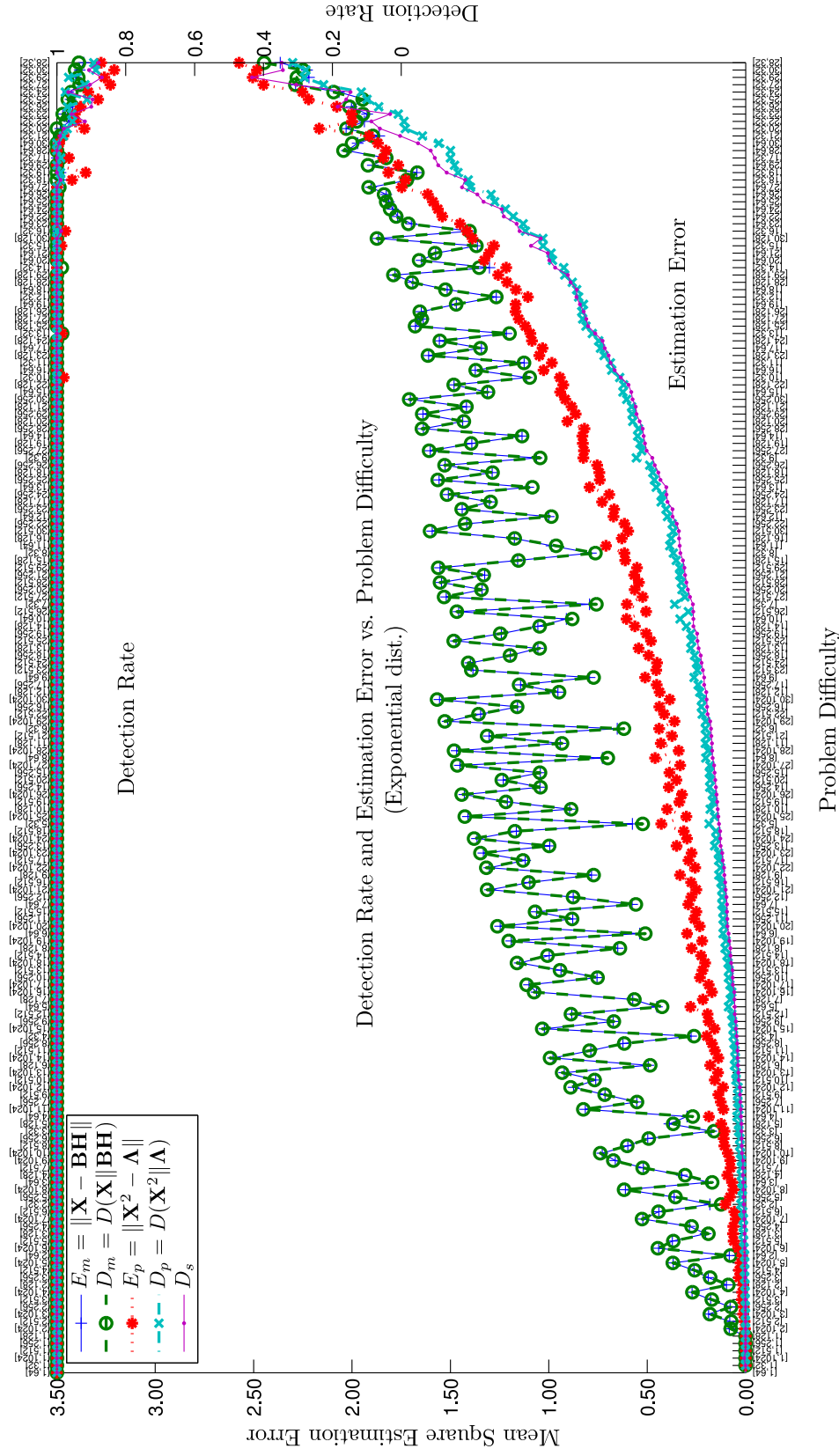


Figure 46: Estimation error and detection rate for components initialized with true **B** drawn from exponential distribution

4.3.3.4 Application to Musical Audio

In this section, we demonstrate the relevance of our approach to the separation of musical audio. We evaluate the five cost functions for the task of separating two overlapping musical notes. We select audio recordings of the same musical note (middle C) for a variety of instruments and playing styles from the Iowa Musical Instrument Samples Database [48]. This represents the most difficult two-component separation task for non-percussive musical instruments.

We select all fortissimo recordings of middle C resulting in 28 one-second audio samples resampled to 22050 Hz. We compute the short-time Fourier transform using an FFT size of 2048 samples, a Hanning window of 1025 samples, and a hop size of 64 samples. We take the magnitude of the complex STFT to attain its magnitude spectrogram. We then approximate each magnitude spectrogram by a rank-one matrix using non-negative matrix factorization with Euclidian distance metric. Each rank-one magnitude spectrogram represents one component in a two component mixture. We use the original phase of the recording to regain the STFT for each component. We construct the mixture magnitude spectrogram by summing two component STFTs and taking the absolute value. This mixture magnitude spectrogram is the input to each of the spectrogram factorization algorithms. We evaluate the success of each algorithm based on the mean square error between the estimated and original rank-one magnitude spectrograms.

Figure 47 and Figure 48 shows the average spectral shape and average amplitude envelope for each of the instrument recordings, respectively. The abbreviations are defined in Table 7. We evaluate each of the algorithms on all pairs of instrument recordings resulting in 378 total trials. Figure 49 shows the relative difficulty of each of the pairings and Figure 50 shows the relative difficulty to separate each instrument (sorted by average mean square error per instrument). Figure 51 shows which cost function had the lowest mean square error for each of pair of instruments. Our proposed cost function, D_s , outperformed the rest on 241 of the trials (64%) and had an average mean square error of 52.3 (36%

better than the second best cost function, D_m , at 82.1). We repeated the experiment using the known spectral shapes for each component and estimating each amplitude envelope. This represents prior knowledge that could be incorporated into the algorithms. Table 8 summarizes the results of both experiments, listing the number of times each algorithm outperforms the rest and each algorithm's average mean square error across all instrument pairs for unknown and known spectral shapes.

4.4 Putting It All Together

In this section we combine our multichannel and phase-aware contributions and apply it to a more complex musical example. First we extend the phase-aware cost function to multiple channels via the same factorization as Section 4.2.1. We minimize the phase-aware cost function between $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{B}}\tilde{\mathbf{A}}\mathbf{H}$:

$$D(\mathbf{1}\|\tilde{\mathbf{X}}^2/\tilde{\mathbf{A}}) = \sum_{mkt} \frac{(\mathbf{X}_m^2)_{kt}}{(\mathbf{A}_m)_{kt}} - 1 + \log \left(\frac{(\mathbf{A}_m)_{kt}}{(\mathbf{X}_m^2)_{kt}} \right), \quad (146)$$

where $\tilde{\mathbf{A}}$ is a stacked version of \mathbf{A} and $(\mathbf{A}_m)_{kt} = \sum_r \mathbf{B}_{kr}^2 \mathbf{A}_{mr}^2 \mathbf{H}_{rt}^2$. The gradient is proportional to the following:

$$\frac{\partial}{\partial(\mathbf{B}_{kr}^2)} D(\mathbf{1}\|\tilde{\mathbf{X}}^2/\tilde{\mathbf{A}}) = \sum_{mt} \mathbf{A}_{mr}^2 \mathbf{H}_{rt}^2 \left(\frac{(\mathbf{A}_m)_{kt} - (\mathbf{X}_m^2)_{kt}}{(\mathbf{A}_m^2)_{kt}} \right) \quad (147)$$

$$\frac{\partial}{\partial(\mathbf{A}_{mr}^2)} D(\mathbf{1}\|\tilde{\mathbf{X}}^2/\tilde{\mathbf{A}}) = \sum_{kt} \mathbf{B}_{kr}^2 \mathbf{H}_{rt}^2 \left(\frac{(\mathbf{A}_m)_{kt} - (\mathbf{X}_m^2)_{kt}}{(\mathbf{A}_m^2)_{kt}} \right) \quad (148)$$

$$\frac{\partial}{\partial(\mathbf{H}_{rt}^2)} D(\mathbf{1}\|\tilde{\mathbf{X}}^2/\tilde{\mathbf{A}}) = \sum_{mk} \mathbf{B}_{kr}^2 \mathbf{A}_{mr}^2 \left(\frac{(\mathbf{A}_m)_{kt} - (\mathbf{X}_m^2)_{kt}}{(\mathbf{A}_m^2)_{kt}} \right) \quad (149)$$

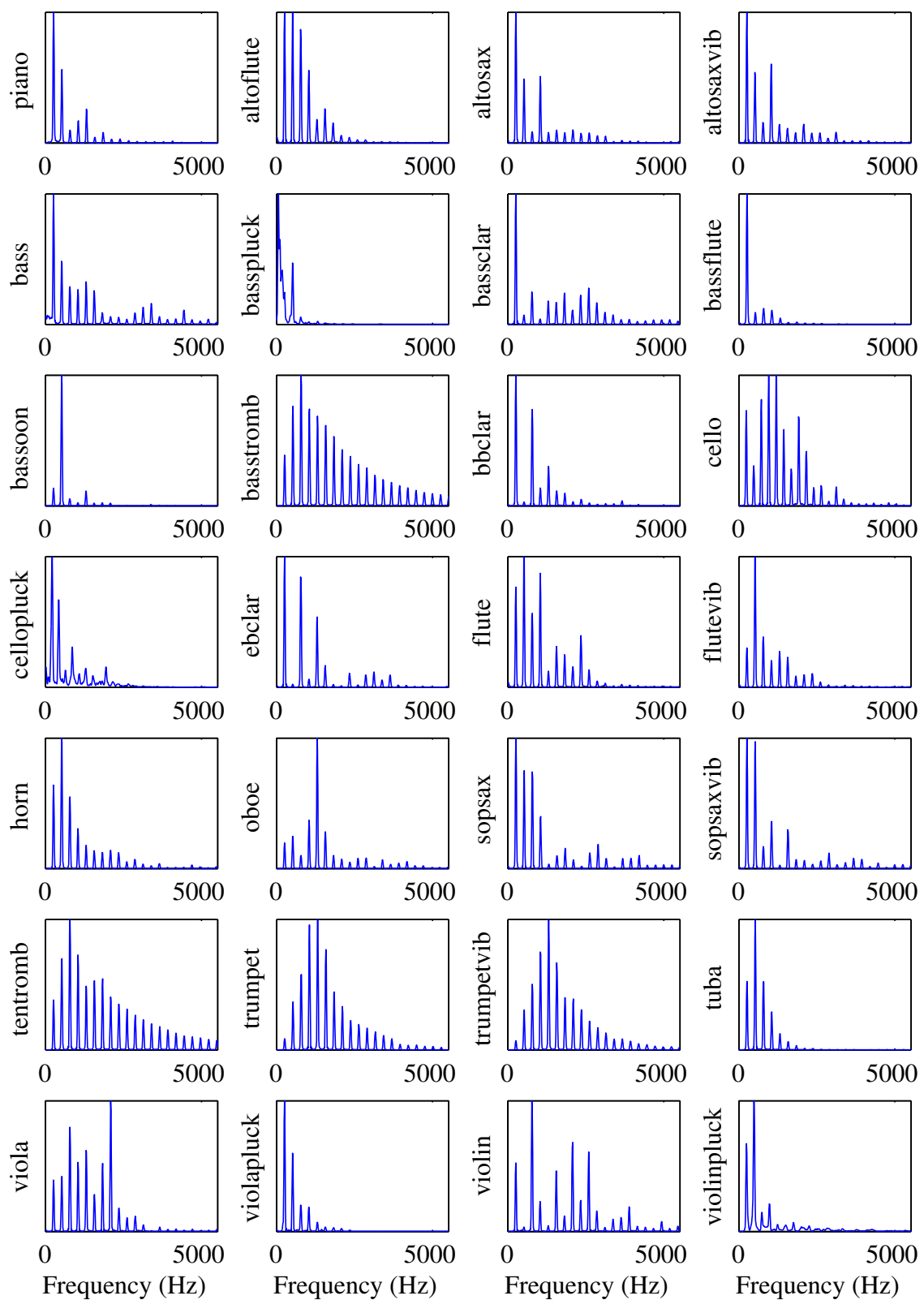


Figure 47: The spectral shape of each of the 28 instrument recordings of middle C

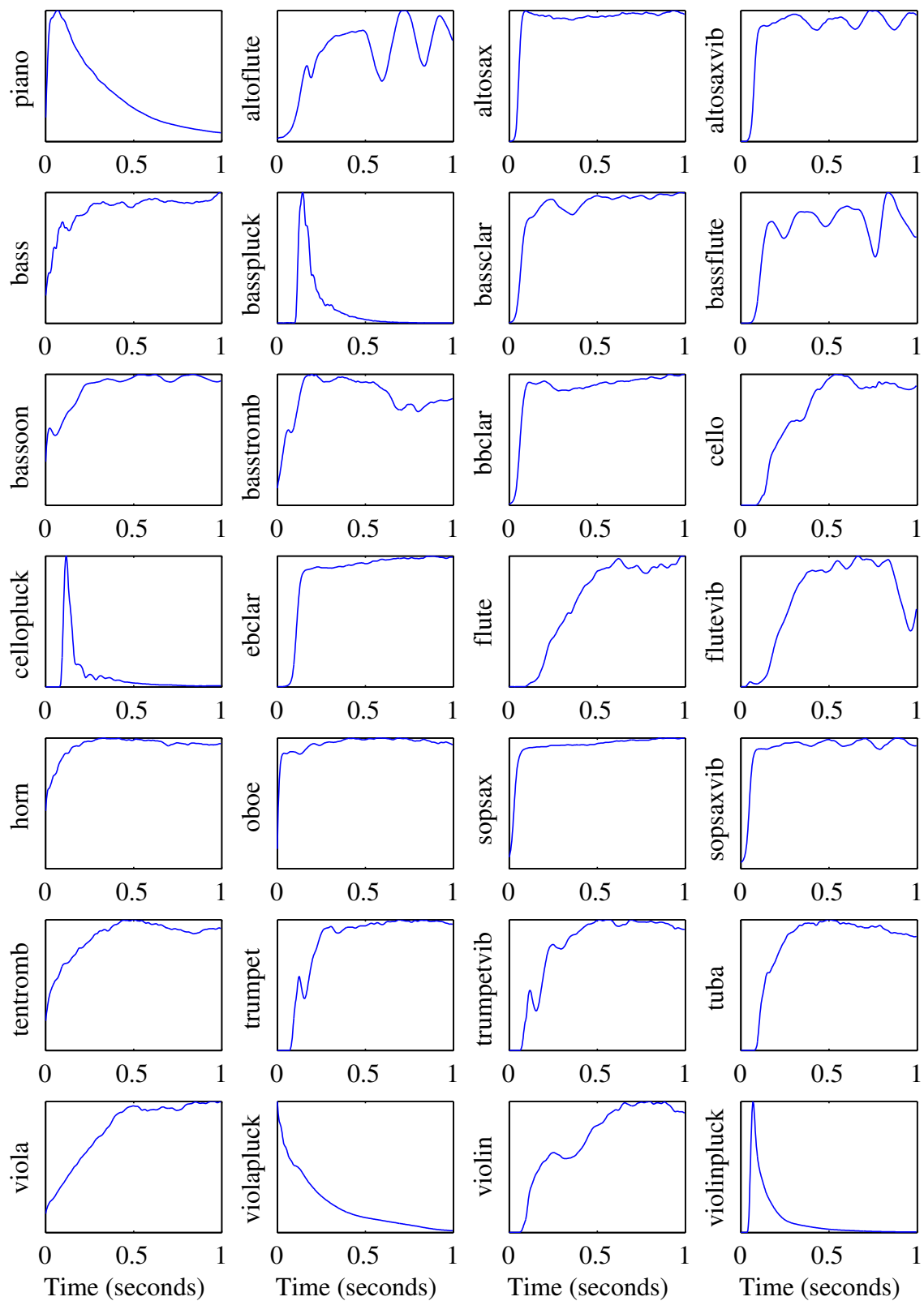


Figure 48: The amplitude envelope of each of the 28 instrument recordings of middle C

Table 7: Instrument recording abbreviation definitions

Abbreviation	Description	Filename (from [48])
piano	Piano	Piano.ff.C4.aiff
altoflute	Alto Flute	AltoFlute.ff.C4B4.aiff
altosax	Alto Saxophone	AltoSax.NoVib.ff.C4B4.aiff
altosaxvib	Alto Saxophone with Vibrato	AltoSax.Vib.ff.C4B4.aiff
bass	Double Bass (Bowed)	Bass.arco.ff.sulG.C4G4.aiff
basspluck	Double Bass (Plucked)	Bass.pizz.ff.sulG.C4G4.aiff
bassclar	Bass Clarinet	BassClarinet.ff.C4B4.aiff
bassflute	Bass Flute	BassFlute.ff.C4B4.aiff
bassoon	Bassoon	Bassoon.ff.C4B4.aiff
basstromb	Bass Trombone	BassTrombone.ff.C4F4.aiff
bbclar	B-flat Clarinet	BbClar.ff.C4B4.aiff
cello	Cello (Bowed)	Cello.arco.ff.sulG.C4B4.aiff
cellopluck	Cello (Plucked)	Cello.pizz.ff.sulG.C4B4.aiff
ebclar	E-flat Clarinet	EbClar.ff.C4B4.aiff
flute	Flute	flute.novib.ff.B3B4.aiff
flutevib	Flute with Pitch Modulation	flute.vib.ff.B3B4.aiff
horn	French Horn	Horn.ff.C4B4.aiff
oboe	Oboe	oboe.ff.C4B4.aiff
sopsax	Soprano Saxophone	SopSax.NoVib.ff.C4B4.aiff
sopsaxvib	Soprano Saxophone with Vibrato	SopSax.Vib.ff.C4B4.aiff
tentromb	Tenor Trombone	TenorTrombone.ff.C4B4.aiff
trumpet	Trumpet	Trumpet.novib.ff.C4B4.aiff
trumpetvib	Trumpet with Vibrato	Trumpet.vib.ff.C4B4.aiff
tuba	Tuba	Tuba.ff.C3C4.aiff
viola	Viola (Bowed)	Viola.arco.sulG.mf.C4B4.aiff
violapluck	Viola (Plucked)	Viola.pizz.sulG.ff.C4B4.aiff
violin	Violin (Bowed)	Violin.arco.ff.sulG.C4B4.aiff
violinpluck	Violin (Plucked)	Violin.pizz.ff.sulG.C4B4.aiff

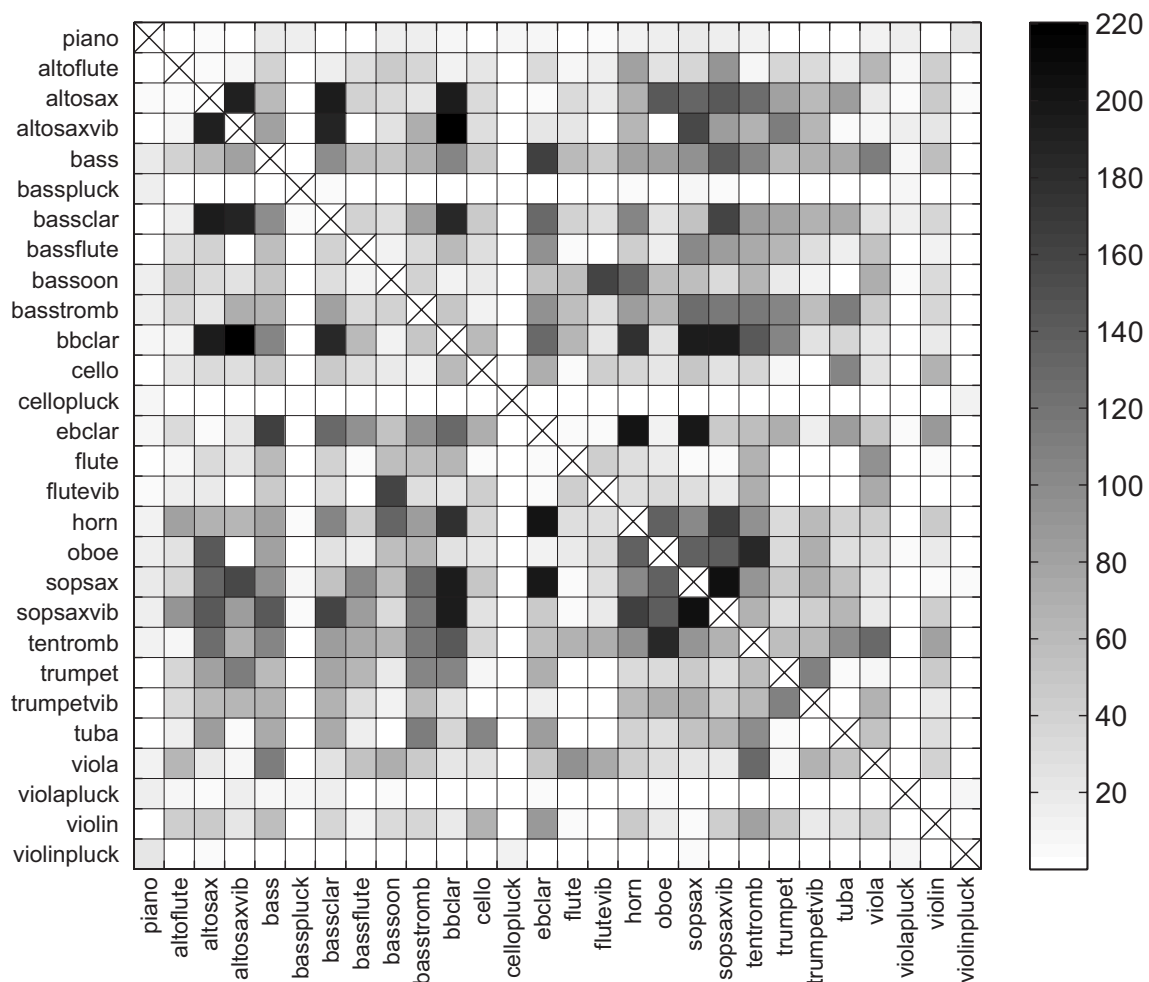


Figure 49: The lowest mean square error for each pair of instrument recordings

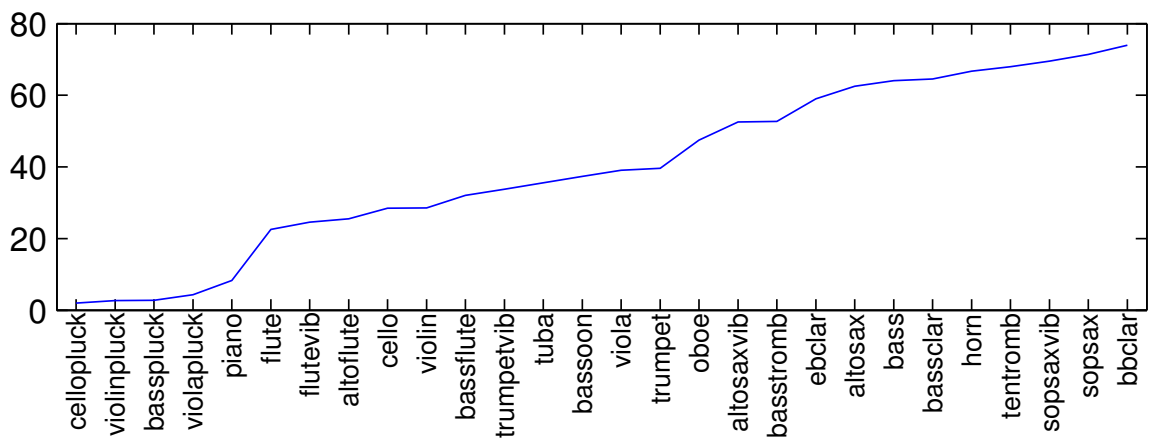


Figure 50: The relative difficulty of separating each instrument

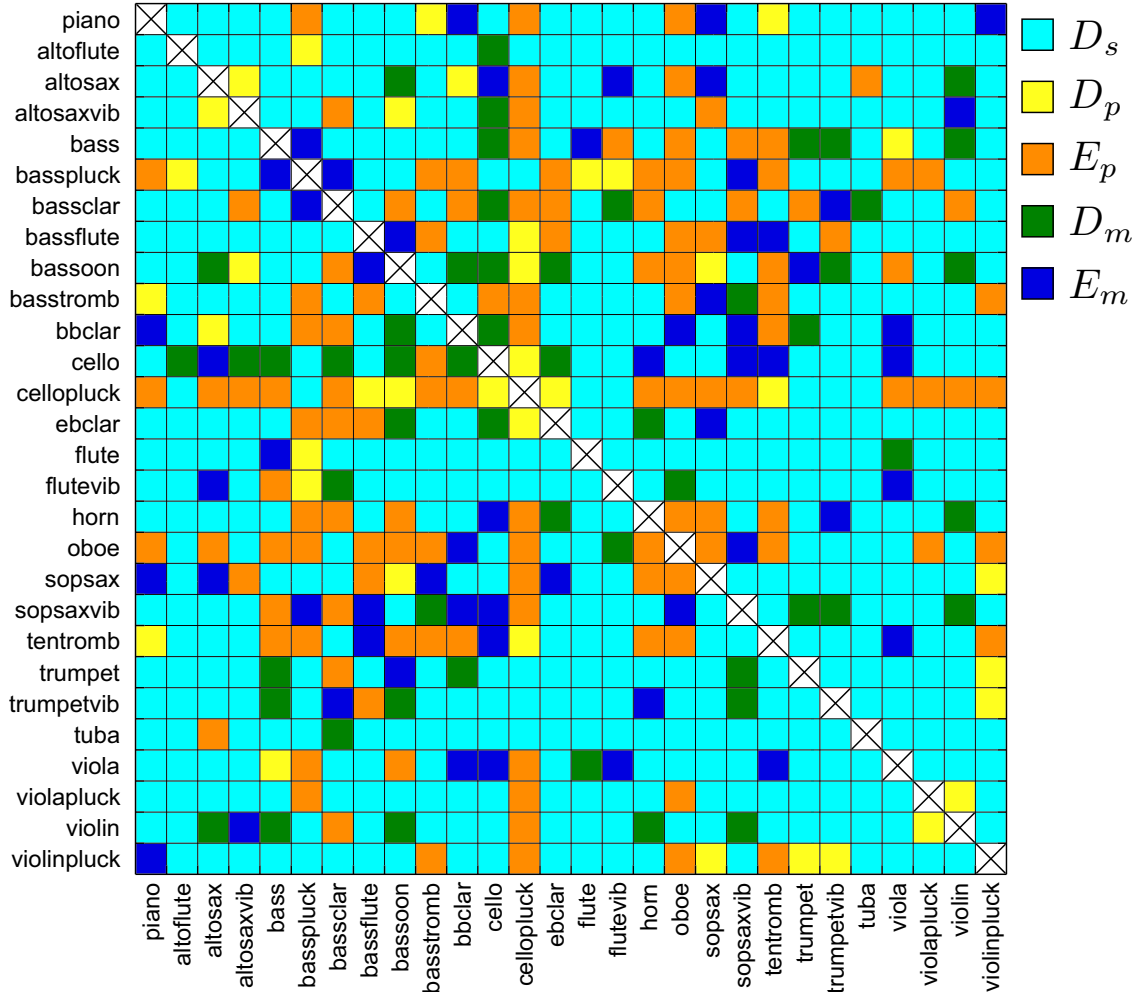


Figure 51: The cost function that makes the minimum mean square error estimate for each pair of instruments

Table 8: Cost function performance for separating pairs of instrument components

Experiment	Evaluation	E_m	D_m	E_p	D_p	D_s
Unknown	Best	29	27	62	19	241
Spectral Shape	MSE<15	87	78	87	92	146
	MSE	83.4	82.1	89.4	89.3	52.3
Known	Best	0	0	7	4	367
Spectral Shape	MSE<15	138	163	196	249	377
	MSE	33.6	21.6	31.8	15.0	0.522

We choose the following learning rates:

$$\beta_{kr} = \mathbf{B}_{kr}^2 / \sum_{mt} \mathbf{A}_{mr}^2 \mathbf{H}_{rt}^2 / (\mathbf{\Lambda}_m)_{kt} \quad (150)$$

$$\alpha_{mr} = \mathbf{A}_{mr}^2 / \sum_{kt} \mathbf{B}_{kr}^2 \mathbf{H}_{rt}^2 / (\mathbf{\Lambda}_m)_{kt} \quad (151)$$

$$\eta_{rt} = \mathbf{H}_{rt}^2 / \sum_{mk} \mathbf{B}_{kr}^2 \mathbf{A}_{mr}^2 / (\mathbf{\Lambda}_m)_{kt} , \quad (152)$$

and derive the following multiplicative updates:

$$\mathbf{B}_{kr}^2 \leftarrow \mathbf{B}_{kr}^2 \frac{\sum_{mt} \mathbf{A}_{mr}^2 \mathbf{H}_{rt}^2 (\mathbf{X}_m^2)_{kt} / (\mathbf{\Lambda}_m)_{kt}^2}{\sum_{mt} \mathbf{A}_{mr}^2 \mathbf{H}_{rt}^2 / (\mathbf{\Lambda}_m)_{kt}} \quad (153)$$

$$\mathbf{A}_{mr}^2 \leftarrow \mathbf{A}_{mr}^2 \frac{\sum_{kt} \mathbf{B}_{kr}^2 \mathbf{H}_{rt}^2 (\mathbf{X}_m^2)_{kt} / (\mathbf{\Lambda}_m)_{kt}^2}{\sum_{kt} \mathbf{B}_{kr}^2 \mathbf{H}_{rt}^2 / (\mathbf{\Lambda}_m)_{kt}} \quad (154)$$

$$\mathbf{H}_{rt}^2 \leftarrow \mathbf{H}_{rt}^2 \frac{\sum_{mk} \mathbf{B}_{kr}^2 \mathbf{A}_{mr}^2 (\mathbf{X}_m^2)_{kt} / (\mathbf{\Lambda}_m)_{kt}^2}{\sum_{mk} \mathbf{B}_{kr}^2 \mathbf{A}_{mr}^2 / (\mathbf{\Lambda}_m)_{kt}} . \quad (155)$$

4.4.1 Application to Musical Audio

We construct musical mixtures containing three instruments playing the same two notes. All three instruments play middle C followed immediately by middle E. Because the tuba does not have a middle E, we remove it from the set of instruments and focus on the remaining 27. We randomly select combinations of three instruments from this set and introduce the spatial mixing matrix, \mathbf{A} , panning the instruments to the left, center, and right:

$$\mathbf{A} = \begin{bmatrix} 0.8944 & 0.7071 & 0.4472 \\ 0.4472 & 0.7071 & 0.8944 \end{bmatrix} . \quad (156)$$

Figures 52-70 show the results for one trial with a mixture of bass, flute, and soprano saxophone. The advantage of D_s is most obvious on the estimates of amplitude envelopes (Figures 54, 54, 60, 61, 66, and 67). With the exception of middle E on flute (Figure 61), the amplitudes estimated by D_s are clearly very similar to the original. The most difficult instrument to separate from this mixture was the flute. In spite of the problem D_s had in

Table 9: Cost function performance for separating bass, flute, and soprano saxophone from two mixtures

Cost Functions:	E_m	D_m	E_p	D_p	D_s
MSE	47.4	52.5	53.7	48.9	12.9

estimating the amplitude envelope, it clearly provided a better estimate than the rest. In fact, D_m made a detection error by estimating middle E during the wrong time interval. The suitability of the D_s cost function is most prevalent in the estimation of the soprano saxophone. D_s clearly succeeds in estimating the spectral shapes *and* amplitude envelopes, whereas the other cost functions clearly fail. For the other instruments, D_s clearly provides better estimates for each component than the other cost functions. In addition, the spatial positions (Figure 70) estimated by D_s more closely match the true positions and provide a means for clustering in the spatial domain. These observations coincide with the mean square error computed for each cost function (Table 9).

Table 10 summarizes the results over 65 trials. Our phase-aware cost function (D_s) performed better than the other cost functions on 72% of the trials. By listening to an assortment of the trials, we determined that a mean square error of less than approximately 15 produced audio files that were nearly indistinguishable from the original components. Therefore we use a threshold of 15 to determine the success rate. D_s was successful more often than the other cost functions (63% of the trials). Over all the trials, D_s had a much better median mean square error but a markedly worse average mean square error. This is due to three outliers where D_s performed poorly. Removing these outliers reduced the average MSE of the other cost functions by 1 or 2 and reduces the D_s average MSE to 15.6.

Table 10: Cost function performance for separating three instruments from two mixtures

Cost Functions:	E_m	D_m	E_p	D_p	D_s
Best	5	7	1	5	47
MSE < 15	16	16	4	17	41
Average MSE	28.8	30.8	40.8	32.7	59.7
Median MSE	30.2	34.0	42.6	35.7	9.0

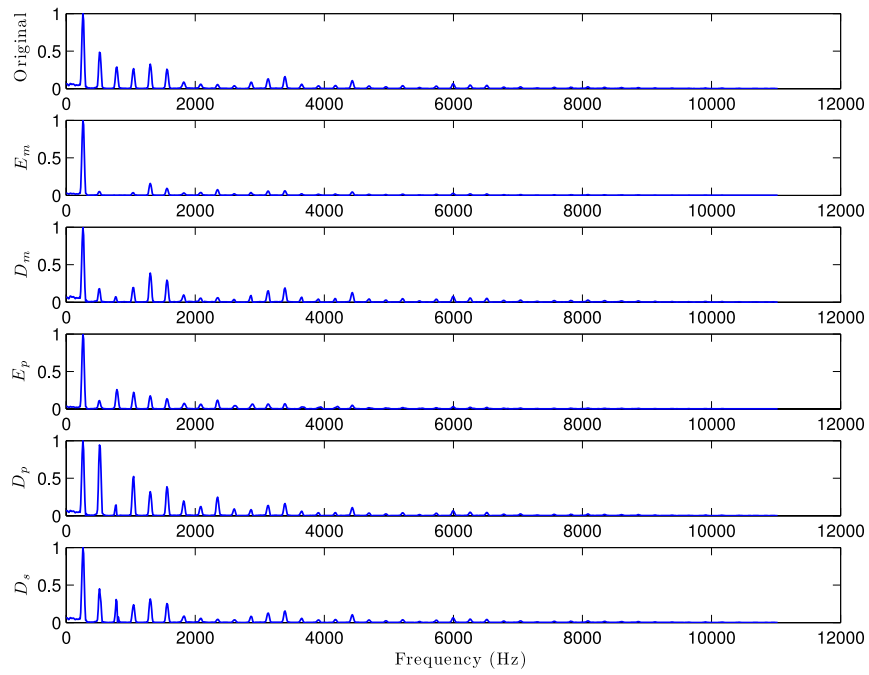


Figure 52: Original (*top*) and estimated spectral shapes for middle C on bass

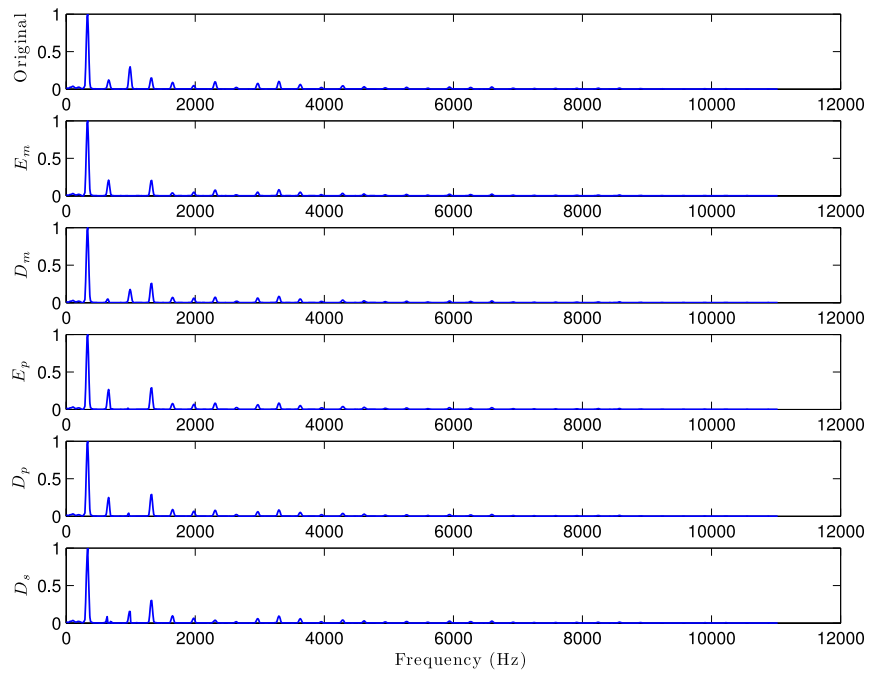


Figure 53: Original (*top*) and estimated spectral shapes for middle E on bass

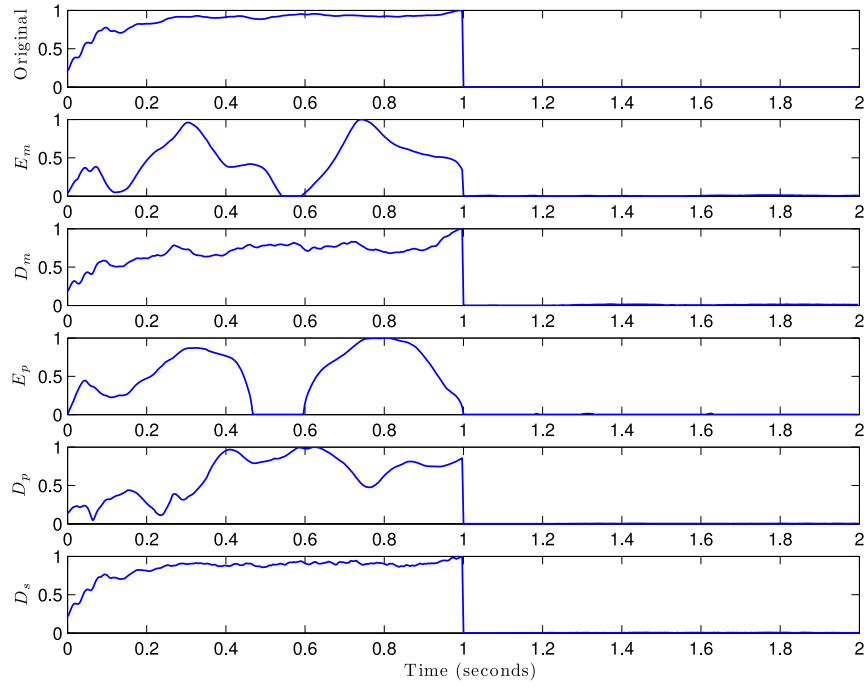


Figure 54: Original (*top*) and estimated amplitude envelopes for middle C on bass

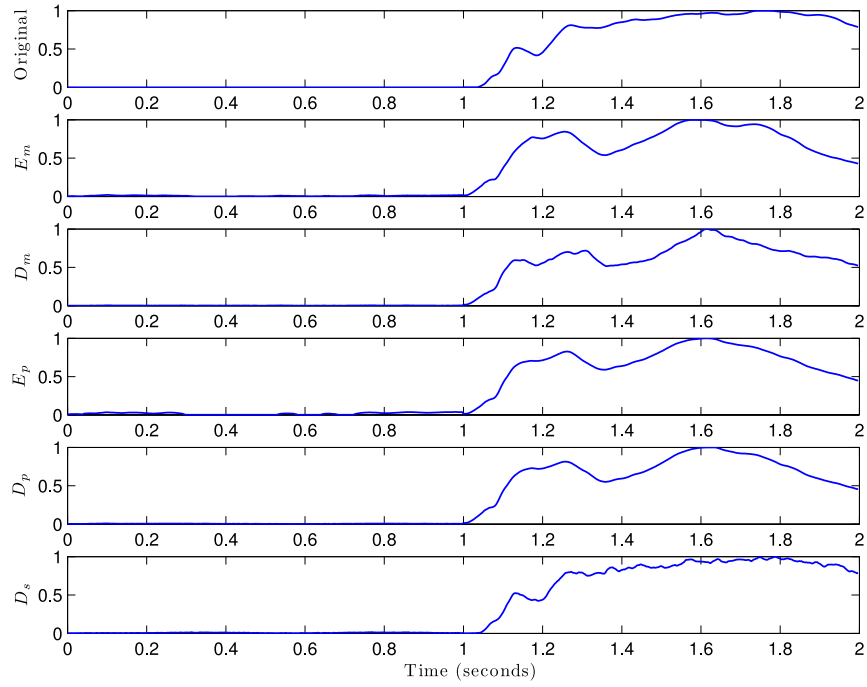


Figure 55: Original (*top*) and estimated amplitude envelopes for middle E on bass

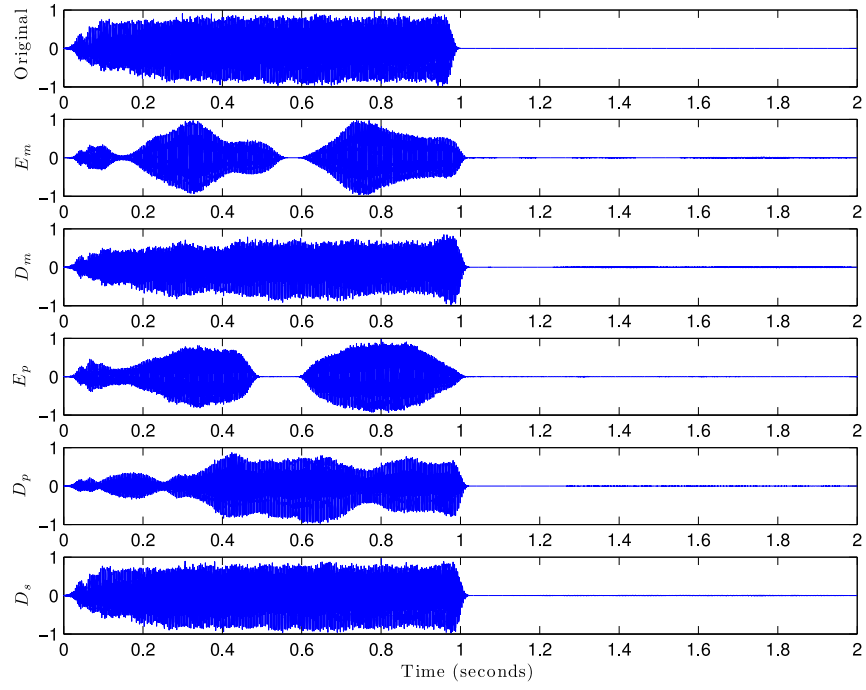


Figure 56: Original (*top*) and estimated audio signals for middle C on bass

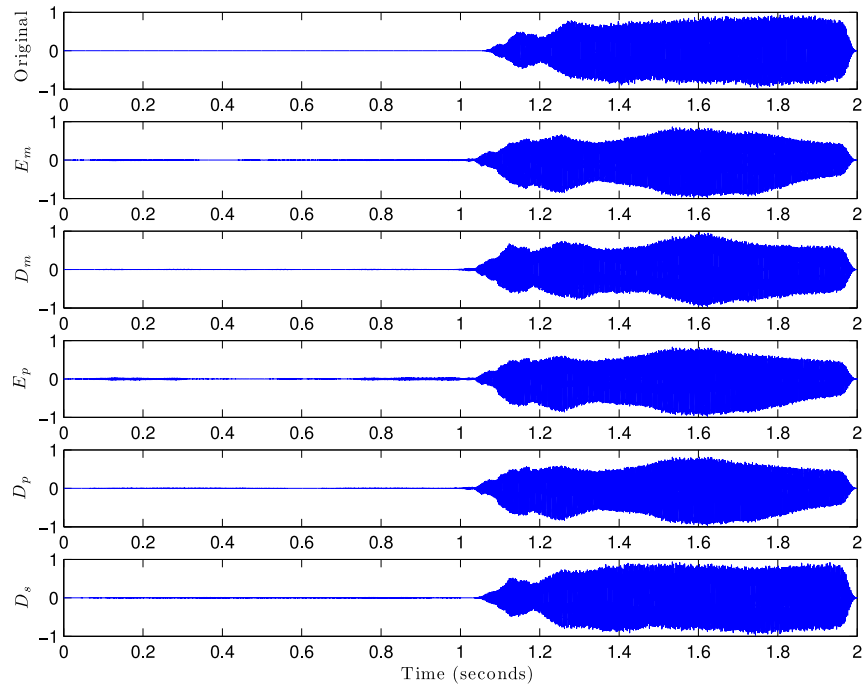


Figure 57: Original (*top*) and estimated audio signals for middle E on bass

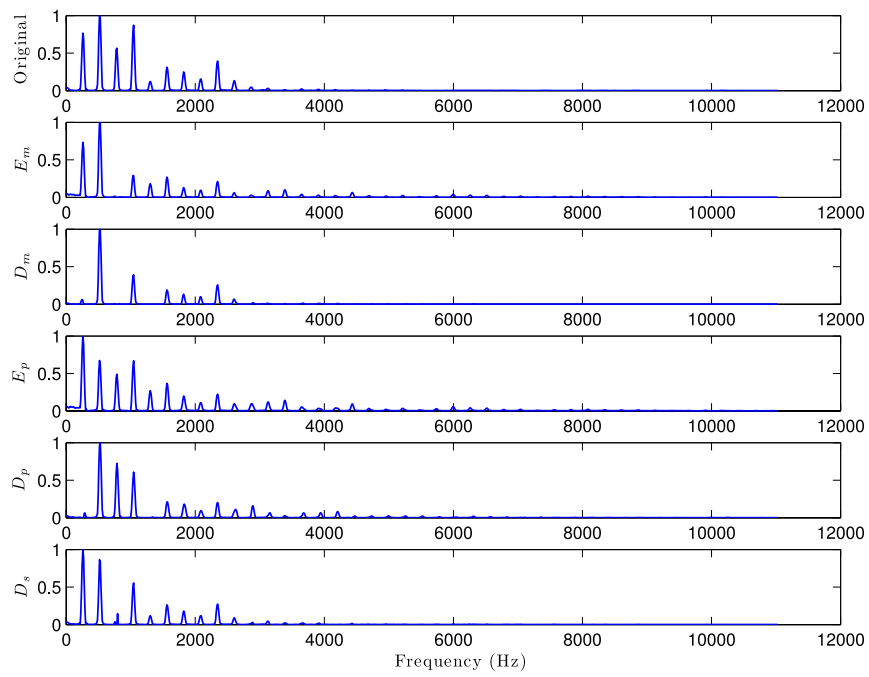


Figure 58: Original (*top*) and estimated spectral shapes for middle C on flute

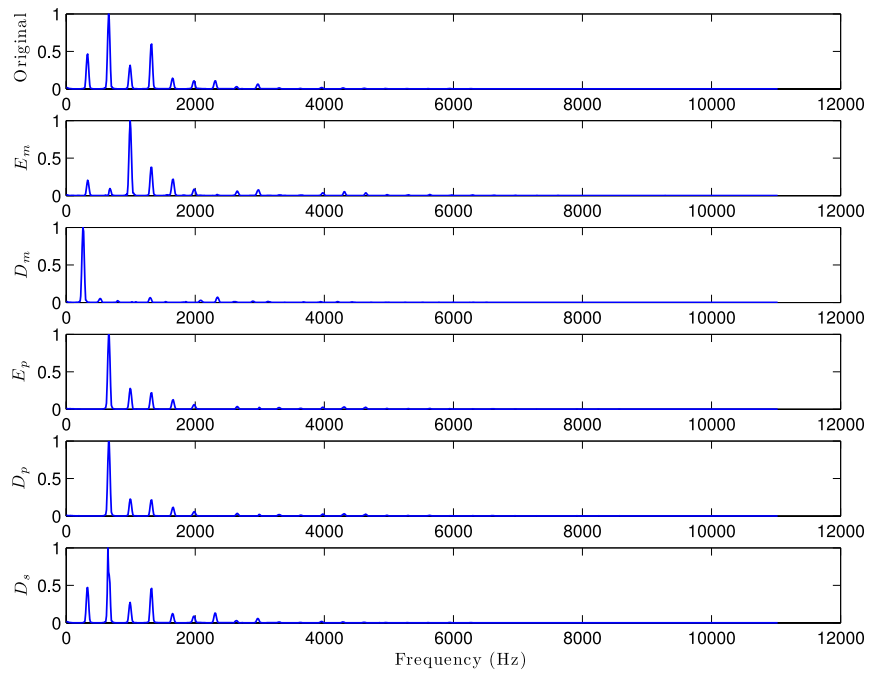


Figure 59: Original (*top*) and estimated spectral shapes for middle E on flute

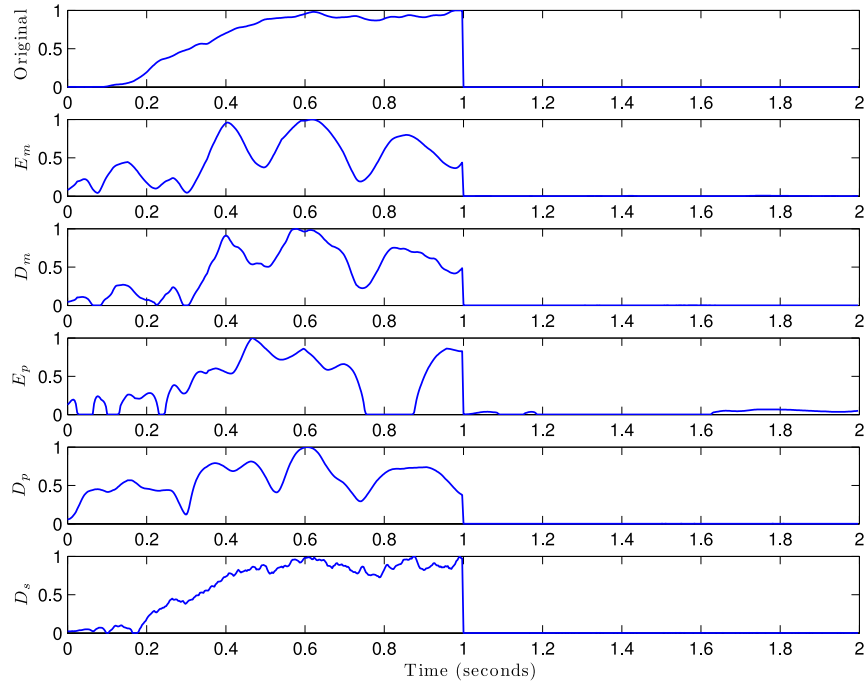


Figure 60: Original (*top*) and estimated amplitude envelopes for middle C on flute

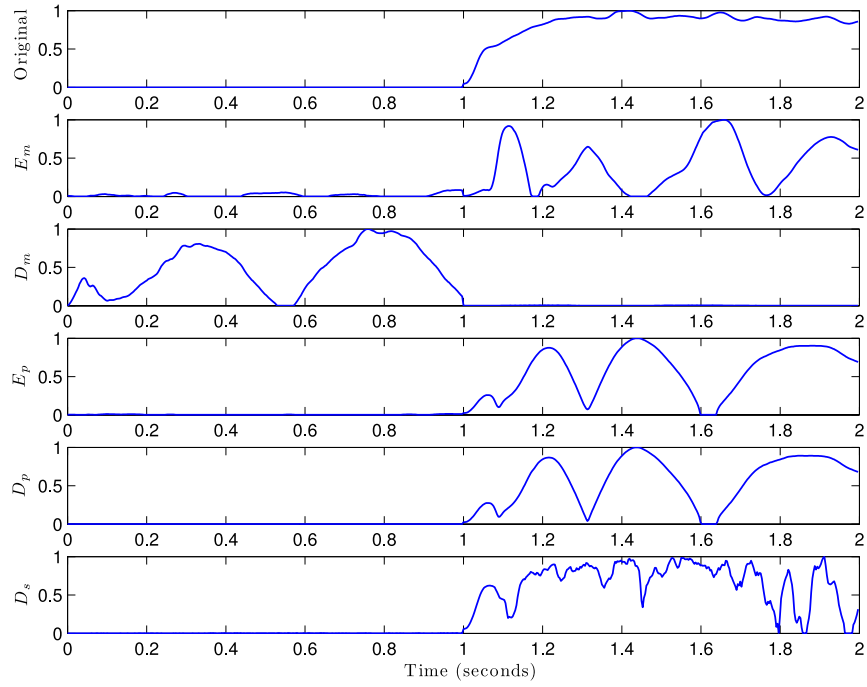


Figure 61: Original (*top*) and estimated amplitude envelopes for middle E on flute

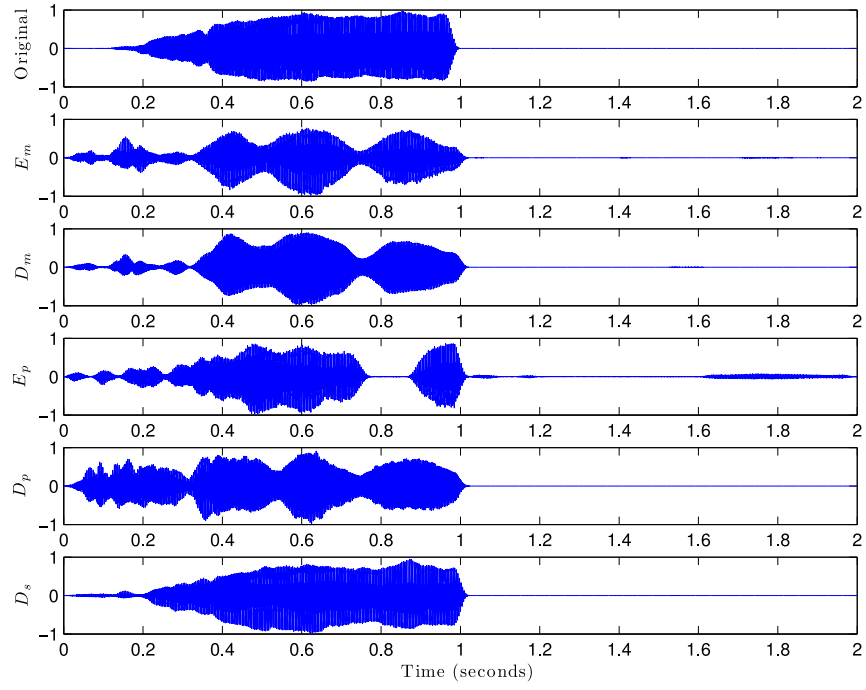


Figure 62: Original (*top*) and estimated audio signals for middle C on flute

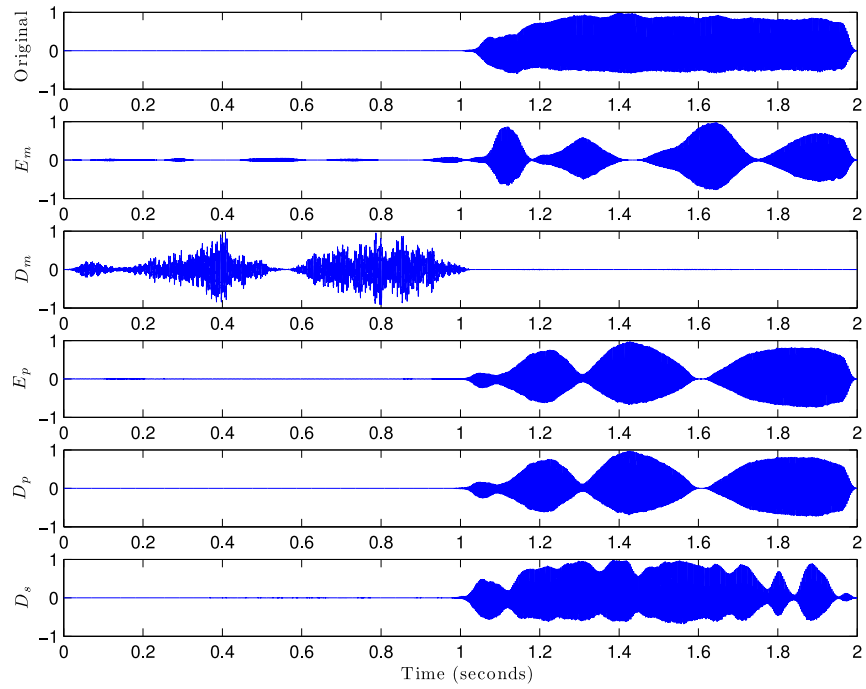


Figure 63: Original (*top*) and estimated audio signals for middle E on flute

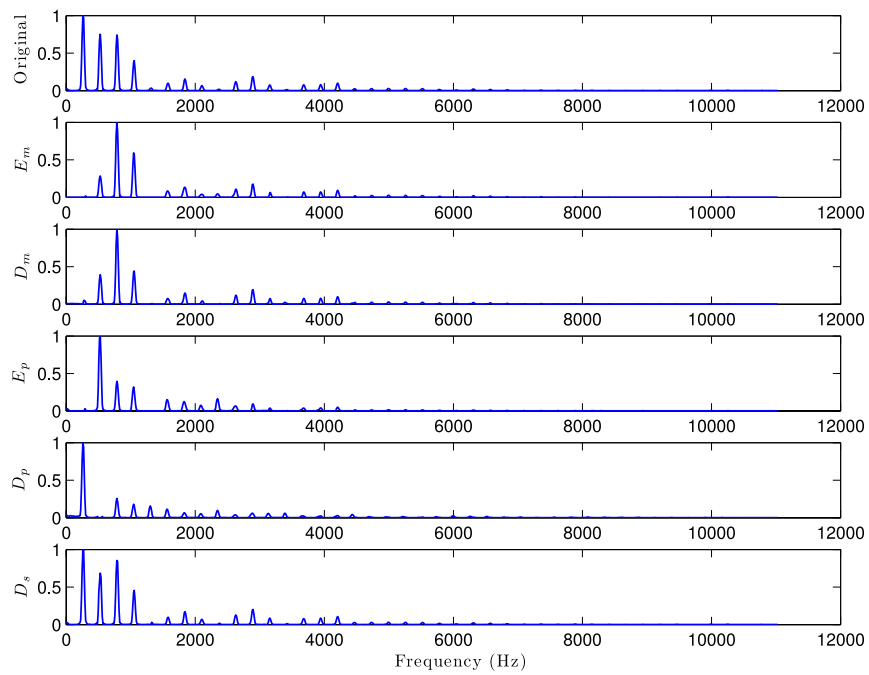


Figure 64: Original (*top*) and estimated spectral shapes for middle C on soprano saxophone

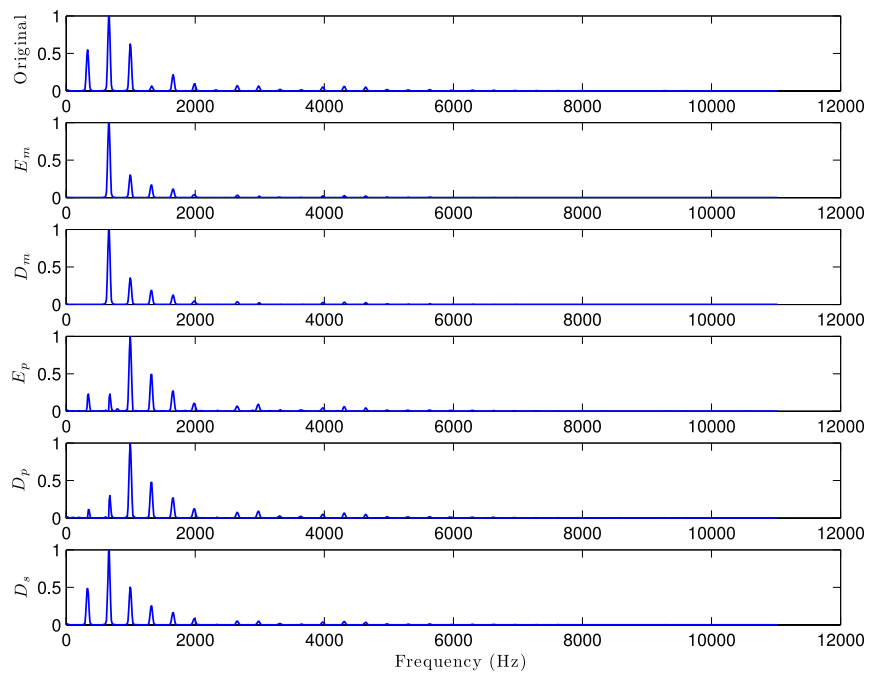


Figure 65: Original (*top*) and estimated spectral shapes for middle E on soprano saxophone

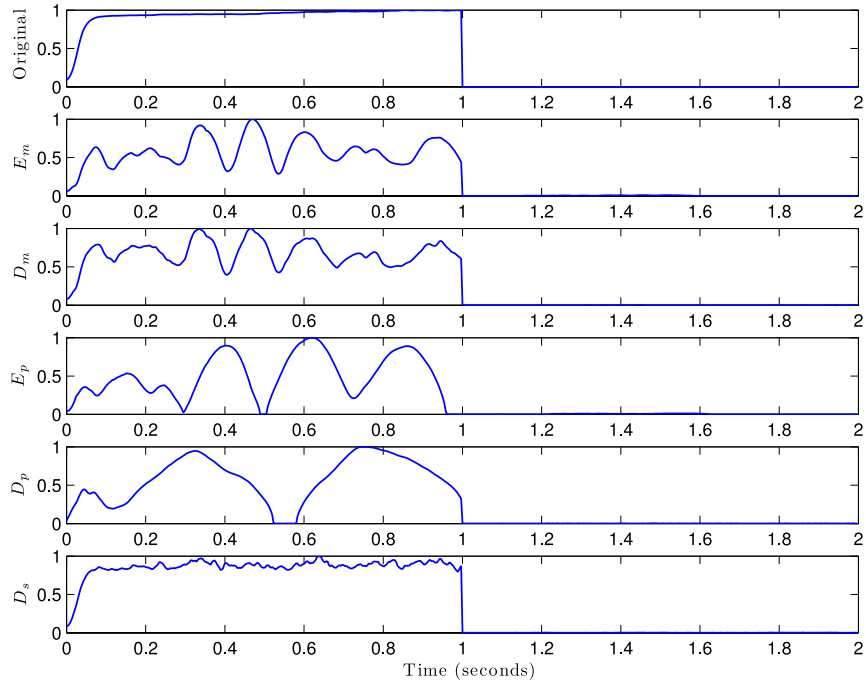


Figure 66: Original (*top*) and estimated amplitude envelopes for middle C on soprano saxophone

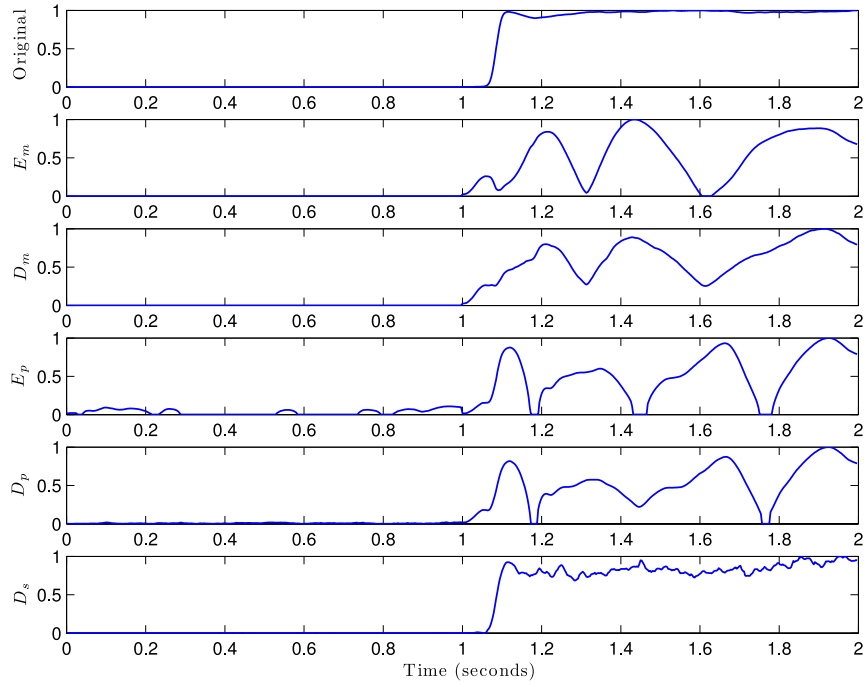


Figure 67: Original (*top*) and estimated amplitude envelopes for middle E on soprano saxophone

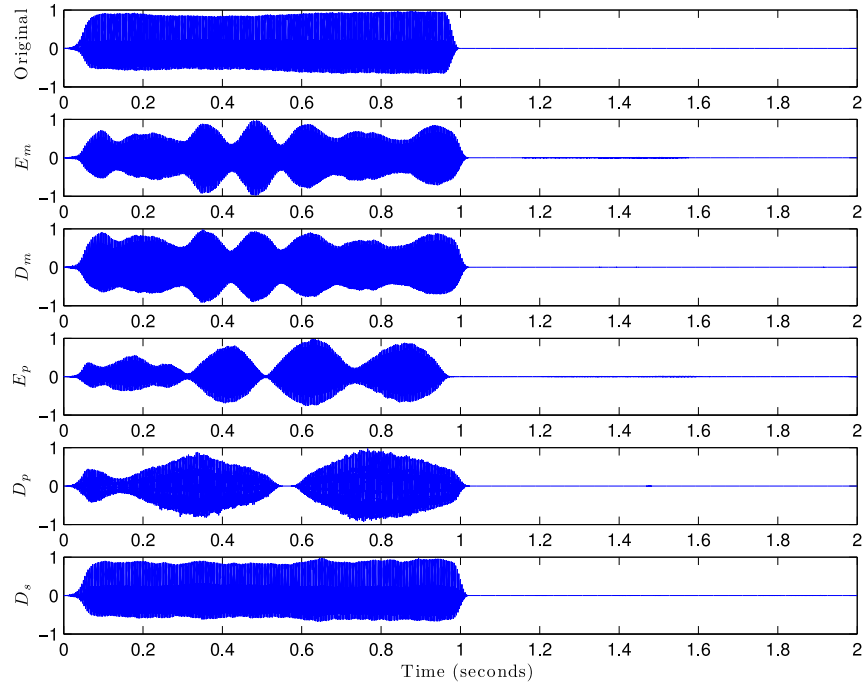


Figure 68: Original (*top*) and estimated audio signals for middle C on soprano saxophone

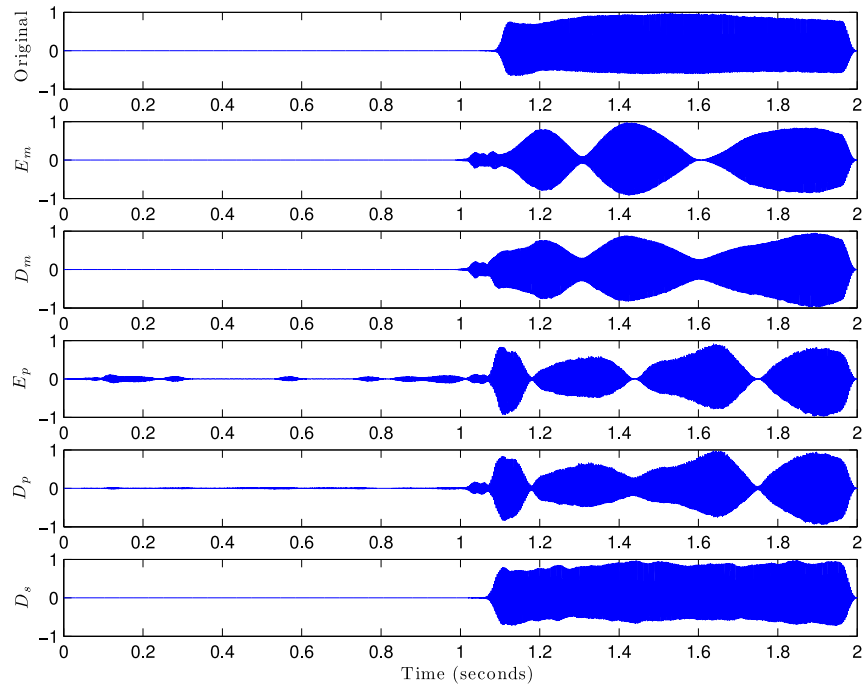


Figure 69: Original (*top*) and estimated audio signals for middle E on soprano saxophone

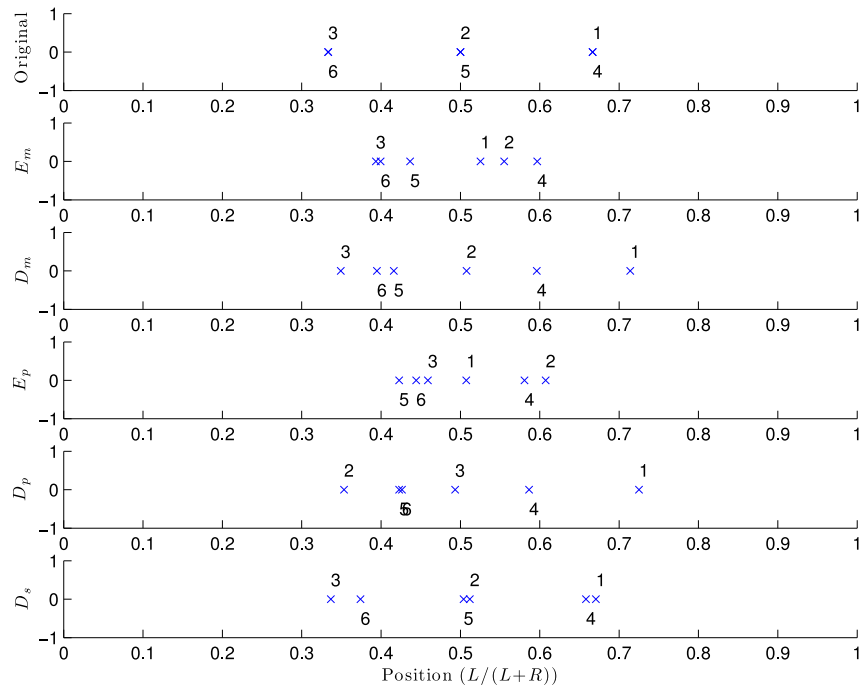


Figure 70: Original (*top*) and estimated positions for all six components

4.5 Application to Real Sounds

In the previous sections, we evaluate our approach using mixtures of rank-one components. Because we know the rank-one components, we can compare each approach quantitatively using mean square error. In addition, each algorithm is parameterized by the (known) number of components, and we produce audio examples using the original phase of the notes. In a real source separation application, we do not know the number of components or the phase. Choosing these parameters is an open problem and will not be addressed in this thesis. Instead, we provide a visual analysis of examples containing real instrument recordings (*i.e.*, not rank-one components). One consequence of using real instrument recordings is that musical notes are not strictly harmonic. The onset of notes (particularly plucked or percussive instruments) typically contains a noisy transient component. This noise does not fit the spectral component model that we have employed and generally complicates the task.

4.5.1 Bass and Organ Example

We begin by mixing two tracks from a song recorded in a studio. The two tracks correspond to an electric bass guitar and an electric organ. During the 5 second segment, the bass guitar repeats two notes and the organ plays one long note. Figure 71 and 72 show the high-energy areas in the magnitude spectrogram for the electric bass and electric organ. All time-frequency points that exceed a magnitude threshold in the mixture spectrogram are considered. If a source contributes at least $1/3$ of the mixture magnitude to a time-frequency point, it is considered to be “active” at that point. If both sources contribute at least $1/3$, we consider the sources to be overlapping at that point. Figure 73 shows the overlap between the two tracks. Both bass notes and the organ note have harmonic energy content near 175 Hz. However, the vast majority of time-frequency points contain energy dominated by one source.

We mixed the two tracks using a mixing matrix that panned the electric bass to the left

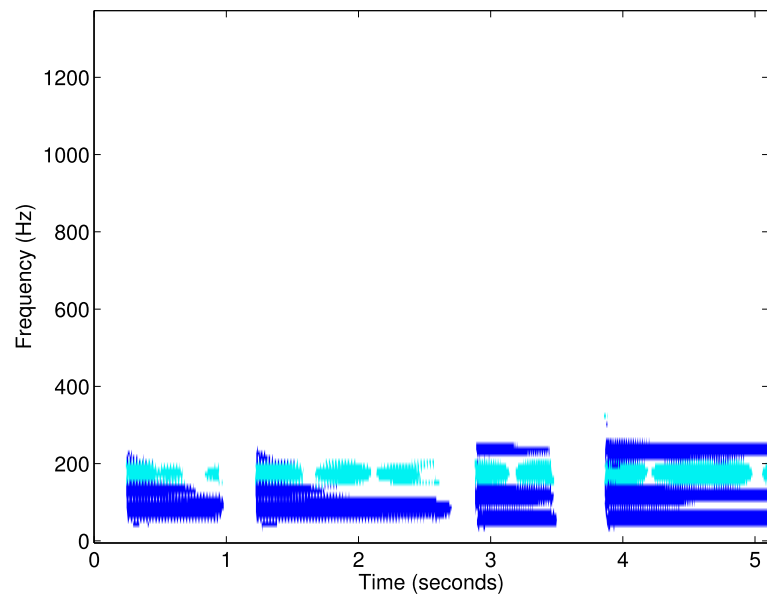


Figure 71: High energy points in the electric bass spectrogram

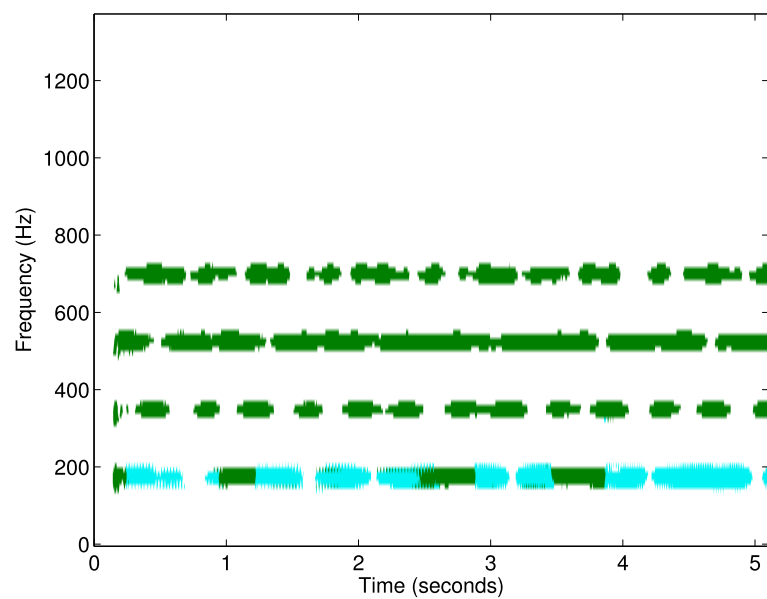


Figure 72: High energy points in the electric organ spectrogram

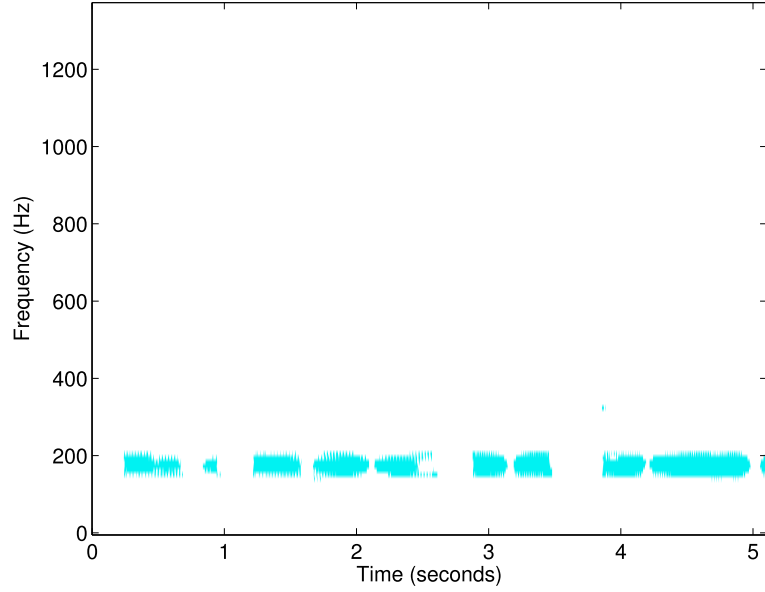


Figure 73: Overlapping high energy in the electric bass and electric organ

and the organ to the right:

$$A = \begin{bmatrix} 0.4472 & 0.8944 \\ 0.8944 & 0.4472 \end{bmatrix}. \quad (157)$$

First, we extract three components (one for each note) using the different cost functions. Figures 74-78 show the three extracted components for each cost function. All four traditional methods correctly identify the three different notes and provide clear spatial clusters near the true instrument locations. The spectral shapes and amplitude envelopes reveal that each component primarily represents one of the bass notes but also contributes to the other note. This is a consequence of using real instrument recordings and that each note cannot be captured by a single rank-one spectrogram. On the other hand, D_s combines both bass notes into one component and breaks the organ note into two components. In addition, the second component captures part of the initial bass note. The lower frequency harmonics in the second component are held roughly constant after the initial onset that coincides with the first bass note. Because so much of the energy is concentrated in a single amplitude peak, its spectral shape captures much of the mixture spectra at that point in time. (Notice

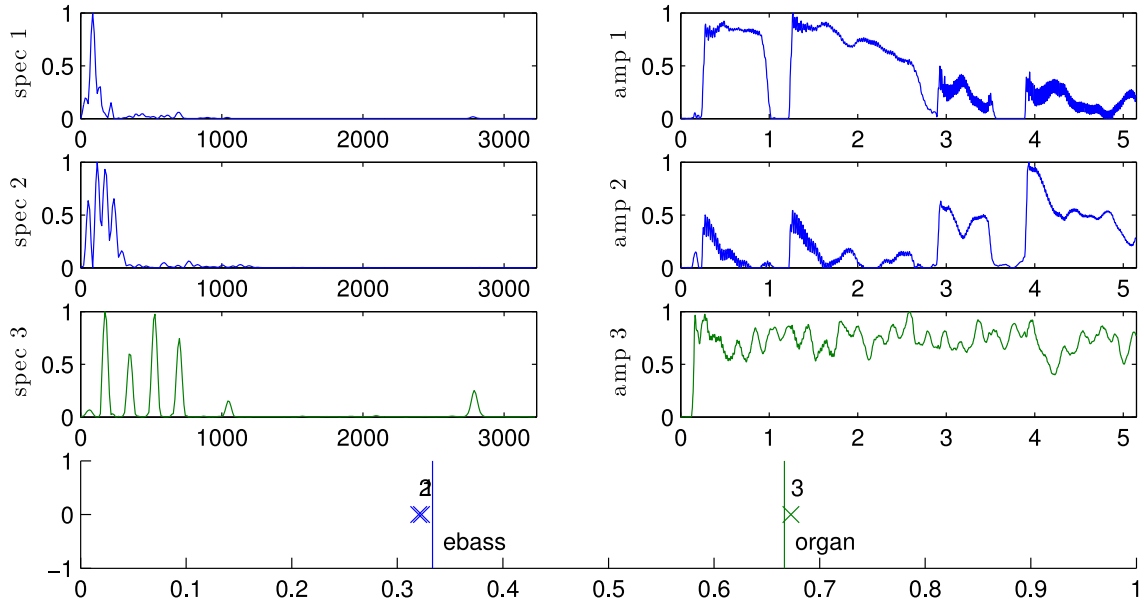


Figure 74: Three components extracted by E_m for bass and organ

the lack of amplitude in the other components at that time.) The third component represents the variation in the high-frequency peak in the organ. Visually, D_s provides the least intuitive decomposition and is most affected by the real recording.

If only one source is active at each time-frequency point, the linear mixing model is valid, *i.e.*, $\mathbf{X} = \sum \mathbf{C}_r$ and $\mathbf{X}^2 = \sum \mathbf{C}_r^2$. For this example, the majority of time-frequency points adhere to this model because there is little overlap. If the number of components is chosen correctly, The Euclidean distance is best suited to this model because it treats overestimation and underestimation equally and the model is exact. The generalized KL-divergence favors overestimation of the mixture spectrogram rather than underestimation. D_s favors overestimation and penalizes underestimation even more.

All of the cost functions are parameterized by the desired number of components. In the previous example, extracting one component per note captures the predominant harmonic content in each note. However, real instruments often contain a noise burst at the onset of a note. By extracting additional components we can hope to reveal some of this structure. Figures 79-83 show eight extracted components for each of the cost functions.

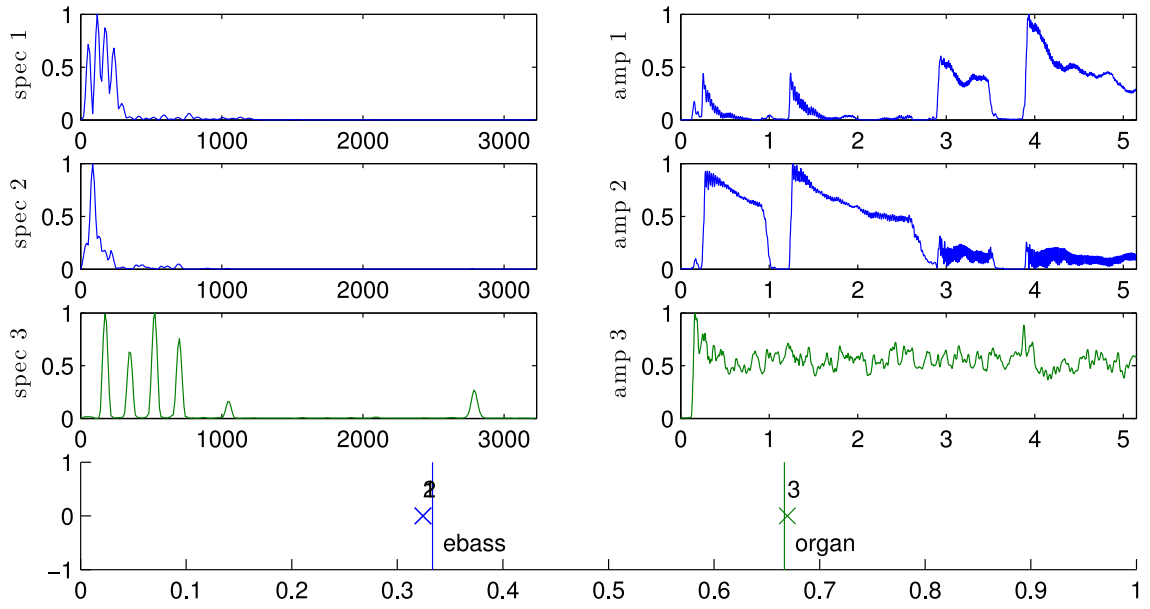


Figure 75: Three components extracted by D_m for bass and organ

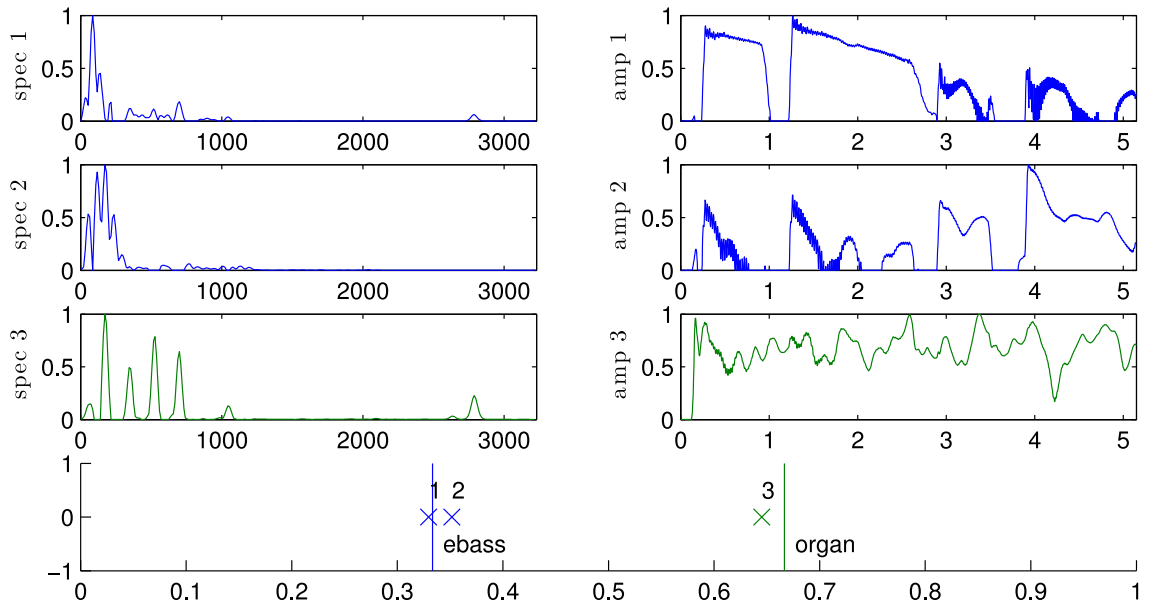


Figure 76: Three components extracted by E_p for bass and organ

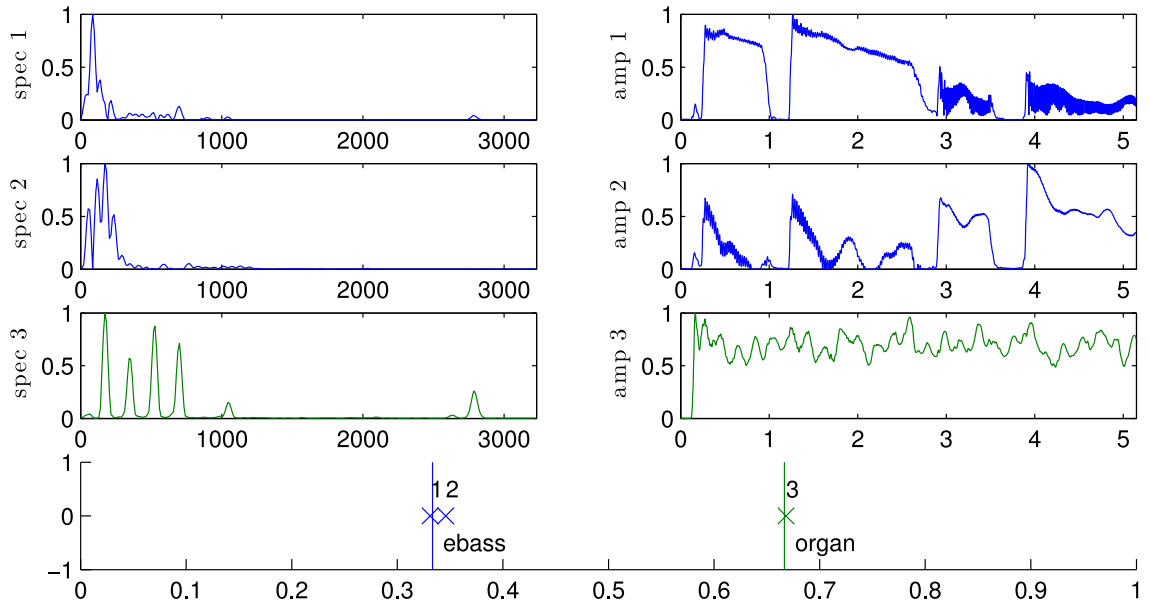


Figure 77: Three components extracted by D_p for bass and organ

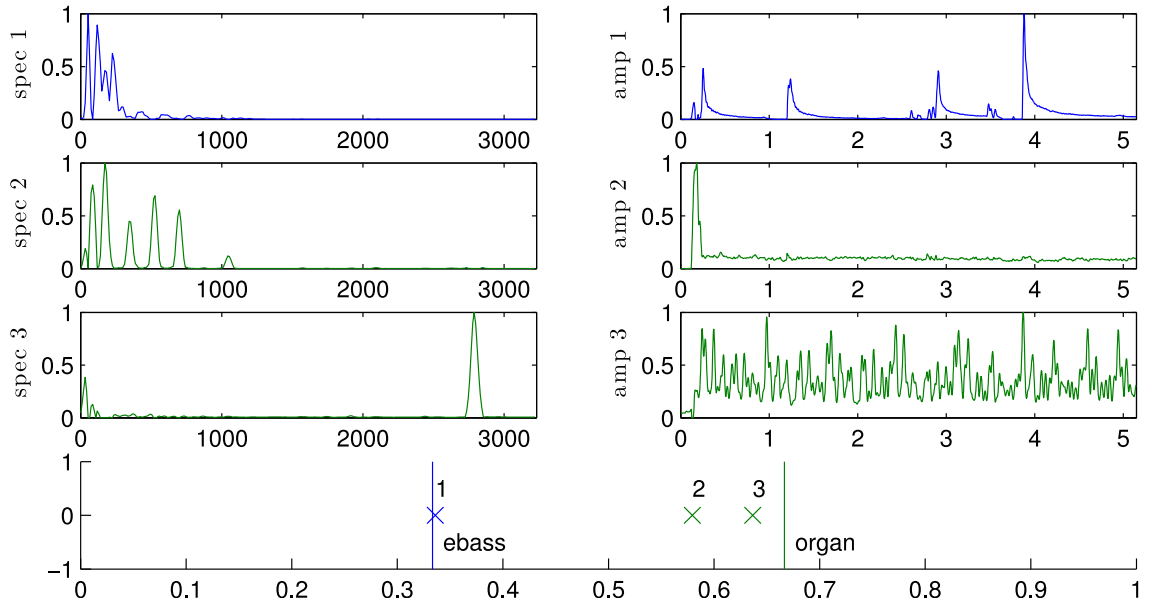


Figure 78: Three components extracted by D_s for bass and organ

E_m (Figure 79) and D_p (Figure 82) provide the most clear delineation of onset and sustain on the electric bass. The first four components capture the same essential information. The amplitudes reveal this structure. Components 1 and 4 represent the sustained portion of each plucked electric bass guitar note. Components 2 and 3 capture the sharp attack and decay of the noisy onset. The spectral shapes still capture the dominant spectral peaks. In particular, components 2 and 3 capture frequency content that decays faster than the rest. The other four components correspond to the electric organ. Each component emphasizes a different subset of the harmonics.

D_m estimates three components for electric bass (Figure 80). Clearly, the 4th component represents part of both instruments. Its low frequency energy and amplitude peaks capture the electric bass while the high-frequency harmonics correspond to the organ. Its spatial position toward the middle of the two instruments reflects this as well. The final four components correspond to subsets of the organ harmonics.

E_p appears to estimate four components for each instrument (Figure 81). Components 1 and 4 capture the sustained part of both bass notes. Component 3 appears to represent the noisy onset of all four notes (regardless of pitch). Components 2 and 6 are highly related. It appears that their similarity in spectral shape and amplitude envelope made their spatial estimation more difficult, pushing component 6 to the right and component 2 to the left. Both components appear to best capture parts of the organ note. However, the bass notes also contain the frequency peak in component 2, resulting in the ambiguity.

Like D_m , D_s estimates only 3 components for the bass notes (Figure 83). The first component captures the same harmonic that is active throughout both notes and peaks at the onsets. Component 2 captures onset information as well as harmonic content active during the second pair of notes. Component 3 contains low-frequency energy that is present during the sustained portion of all four notes. The remaining four components capture the organ note. Although, component 5 is most clearly associated with the onset. One apparent benefit of D_s is that it avoids the very noisy content in the amplitude envelopes of the other

methods (particularly the electric bass components).

In all, using more components to represent this real instrument mixture captures more detail about the notes. In particular, the algorithms appear to learn more information about each onset. The spatial position of each component provides a good indication of whether a component captures information specific to a single instrument or multiple instruments. However, choosing the “right” number of components presents a challenge.

4.5.2 Bass, Vocals, and Organ Example

In this example, we add another track to the recording, creating a stereo mixture of three sources. Figures 84-87 show the high-frequency content for each source and the overlapping content. While the bass and organ have a relatively constant spectral shape (Figure 84 and 86), the singing voice varies considerably (Figure 85). The voice may be the most difficult signal to separate using spectrogram factorization because the pitch can vary continuously. This causes the spectral shape to stretch as the pitch changes over time. Representing this smooth change with static spectral shapes requires many more components that capture specific time instants of the signal.

Figures 88-90 show the extracted components for D_s . Components 1 and 2 correspond to the single repeated bass note. Components 27 and 28 correspond to the single sustained organ note. The remaining components capture aspects of the voice. Clearly, the task of separating the voice signals is much more difficult than the bass and organ notes and requires many more components. We tried a variety of numbers of components and chose 28 because it is the fewest number of components that still represents the bass and organ with two components. Although the components cluster nicely around the true instrument positions, the voice components often contain frequency content associated with the organ and to a lesser extent the bass. Because the voice components are concentrated at specific points in time, they have more flexibility in representing other content at those times. For a single channel mixture, a component that is only active at one point in time is free to

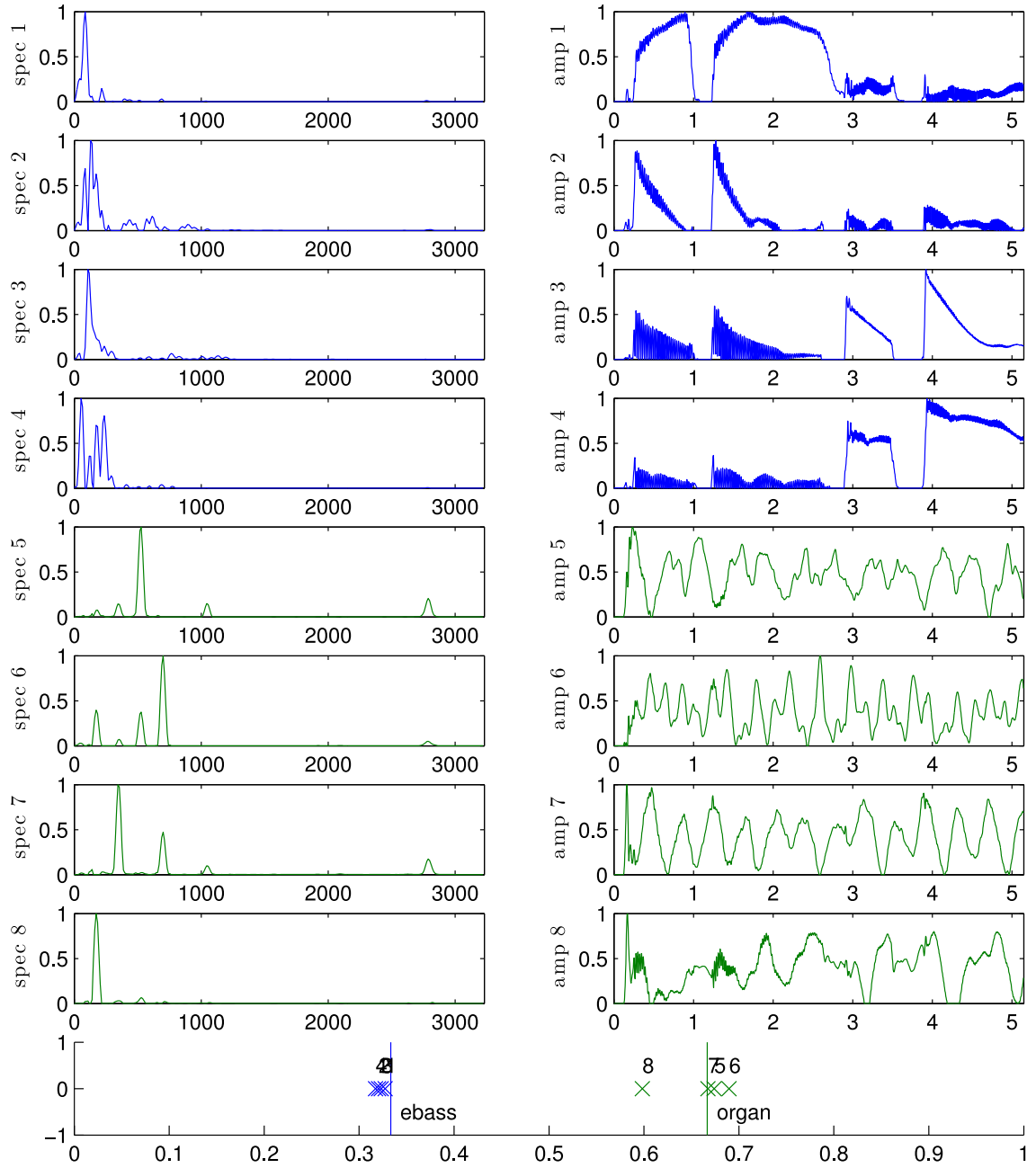


Figure 79: Eight components extracted by E_m for bass and organ

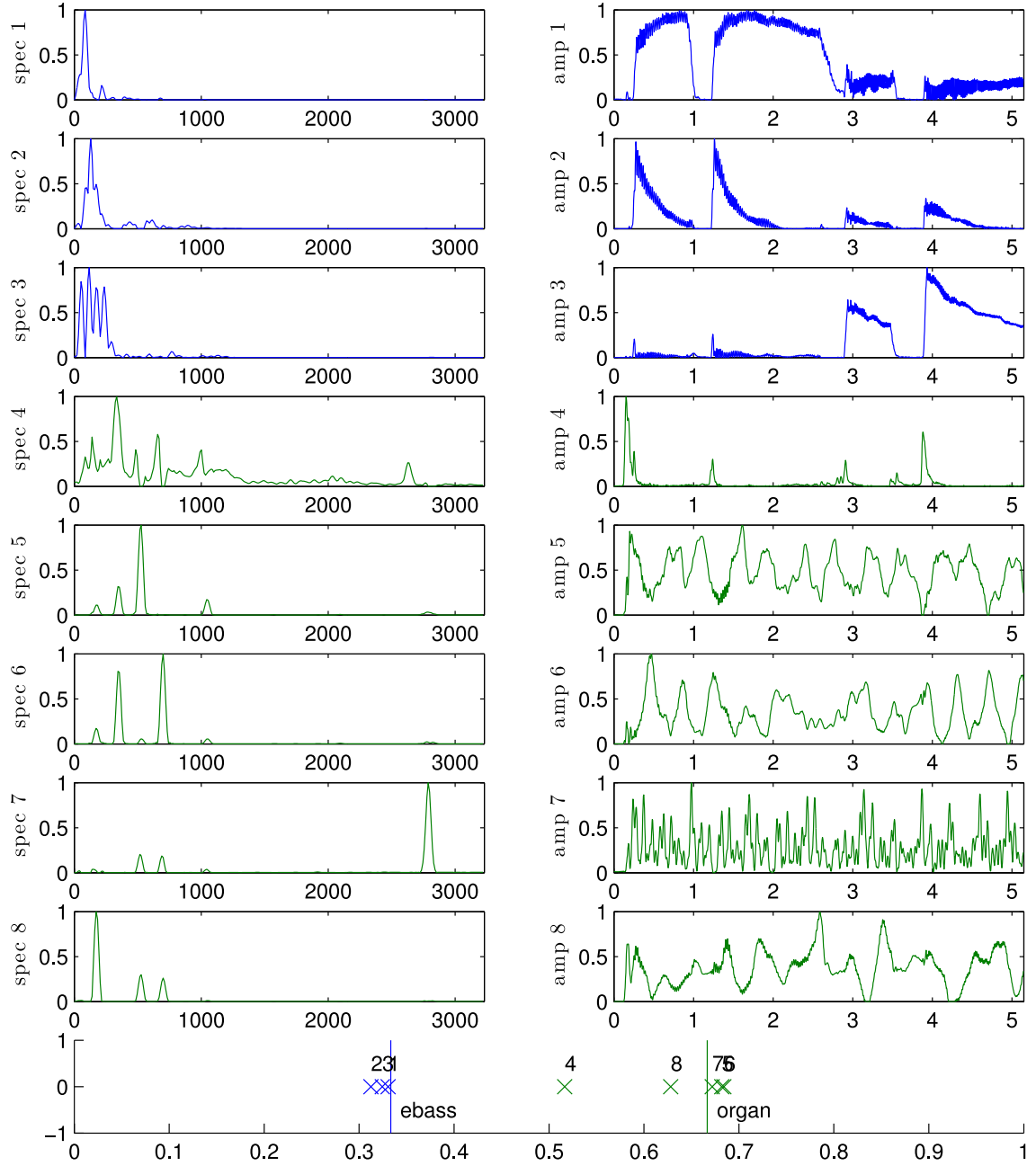


Figure 80: Eight components extracted by D_m for bass and organ

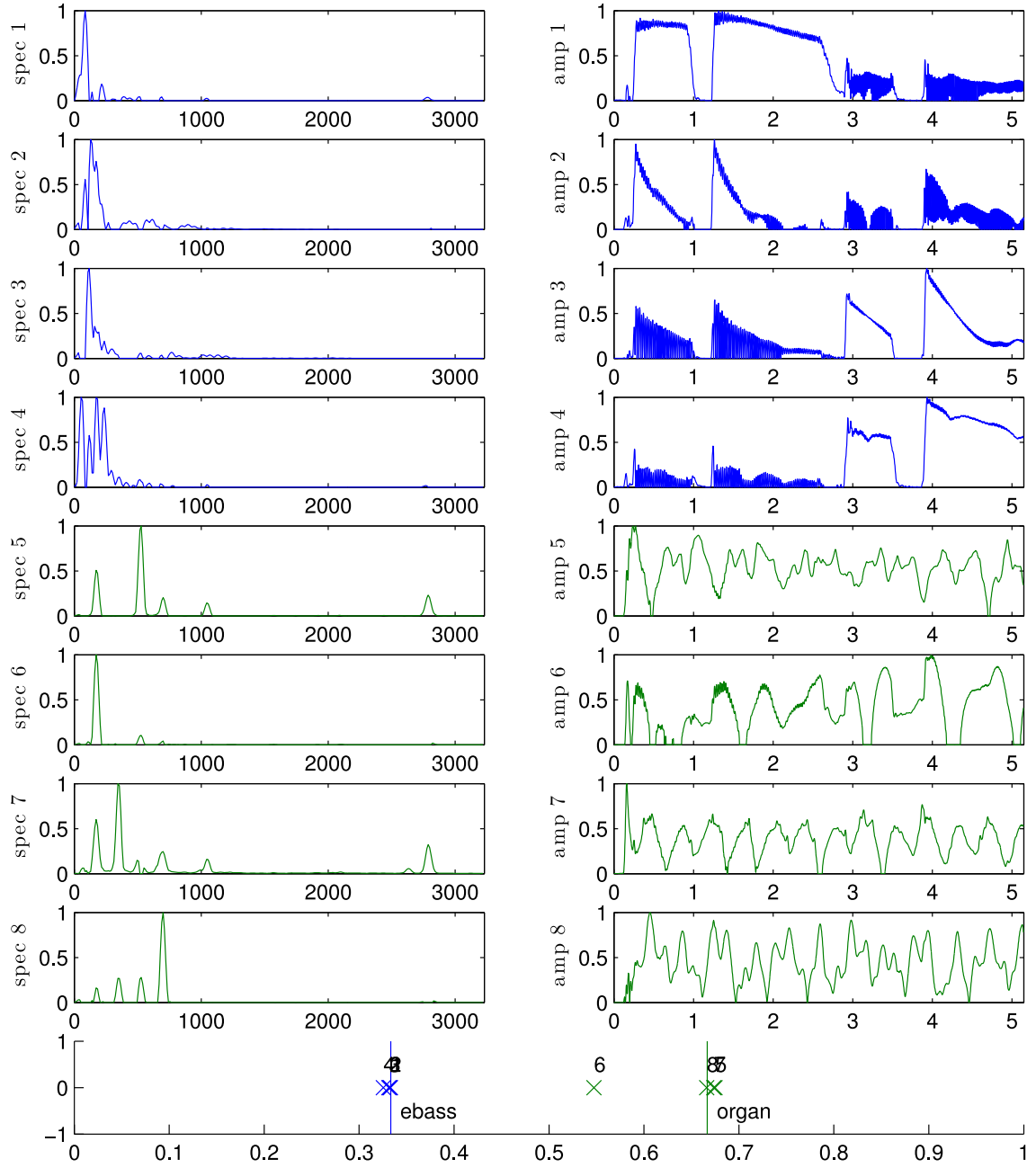


Figure 82: Eight components extracted by D_p for bass and organ

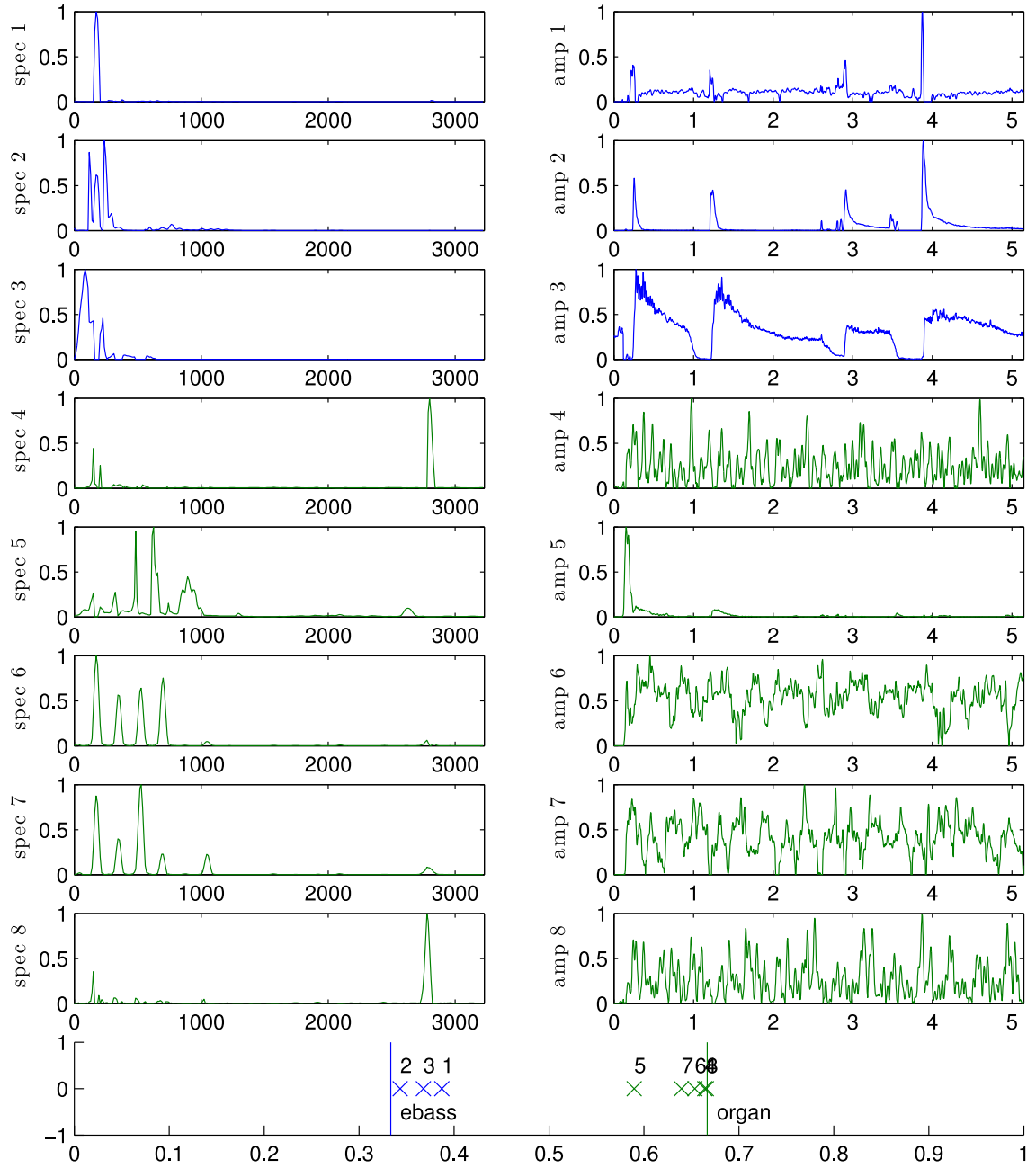


Figure 83: Eight components by D_s for bass and organ

incorporate any frequency information at that point. In the current stereo mixture, it appears that some of this information is still present even though the spatial position should dampen the effect.

This example reveals some of the shortcomings of spectrogram factorization techniques for source separation. The difficulty of estimating sources with smoothly varying spectra (and not static spectral shapes) has been addressed with shifted versions of NMF. Each component is allowed to stretch in the frequency domain to represent multiple different musical notes with the identical pitch-normalized shape (*e.g.*, [43]). In addition, a “convolutive” version of NMF estimates components with multiple concatenated spectral shapes that can capture the evolution of a spectral shape over time [115]. These components are more suited for modeling smooth transitions in pitch but require that the exact same transition occurs multiple times in the recording. Both of these advances represent tailoring NMF to a particular type of source. In addition, a prior distribution on the components (*e.g.*, a sparse prior [2]) could further inform the algorithms. In addition, some cost functions (E_m and D_p) appear to perform better when there is little overlap between components. Whereas, the phase-aware cost function is designed to perform better when there is heavy overlap (*e.g.*, when different instruments play the same note). We speculate that improvements could be obtained by integrating two cost functions so that E_m dominates the cost when only one component is active and D_s dominates when multiple components are active. All of the cost functions are parameterized by the sum of magnitude components or sum of power components. Each cost could be weighted by how much a single component dominates this sum.

Noisy transients in musical notes are not well suited to the rank-one spectrogram model. Perhaps a preprocessing step that separates harmonic parts of the spectra from the noisy parts would allow NMF to operate on the harmonic-only content. In addition, the phase of the sources must be estimated in order to transform the spectrogram into a time-domain signal. We use a probabilistic representation of phase to estimate the magnitude spectrogram

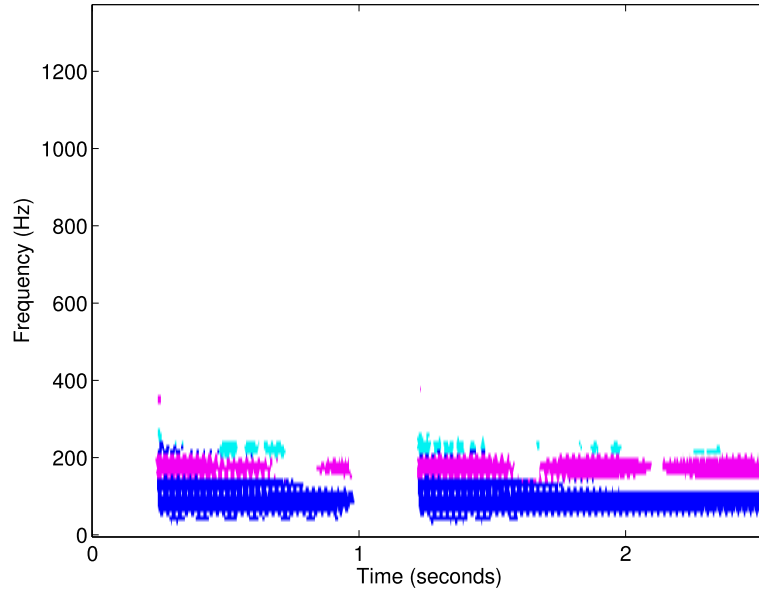


Figure 84: High energy points in the electric bass spectrogram

but we never estimate the actual phase. Phase estimation is a difficult problem without an efficient solution. Because the phase is important in the estimation of the magnitude spectrogram, perhaps it is worthwhile to concurrently estimate the phase during the estimation of the magnitude spectrograms.

Finally, all of these algorithms inherently depend on the chosen number of components. Choosing too few components makes different notes or instruments combine into a single component. Choosing too many components allows components that focus on a specific frequency or point in time and do not correspond to a single source. Ideally, choosing the right number of components estimates components that capture aspects of the signal that are specific to a single source.

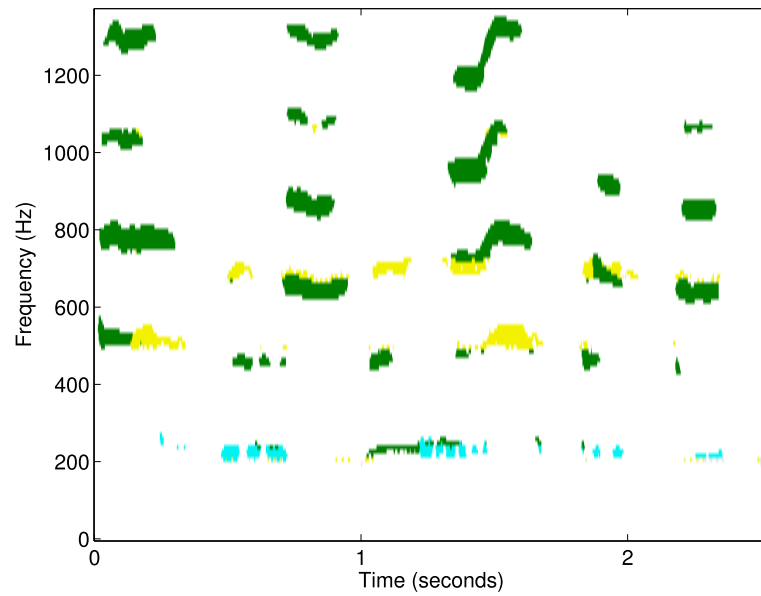


Figure 85: High energy points in the vocals spectrogram

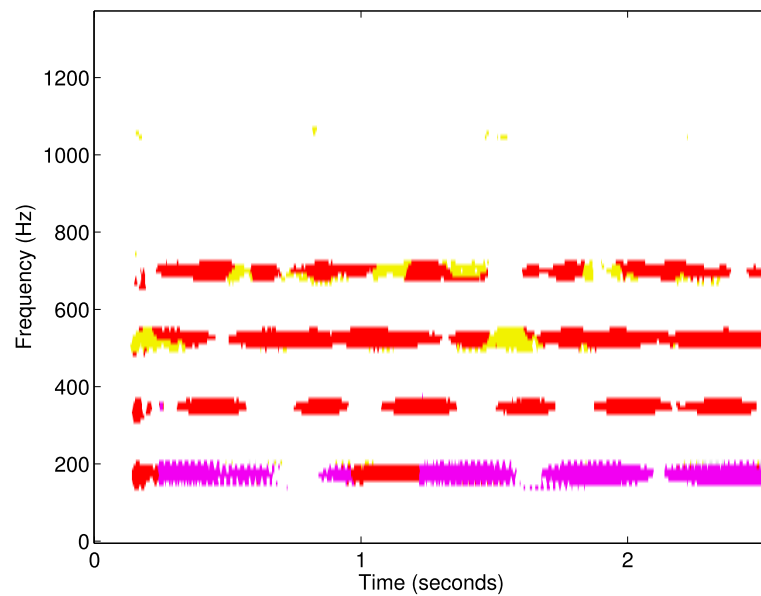


Figure 86: High energy points in the electric organ spectrogram

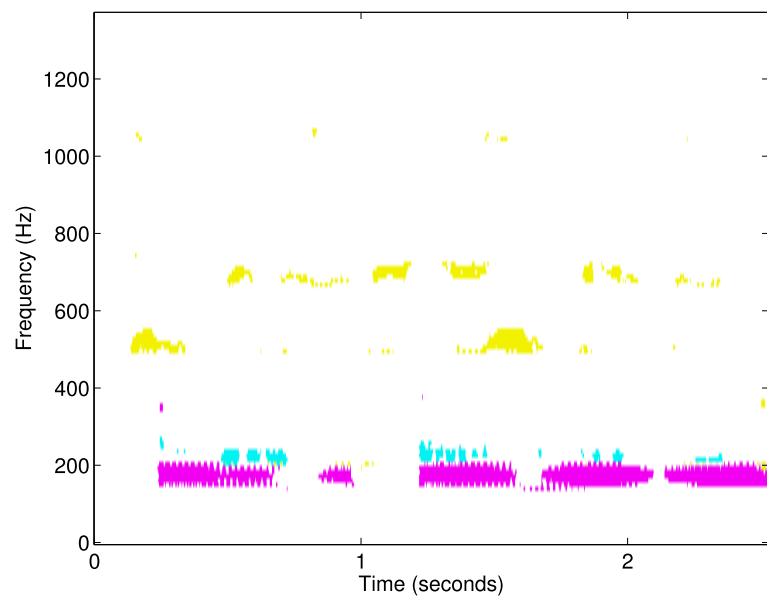


Figure 87: Overlapping high energy in the electric bass, vocals, and electric organ

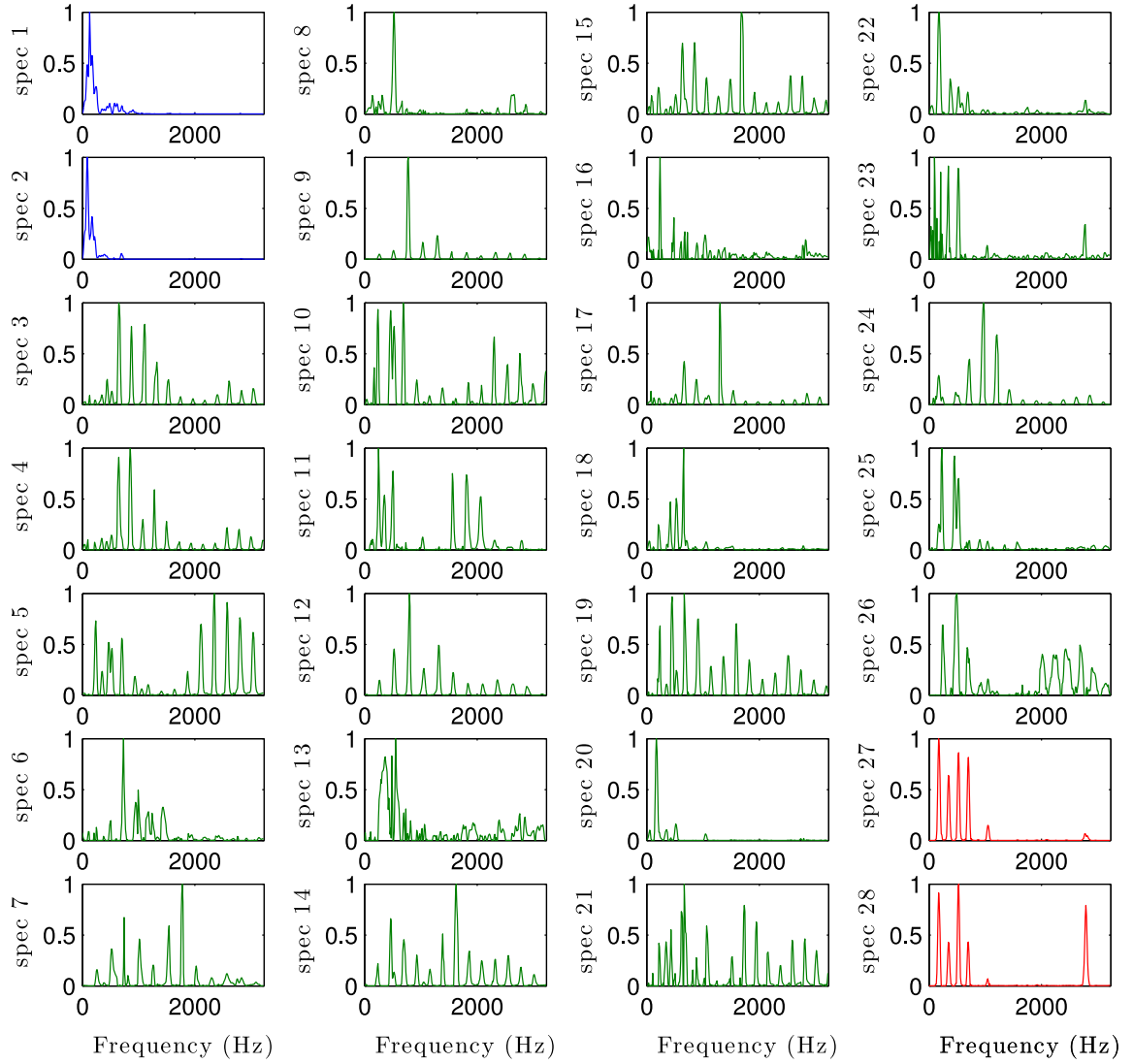


Figure 88: Spectral shapes extracted by D_s for bass, vocals, and organ mixture

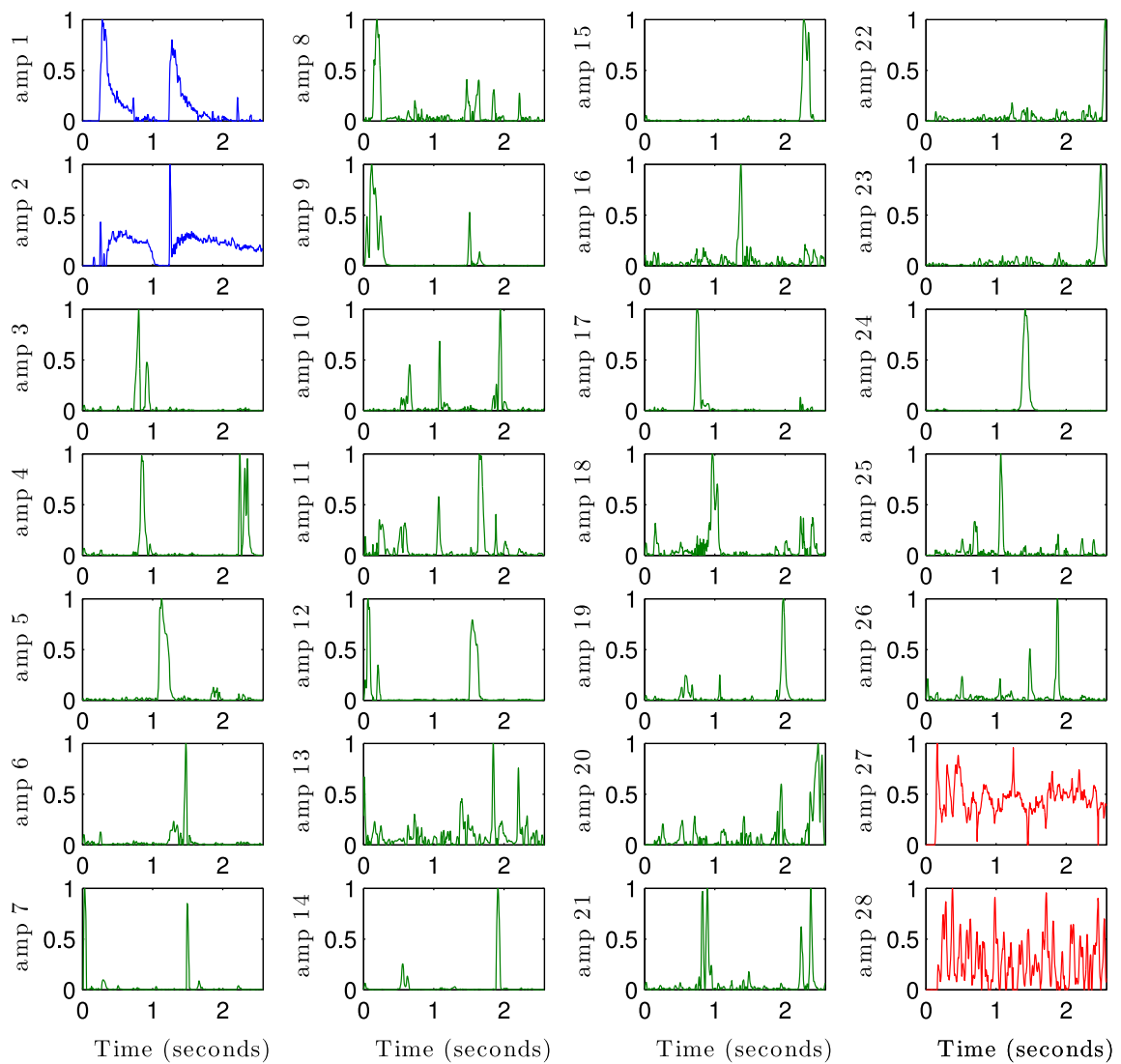


Figure 89: Amplitude envelopes extracted by D_s for bass, vocals, and organ mixture

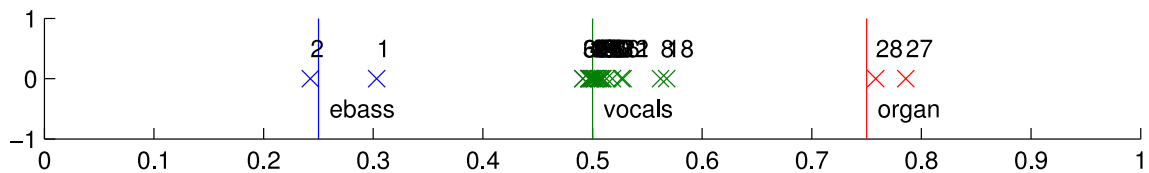


Figure 90: Spatial positions extracted by D_s for bass, vocals, and organ mixture

CHAPTER V

SUMMARY AND FUTURE WORK

This thesis focuses on separating musical instruments from a recording of their mixture. We are motivated by potential analysis and remixing applications, and by the possible extension to other data. While we have shown the relevance of our theory and techniques to musical source separation, we believe that the ideas presented here could be applied to other data appropriate for source separation.

In Chapter 3, we provide a detailed description of source separation techniques based on joint diagonalization. The various approaches leverage different types of source structure including non-Gaussian probability density functions, time-varying energy, autocovariance, and time-frequency sparseness. By borrowing the locality of time-varying energy and the time-lags of autocovariance, we present a time-aligned representation that captures the repetitions between signals. By manipulating the pseudo Wigner time-frequency representation to utilize two points in time and removing the dependency on frequency, we present a time-reversed representation that captures the time-reversed repetitions between signals. Both are time-time representations that capture the self-similarity within a signal and the cross-similarity between different signals (Section 3.4). We show that these representations capture unique information that separates sources in the joint diagonalization framework (Section 3.5). In addition, we use the time-reversed version to inform a source detection algorithm (Section 3.6).

Our time-time representations capture the repetitive structure in the source and mixture signals. This structure is prevalent in musical recordings but can reasonably be expected in speech and other audio as well as other signals. In particular, the foetal electrocardiogram (EKG) measures the heartbeat of a foetus mixed with the heartbeat of the mother and other

noise. The repetitive heartbeats potentially provide the type of time structure necessary for separation.

In Chapter 4, we extend single channel source separation techniques based on spectrogram factorization to apply to multiple channels and incorporate phase information. One difficulty in spectrogram factorization methods for source separation is determining which spectral components belong to which source. By extending spectrogram factorization methods to multiple channels, we show that the components can be clustered according to spatial position (Section 4.2.3).

Although we apply the multichannel extensions for spectrogram factorization to music audio, the underlying technology is a tensor factorization in three dimensions. For our application to audio, the dimensions are space, time, and frequency. Alternatively, our techniques apply to general three-dimensional tensor factorizations (and can easily be extended to more dimensions). Our ICA-based approach determines the factorization that makes one dimension as independent as possible, whereas our NMF-based approach preserves non-negativity in the data (Section 4.2). In particular, we envision applying this work to microarray data which is also non-negative and collected across three dimensions. Specifically, thousands of gene expression levels are measured for multiple patients at multiple points in time.

In Section 4.3, we incorporate the unknown phase of the component spectrograms in a probabilistic framework to improve the estimation of multiple overlapping components. We derive the likelihood function for the mixture spectrogram with respect to the component spectrograms for the case of two components (Section 4.3.2) and generalized to the case of an arbitrary number of components (Section 4.3.3). The two component version improves the estimation by estimating components that more closely follow the true underlying distribution (Section 4.3.2.2). We derive a cost function based on the likelihood function for an arbitrary number of sources and show that for a variety of spectrogram sizes, numbers of components, and component distributions, our proposed cost function

outperforms the competition on synthetic examples (Section 4.3.3.3). In Section 4.3.3.4 and 4.4.1, we apply the methods to musical examples composed of rank-one musical notes. In Section 4.5, we extend to the case of real (*i.e.*, full rank) musical recordings.

Incorporating phase information for spectrogram factorization is specific to the spectrogram representation but is not restricted to music or audio. The same issues and our proposed approach apply to any time-varying signal for which spectrogram factorization is reasonably applied. Of course, this requires that each source is well-approximated by the combination of rank-one component spectrograms. This assumption appears most appropriate for harmonic signals such as music and voiced speech. However, other signals may comprise a static spectral shape and amplitude envelope.

In addition to applying this work to other data sets, technical challenges remain for future work. In particular, we have used a probabilistic representation of phase to improve the estimate of magnitude spectrograms. However, in order to recover the time domain signal, we must also estimate the phase. Drawing the phase from a uniform distribution fits with our approach to estimating spectral components. Although the phase has a uniform distribution, the phase at different time-frequency points is not independent. Therefore, drawing from a uniform distribution satisfies the overall distribution of the phase but not the interdependencies. Alternatively, the phase of the mixture is often used to reconstruct the source signals from the source magnitude spectrograms. In this approach, the phase is accurate at time-frequency points where exactly one source is active. However, when more sources overlap, the mixture phase contains a combined effect. Especially when there is large overlap something better is needed. Some work has addressed the issue of estimating a time domain signal from a phaseless spectrogram [51, 3]. These approaches operate on general phaseless spectrograms. Our component spectrograms have special structure in that they are rank-one. We expect that tailoring Achan’s method to rank-one spectrograms to be an interesting extension with direct application to spectrogram factorization approaches to source separation.

APPENDIX A

ICA EXAMPLE: JOINT APPROXIMATE DIAGONALIZATION OF EIGENMATRICES

Cardoso's joint approximate diagonalization of eigenmatrices (JADE) algorithm [24,21,23] is one of the more popular ICA algorithms and is parameterized only by the number of desired sources (at most as many sources as mixtures). If the number of sources is less than the number of mixtures, the model assumes Gaussian white noise. Whitening removes the 2nd-order correlations, however, statistical independence requires no n th-order correlations (for all integers n). In practice, Cardoso finds that removing 2nd-order correlations and minimizing 4th-order correlations sufficiently separates independent components. The algorithm proceeds in three steps: decorrelation (whitening), construction and eigen-decomposition of 4th-order cumulants, and joint diagonalization of the more significant eigenmatrices. We describe these steps separately.

Cardoso assumes the original sources have a mean of zero. This is usually the case and can be made so by shifting the input mixtures to zero mean. Whitening decorrelates the data by removing 2nd-order relationships between mixtures and normalizes it to have unit variance. After whitening, the data may undergo an arbitrary rotation and still be uncorrelated. The rest of the algorithm estimates a rotation that minimizes 4th-order correlations.

The second part of the JADE algorithm estimates the 4th-order relationships using cumulants. Strictly, JADE only deals with 2nd-order and 4th-order information, whereas true independence requires cancelation of all n th-order relationships. However, this criterion is sufficient for practical applications. Fourth-order cumulants are defined as

$$\text{Cum}(a, b, c, d) = E\{abcd\} - E\{ab\}E\{cd\} - E\{ac\}E\{bd\} - E\{ad\}E\{bc\}. \quad (158)$$

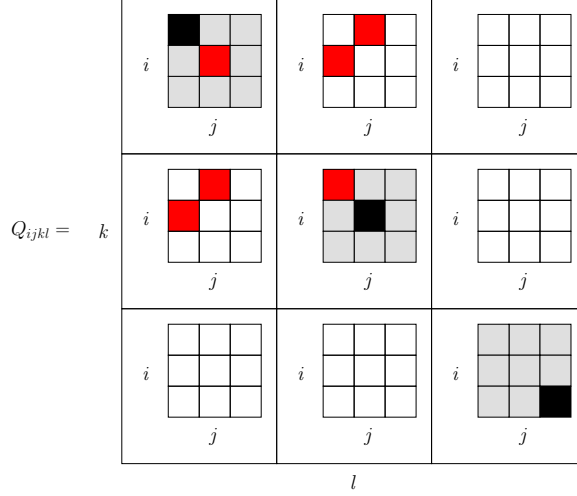


Figure 91: Redundancy in cumulant matrix

When sources a , b , c , and d are independent, their cumulant is zero. The JADE algorithm computes the 4th-order cumulants in a four-dimensional matrix, \mathbf{Q} by

$$\mathbf{Q}_{ijkl} = \text{Cum}[z_i, z_j^*, z_k, z_l^*], \quad (159)$$

where $*$ indicates complex conjugation. The goal of the algorithm is to make all entries of \mathbf{Q} zero except the main diagonal, \mathbf{Q}_{iiii} . Because \mathbf{Q} is four dimensional, explicit diagonalization is quite cumbersome. Cardoso suggests the use of two-dimensional cumulant matrices constructed from k /th matrix slice of \mathbf{Q} :

$$[\mathbf{Q}_{kl}]_{ij} = \mathbf{Q}_{ijkl} \quad (160)$$

where $i, j, k, l \in [1, N]$. Consider \mathbf{Q} as an $N \times N$ matrix where every element is an $N \times N$ matrix. Figure 91 illustrates the tensor \mathbf{Q}_{ijkl} for $N = 3$ with the indices k and l indicating a matrix and the indices i and j indicating an element. We would like each matrix along the main diagonal of \mathbf{Q} (in gray) to contain exactly one nonzero element (the black entries in Figure 91 where $i = j = k = l$), and every off-diagonal matrix of \mathbf{Q} to contain all zeros.

Cardoso and Souloumari show that if a matrix \mathbf{U} jointly diagonalizes the set of all cumulant matrices, it identifies the mixing matrix $\mathbf{A} = \mathbf{W}^{-1}\mathbf{U}$ [24]. For an intuitive explanation, consider the set of cumulant matrices in Figure 91. The matrix \mathbf{Q} is redundant

in the same way a covariance matrix is symmetric. For real signals, any ordering of the indices into \mathbf{Q} has the same value (e.g., $\mathbf{Q}_{ijkl} = \mathbf{Q}_{kji l}$). For complex signals, the magnitudes of the entries are the same. In either case, only elements \mathbf{Q}_{iiii} are represented once. For example, the red (dark gray in grayscale) elements in Figure 91 indicate permutations of $ijkl = 1122$. Although this entry appears on the diagonal of two matrices, it also appears off the diagonal of two others. Therefore, minimizing the off-diagonal entries in $[\mathbf{Q}_{ij}]_{12}$ and $[\mathbf{Q}_{ij}]_{21}$ also minimizes the entries $[\mathbf{Q}_{22}]_{11}$ and $[\mathbf{Q}_{11}]_{22}$, thus emphasizing only the element $[\mathbf{Q}_{11}]_{11}$.

An initial approach might attempt to diagonalize all of the matrix slices. However, diagonalizing all cumulant matrices would require processing N^2 matrices (each of which is $N \times N$). To make the algorithm more efficient, Cardoso instead diagonalizes only the first N eigenmatrices [23]. Eigenmatrices are computed by vectorizing every ij th matrix slice of the cumulant tensor to form a single $N^2 \times N^2$ matrix. The N eigenvectors corresponding to the largest N eigenvalues are converted back into matrix form (eigenmatrices) and jointly diagonalized. Eigenmatrices are linear combinations of the cumulant slices and diagonalizing them diagonalizes the original slices.

One way to jointly diagonalize of a set of matrices is to maximize the criterion,

$$C(\mathbf{U}, \mathbf{N}) = \sum_{r=1}^N |\text{diag}(\mathbf{U}^H \mathbf{N}_r \mathbf{U})|^2. \quad (161)$$

Cardoso maximizes the sum of the energies in the diagonal of the N matrices \mathbf{N}_r by extending the Jacobi technique for matrix diagonalization to multiple matrices [23]. For a two-dimensional matrix, a Givens rotation may be solved in closed form to diagonalize it. For more than two dimensions, the Jacobi technique applies successive Givens rotations to every pair of indices in a matrix. Cardoso and Souloumari extend the Jacobi technique to multiple matrices by solving for the best Givens rotation for all matrices [24]. This is accomplished for each index pair by considering all the Givens rotations (computed for every matrix). The eigenrotation corresponding to the largest eigenvalue of these rotations yields the single best rotation, which is applied. This is repeated for all index pairs and the whole

process is repeated until convergence.

We have described in detail one ICA algorithm that we discuss in the context of joint diagonalization approaches that incorporate various aspects of source structure in Chapter 3. The only difference is how to generate a set of correlation matrices that capture a form of source structure (in this case 4th-order correlations) that leads to separation.

APPENDIX B

DERIVATION OF MULTICHANNEL ICA-BASED NON-NEGATIVE SPECTROGRAM FACTORIZATION

In this appendix, we differentiate the entropy equations discussed in Chapter 4 starting with Bell and Sejnowski's [13] information maximization approach. Then, we differentiate Stone's undercomplete version of the entropy equation. Finally, we differentiate our undercomplete version of entropy that includes a spatial *and* spectral unmixing matrices.

B.1 Bell and Sejnowski's Information Maximization

First we derive Bell and Sejnowski's [13] gradient for maximizing the entropy of a nonlinear function of the estimated sources:

$$H(\mathbf{Y}) = H(\mathbf{Z}) + \ln |\det \mathbf{W}| + F(\mathbf{Y}) \quad (162)$$

$$F(\mathbf{Y}) = \frac{1}{T} \sum_{t=1}^T \sum_{r=1}^R \ln |1 - \mathbf{Y}_{rt}^2|, \quad (163)$$

where $\mathbf{Y} = \tanh(\mathbf{U})$ with $\mathbf{U} = \mathbf{W}\mathbf{Z}$. This applies to general signals and can be used to separate signals spatially (with multiple mixtures) or spectrally (multiple frequency channels).

We derive the gradient of $H(\mathbf{Y})$ w.r.t. \mathbf{W} :

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = \frac{\partial H(\mathbf{Z})}{\partial \mathbf{W}_{ij}} + \frac{\partial \ln |\det(\mathbf{W})|}{\partial \mathbf{W}_{ij}} + \frac{\partial F(\mathbf{Y})}{\partial \mathbf{W}_{ij}}. \quad (164)$$

Because $H(\mathbf{Z})$ does not depend on \mathbf{W} , we remove that term:

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = \frac{\partial \ln |\det(\mathbf{W})|}{\partial \mathbf{W}_{ij}} + \frac{\partial F(\mathbf{Y})}{\partial \mathbf{W}_{ij}}. \quad (165)$$

First we differentiate the log term w.r.t. \mathbf{W} :

$$\begin{aligned}
\frac{\partial \ln |\det(\mathbf{W})|}{\partial \mathbf{W}_{ij}} &= \text{Tr}(\mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \mathbf{W}_{ij}}) & \left[\partial(\ln(\det(\mathbf{A}))) &= \text{Tr}(\mathbf{A}^{-1} \partial \mathbf{A}) \right] \\
&= \text{Tr}(\mathbf{W}^{-1} \mathbf{J}^{ij}) & \left[\frac{\partial \mathbf{A}}{\partial \mathbf{A}_{ij}} &= \mathbf{J}^{ij} \right] \\
&= (\mathbf{W}^{-1})_{ji} & \left[\text{Tr}(\mathbf{A} \mathbf{J}^{ij}) &= \mathbf{A}_{ji} \right] \\
&= \mathbf{W}^{-T}, & \left[\mathbf{A}_{ij} &= (\mathbf{A}^T)_{ji} \right]
\end{aligned} \tag{166}$$

where \mathbf{J}^{ij} is a matrix with only one nonzero element, $\mathbf{J}_{ij}^{ij} = 1$. Second, we differentiate $F(\mathbf{Y})$:

$$\begin{aligned}
\partial F(\mathbf{Y}) &= \partial \left(\frac{1}{T} \sum_{rt} \ln |1 - \mathbf{Y}_{rt}^2| \right) \\
&= \frac{1}{T} \sum_{rt} \partial \ln |1 - \tanh^2(\mathbf{U}_{rt})| & [\mathbf{Y} = \tanh(\mathbf{U})] \\
&= \frac{1}{T} \sum_{rt} \partial \ln(\text{sech}^2(\mathbf{U}_{rt})) & [\text{sech}^2(x) = 1 - \tanh^2(x)] \\
&= \frac{1}{T} \sum_{rt} \frac{\partial(\text{sech}^2(\mathbf{U}_{rt}))}{\text{sech}^2(\mathbf{U}_{rt})} & [\partial \ln(u) = \frac{\partial u}{u}] \\
&= -\frac{2}{T} \sum_{rt} \tanh(\mathbf{U}_{rt}) \partial \mathbf{U}_{rt} & [\partial \text{sech}^2(u) = -2 \text{sech}^2(u) \tanh(u) \partial u] \\
&= -\frac{2}{T} \sum_{rt} \mathbf{Y}_{rt} \partial \mathbf{U}_{rt}. & [\mathbf{Y} = \tanh(\mathbf{U})]
\end{aligned} \tag{167}$$

The partial derivative of \mathbf{U}_{rt} w.r.t. \mathbf{W}_{ij} is the following:

$$\begin{aligned}
\frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} (\mathbf{W} \mathbf{Z})_{rt} & [\mathbf{U} = \mathbf{W} \mathbf{Z}] \\
&= \frac{\partial}{\partial \mathbf{W}_{ij}} \sum_k \mathbf{W}_{rk} \mathbf{Z}_{kt} \\
\frac{\partial \mathbf{U}_{it}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} \mathbf{W}_{ij} \mathbf{Z}_{jt} & \left[(\mathbf{AB})_{ij} = \sum_k \mathbf{A}_{ik} \mathbf{B}_{kj} \right] \\
&= \mathbf{Z}_{jt}. & \left[(r \neq i | k \neq j) \rightarrow \frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{W}_{ij}} = 0 \right]
\end{aligned} \tag{168}$$

The derivative is zero unless $r = i$. Substituting Equation 168 into Equation 167 we have the following:

$$\begin{aligned}\frac{\partial F(\mathbf{Y})}{\partial \mathbf{W}_{ij}} &= -\frac{2}{T} \sum_i \mathbf{Y}_{it} \mathbf{Z}_{jt} \\ &= -\frac{2}{T} (\mathbf{Y} \mathbf{Z}^T)_{ij} .\end{aligned}\quad (169)$$

Therefore, substituting Equation 166 and 169 into Equation 165 the derivative of $H(\mathbf{Y})$ w.r.t. \mathbf{W} is the following:

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}} = \mathbf{W}^{-T} - \frac{2}{T} \mathbf{Y} \mathbf{Z}^T . \quad (170)$$

B.2 Stone's Undercomplete Information Maximization

We differentiate Stone's [113] undercomplete approximation to the entropy of a nonlinear function of the sources:

$$H(\mathbf{Y}) \approx \frac{1}{2} \ln |\det \mathbf{R}_{\hat{\mathbf{H}}}| + F(\mathbf{Y}) , \quad (171)$$

where $F(\mathbf{Y})$ is in Equation 163, $\mathbf{R}_{\hat{\mathbf{H}}} = \mathbf{W} \mathbf{R}_Z \mathbf{W}^T$ and $\mathbf{R}_Z = \mathbf{Z} \mathbf{Z}^T / (T - 1)$ is the covariance of the rows of \mathbf{Z} . We differentiate $H(\mathbf{Y})$ w.r.t. \mathbf{W} :

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = \frac{1}{2} \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}_{ij}} + \frac{\partial F(\mathbf{Y})}{\partial \mathbf{W}_{ij}} , \quad (172)$$

where $L(\mathbf{W}) = \ln |\det \mathbf{R}_{\hat{\mathbf{H}}}|$ is the log term. The partial derivative of $F(\mathbf{Y})$ w.r.t. \mathbf{W}_{ij} is the same as before. However, the log term is different:

$$\partial L(\mathbf{W}) = \text{Tr}(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} (\partial \mathbf{R}_{\hat{\mathbf{H}}})) . \quad \left[\partial(\ln(\det(\mathbf{A}))) = \text{Tr}(\mathbf{A}^{-1} \partial \mathbf{A}) \right] \quad (173)$$

We differentiate the estimated covariance of the sources, $\mathbf{R}_{\hat{\mathbf{H}}}$, which is a quadratic function of \mathbf{W} :

$$\begin{aligned}\frac{\partial \mathbf{R}_{\hat{\mathbf{H}}}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} \mathbf{W} \mathbf{R}_Z \mathbf{W}^T & \left[\mathbf{R}_{\hat{\mathbf{H}}} = \mathbf{W} \mathbf{R}_Z \mathbf{W}^T \right] \\ &= \left(\frac{\partial \mathbf{W}}{\partial \mathbf{W}_{ij}} \mathbf{R}_Z \mathbf{W}^T + \mathbf{W} \mathbf{R}_Z \frac{\partial \mathbf{W}^T}{\partial \mathbf{W}_{ij}} \right) & \left[\partial(\mathbf{A} \mathbf{B}) = (\partial \mathbf{A}) \mathbf{B} + \mathbf{A} (\partial \mathbf{B}) \right] \\ &= (\mathbf{J}^{ij} \mathbf{R}_Z \mathbf{W}^T + \mathbf{W} \mathbf{R}_Z \mathbf{J}^{ji}) . & \left[\frac{\partial \mathbf{A}}{\partial \mathbf{A}_{ij}} = \mathbf{J}^{ij} \right]\end{aligned}\quad (174)$$

Substituting Equation 174 into Equation 173 yields the following differentiation for $L(\mathbf{W})$ w.r.t. \mathbf{W} :

$$\begin{aligned}
\frac{\partial L(\mathbf{W})}{\partial \mathbf{W}_{ij}} &= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \left(\mathbf{J}^{ij} \mathbf{R}_Z \mathbf{W}^T + \mathbf{W} \mathbf{R}_Z \mathbf{J}^{ji} \right) \right) \\
&= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{J}^{ij} \mathbf{R}_Z \mathbf{W}^T \right) + \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{W} \mathbf{R}_Z \mathbf{J}^{ji} \right) \quad [\text{Tr}(\mathbf{A} + \mathbf{B}) = \text{Tr}(\mathbf{A}) + \text{Tr}(\mathbf{B})] \\
&= \text{Tr} \left(\mathbf{R}_Z \mathbf{W}^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{J}^{ij} \right) + \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{W} \mathbf{R}_Z \mathbf{J}^{ji} \right) \quad [\text{Tr}(\mathbf{ABCD}) = \text{Tr}(\mathbf{CDAB})] \\
&= \left(\mathbf{R}_Z \mathbf{W}^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \right)_{ji} + \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{W} \mathbf{R}_Z \right)_{ij} \quad [\text{Tr}(\mathbf{A} \mathbf{J}^{ij}) = \mathbf{A}_{ji}] \\
&= \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-T} \mathbf{W} \mathbf{R}_Z^T \right)_{ij} + \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{W} \mathbf{R}_Z \right)_{ij} \quad [(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T] \\
&= 2 \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{W} \mathbf{R}_Z \right)_{ij} . \quad [\mathbf{R}_{\hat{\mathbf{H}}} = \mathbf{R}_{\hat{\mathbf{H}}}^T, \mathbf{R}_Z = \mathbf{R}_Z^T] \quad (175)
\end{aligned}$$

Therefore, substituting Equation 169 and 175 into Equation 172, the partial derivative of $H(\mathbf{Y})$ w.r.t. \mathbf{W} is the following:

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = (\mathbf{W} \mathbf{R}_Z \mathbf{W}^T)^{-1} \mathbf{W} \mathbf{R}_Z - \frac{2}{T} \mathbf{Y} \mathbf{Z}^T . \quad (176)$$

B.3 Our Undercomplete Information Maximization for Multichannel NSF

We derive the gradient for our multichannel NSF version of undercomplete ICA using the factorization $\mathbf{H} = \bar{\mathbf{V}} \bar{\mathbf{W}} \bar{\mathbf{D}} \bar{\mathbf{X}}$. For two channels ($M = 2$) the factorization looks like this:

$$\mathbf{H} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{V}_2 \end{bmatrix} \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}, \quad (177)$$

where \mathbf{V}_m is a diagonal matrix containing the m -th row of \mathbf{V} . The matrix \mathbf{V} is the spatial unmixing matrix, whereas \mathbf{W} is the spectral unmixing matrix, and \mathbf{D} is the whitening matrix for the mean of the mixtures, \mathbf{X}_m . The undercomplete approximation to the entropy of the sources is the following:

$$H(\mathbf{Y}) \approx \frac{1}{2} \ln |\det(\mathbf{R}_{\hat{\mathbf{H}}})| + F(\mathbf{Y}), \quad (178)$$

where $\mathbf{R}_{\hat{\mathbf{H}}} = \bar{\mathbf{V}} \bar{\mathbf{W}} \mathbf{R}_Z \bar{\mathbf{W}}^T \bar{\mathbf{V}}^T$ and $\mathbf{Y} = \tanh(\mathbf{U})$, with a change in \mathbf{U} such that $\mathbf{U} = \bar{\mathbf{V}} \bar{\mathbf{W}} \bar{\mathbf{Z}}$. We differentiate w.r.t. \mathbf{W} and \mathbf{V} to find the partial derivatives of H :

$$\partial H(\mathbf{Y}) = \frac{1}{2} \partial L(\bar{\mathbf{V}}, \bar{\mathbf{W}}) + \partial F(\mathbf{Y}), \quad (179)$$

where $L(\bar{\mathbf{V}}, \bar{\mathbf{W}}) = \ln |\det(\mathbf{R}_{\hat{\mathbf{H}}})|$. First, we find the partial derivative of $L(\bar{\mathbf{V}}, \bar{\mathbf{W}})$:

$$\partial L(\bar{\mathbf{V}}, \bar{\mathbf{W}}) = \text{Tr}(\mathbf{R}_{\hat{\mathbf{H}}}^{-1}(\partial \mathbf{R}_{\hat{\mathbf{H}}})) . \quad \left[\partial(\ln(\det(\mathbf{A}))) = \text{Tr}(\mathbf{A}^{-1} \partial \mathbf{A}) \right] \quad (180)$$

For convenience, we write the estimated covariance of the sources, $\mathbf{R}_{\hat{\mathbf{H}}}$, as a sum of the product of simpler matrices:

$$\mathbf{R}_{\hat{\mathbf{H}}} = \sum_m \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T . \quad (181)$$

We differentiate the estimated covariance of the sources, $\mathbf{R}_{\hat{\mathbf{H}}}$, which is a quadratic function of \mathbf{V}_m and \mathbf{W} :

$$\begin{aligned} \frac{\partial \mathbf{R}_{\hat{\mathbf{H}}}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} \sum_m \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T \\ &= \sum_m (\mathbf{V}_m \mathbf{J}^{ij} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T + \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{J}^{ji} \mathbf{V}_m^T) \quad \left[\frac{\partial \mathbf{A}}{\partial \mathbf{A}_{ij}} = \mathbf{J}^{ij} \right] \end{aligned} \quad (182)$$

$$\frac{\partial \mathbf{R}_{\hat{\mathbf{H}}}}{\partial \mathbf{V}_{ij}} = \mathbf{J}^{jj} \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{V}_i^T + \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{J}^{jj} . \quad \left[\frac{\partial \mathbf{V}_i}{\partial \mathbf{V}_{ij}} = \mathbf{J}^{jj} \right] \quad (183)$$

Because \mathbf{V}_m is a diagonal matrix containing the m th row of \mathbf{V} , $\partial \mathbf{V}_m / \partial \mathbf{V}_{ij}$ is only nonzero when $m = i$. Substituting Equation 182 into Equation 180, we find the derivative of $L(\bar{\mathbf{V}}, \bar{\mathbf{W}})$ w.r.t. \mathbf{W} :

$$\begin{aligned} \frac{\partial L(\bar{\mathbf{V}}, \bar{\mathbf{W}})}{\partial \mathbf{W}_{ij}} &= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \left(\sum_m (\mathbf{V}_m \mathbf{J}^{ij} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T + \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{J}^{ji} \mathbf{V}_m^T) \right) \right) \\ &= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \sum_m \mathbf{V}_m \mathbf{J}^{ij} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T \right) + \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \sum_m \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{J}^{ji} \mathbf{V}_m^T \right) \\ &= \sum_m \text{Tr} (\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{J}^{ij} \mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T) + \sum_m \text{Tr} (\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{J}^{ji} \mathbf{V}_m^T) \\ &= \sum_m \text{Tr} (\mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{J}^{ij}) + \sum_m \text{Tr} (\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} \mathbf{J}^{ji}) \\ &= \sum_m (\mathbf{R}_{\mathbf{Z}_m} \mathbf{W}^T \mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m)_{ji} + \sum_m (\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m})_{ij} \\ &= \sum_m (\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m}^T)_{ij} + \sum_m (\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m})_{ij} \\ &= 2 \sum_m (\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m})_{ij} . \end{aligned} \quad (184)$$

Substituting Equation 183 into Equation 180, we find the derivative of $L(\bar{\mathbf{V}}, \bar{\mathbf{W}})$ w.r.t. \mathbf{V} :

$$\begin{aligned}
\frac{\partial L(\bar{\mathbf{V}}, \bar{\mathbf{W}})}{\partial \mathbf{V}_{ij}} &= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \left(\mathbf{J}^{jj} \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{V}_i^T + \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{J}^{jj} \right) \right) \\
&= \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{J}^{jj} \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{V}_i^T \right) + \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{J}^{jj} \right) \\
&= \text{Tr} \left(\mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{V}_i^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{J}^{jj} \right) + \text{Tr} \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{J}^{jj} \right) \\
&= \left(\mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \mathbf{V}_i^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \right)_{jj} + \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \right)_{jj} \\
&= \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-T} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \right)_{jj} + \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \right)_{jj} \\
&= 2 \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T \right)_{jj} .
\end{aligned} \tag{185}$$

The function $F(\mathbf{Y})$ takes the same form as Equation 163 except $\mathbf{U} = \sum_m \mathbf{V}_m \mathbf{W} \mathbf{Z}_m$. We find the derivative of \mathbf{U}_{rt} w.r.t. \mathbf{W}_{ij} :

$$\begin{aligned}
\frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} \left(\sum_m \mathbf{V}_m \mathbf{W} \mathbf{Z}_m \right)_{rt} \\
&= \frac{\partial}{\partial \mathbf{W}_{ij}} \sum_{ml} (\mathbf{V}_m)_{rr} \mathbf{W}_{rl} (\mathbf{Z}_m)_{lt} \\
\frac{\partial \mathbf{U}_{it}}{\partial \mathbf{W}_{ij}} &= \frac{\partial}{\partial \mathbf{W}_{ij}} \sum_m (\mathbf{V}_m)_{ii} \mathbf{W}_{ij} (\mathbf{Z}_m)_{jt} \quad \left[(r \neq i | l \neq j) \rightarrow \frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{W}_{ij}} = 0 \right] \\
&= \sum_m (\mathbf{V}_m)_{ii} (\mathbf{Z}_m)_{jt} .
\end{aligned} \tag{186}$$

The derivative is zero unless $r = i$ and $l = j$. We find the derivative of \mathbf{U}_{rt} w.r.t. \mathbf{V}_{ij} :

$$\begin{aligned}
\frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{V}_{ij}} &= \frac{\partial}{\partial \mathbf{V}_{ij}} \sum_{ml} (\mathbf{V}_m)_{rr} \mathbf{W}_{rl} (\mathbf{Z}_m)_{lt} \\
\frac{\partial \mathbf{U}_{jt}}{\partial \mathbf{V}_{ij}} &= \frac{\partial}{\partial \mathbf{V}_{ij}} \sum_l (\mathbf{V}_i)_{jj} \mathbf{W}_{jl} (\mathbf{Z}_i)_{lt} \quad \left[(r \neq j | m \neq i) \rightarrow \frac{\partial \mathbf{U}_{rt}}{\partial \mathbf{V}_{ij}} = 0 \right] \\
&= \sum_l \mathbf{W}_{jl} (\mathbf{Z}_i)_{lt} \\
&= (\mathbf{W} \mathbf{Z}_i)_{jt} .
\end{aligned} \tag{187}$$

The derivative is zero unless $r = j$ and $m = i$. Substituting Equation 186 into Equation 167 we find the derivative of $F(\mathbf{Y})$ w.r.t. \mathbf{W}_{ij} :

$$\begin{aligned}
\frac{\partial F(\mathbf{Y})}{\partial \mathbf{W}_{ij}} &= -\frac{2}{T} \sum_t \mathbf{Y}_{it} \sum_m (\mathbf{V}_m)_{ii} (\mathbf{Z}_m)_{jt} \\
&= -\frac{2}{T} \sum_{mt} \mathbf{Y}_{it} (\mathbf{V}_m)_{ii} (\mathbf{Z}_m)_{jt} \\
&= -\frac{2}{T} \sum_m (\mathbf{V}_m)_{ii} (\mathbf{Y} \mathbf{Z}_m^T)_{ij} \\
&= -\frac{2}{T} \sum_m (\mathbf{V}_m \mathbf{Y} \mathbf{Z}_m^T)_{ij} .
\end{aligned} \tag{188}$$

Substituting Equation 187 into Equation 167 we find the derivative of $F(\mathbf{Y})$ w.r.t. \mathbf{V}_{ij} :

$$\begin{aligned}
\frac{\partial F(\mathbf{Y})}{\partial \mathbf{V}_{ij}} &= -\frac{2}{T} \sum_t \mathbf{Y}_{jt} (\mathbf{W} \mathbf{Z}_i)_{jt} \\
&= -\frac{2}{T} (\mathbf{Y} (\mathbf{W} \mathbf{Z}_i)^T)_{jj} \\
&= -\frac{2}{T} (\mathbf{Y} \mathbf{Z}_i^T \mathbf{W}^T)_{jj} .
\end{aligned} \tag{189}$$

Therefore, substituting Equation 184 and 188 into Equation 179 we find the derivative of $H(\mathbf{Y})$ w.r.t. \mathbf{W}_{ij} in our spatial-spectral entropy maximization:

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}_{ij}} = \sum_m \left(\mathbf{V}_m^T \mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_m \mathbf{W} \mathbf{R}_{\mathbf{Z}_m} - \frac{2}{T} \mathbf{V}_m \mathbf{Y} \mathbf{Z}_m^T \right)_{ij} . \tag{190}$$

Substituting Equation 185 and 189 into Equation 179 we find the derivative of $H(\mathbf{Y})$ w.r.t. \mathbf{V}_{ij} :

$$\frac{\partial H(\mathbf{Y})}{\partial \mathbf{V}_{ij}} = \left(\mathbf{R}_{\hat{\mathbf{H}}}^{-1} \mathbf{V}_i \mathbf{W} \mathbf{R}_{\mathbf{Z}_i} \mathbf{W}^T - \frac{2}{T} \mathbf{Y} \mathbf{Z}_i^T \mathbf{W}^T \right)_{jj} . \tag{191}$$

REFERENCES

- [1] ABDALLAH, S. A. and PLUMBLEY, M. D., “Application of geometric dependency analysis to the separation of convolved mixtures,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Granada, Spain), September 2004.
- [2] ABDALLAH, S. A. and PLUMBLEY, M. D., “Polyphonic transcription by non-negative sparse coding of power spectra,” in *Proceedings of the International Conference on Music Information Retrieval*, (Barcelona, Spain), pp. 318–325, October 2004.
- [3] ACHAN, K., ROWEIS, S. T., and FREY, B. J., “Probabilistic inference of speech signals from phaseless spectrograms,” in *Advances in Neural Information Processing Systems 16*, MIT Press, 2004.
- [4] AMARI, S.-I., “Natural gradient learning for over- and under-complete bases in ICA,” *Neural Computation*, vol. 11, pp. 1875–1883, November 1999.
- [5] AMARI, S.-I., CICHOCKI, A., and YANG, H. H., “A new learning algorithm for blind source separation,” in *Advances in Neural Information Processing Systems 8*, pp. 757–763, MIT Press, 1996.
- [6] AOUDA, S., ZOUBIR, A. M., and SEE, C. M. S., “A comparative study on source number detection,” in *Proceedings of the International Symposium on Signal Processing and its Applications*, vol. 1, (Paris), pp. 173–176, July 2003.
- [7] AUCOUTURIER, J.-J. and SANDLER, M., “Using long-term structure to retrieve music: Representation and matching,” in *Proceedings of the International Conference on Music Information Retrieval*, (Bloomington, IN), October 2001.
- [8] BACH, F. R. and JORDAN, M. I., “Blind one-microphone speech separation: A spectral learning approach,” in *Advances in Neural Information Processing Systems 17* (SAUL, L. K., WEISS, Y., and BOTTOU, L., eds.), pp. 65–72, MIT Press, 2005.
- [9] BANSAL, D., RAJ, B., and SMARAGDIS, P., “Bandwidth expansion of narrowband speech using non-negative matrix factorization,” in *Eurospeech*, September 2005.
- [10] BARRY, D., LAWLOR, B., and COYLE, E., “Sound source separation: Azimuth discrimination and resynthesis,” in *Proceedings of International Conference on Digital Audio Effects*, (Naples, Italy), pp. 240–244, October 2004.
- [11] BARTSCH, M. and WAKEFIELD, G. H., “To catch a chorus: Using chroma-based representations for audio thumbnailing,” in *Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), October 2001.

- [12] BASU, S., “ICA: A critical review of three prominent approaches,” tech. rep., Perceptual Computing Section, The MIT Media Laboratory, Cambridge, Massachusetts, April 2000.
- [13] BELL, A. and SEJNOWSKI, T. J., “An information-maximization approach to blind separation and blind deconvolution,” *Neural Computation*, vol. 7, pp. 1129–1159, 1995.
- [14] BELLO, J. P., DUXBURY, C., DAVIES, M., and SANDLER, M., “On the use of phase and energy for musical onset detection in the complex domain,” *IEEE Signal Processing Letters*, vol. 11, no. 6, pp. 553–556, 2004.
- [15] BELLO, J. P. and SANDLER, M. B., “Phase-based note onset detection for music signals,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 441–444, 2003.
- [16] BELOUCHRANI, A., ABED-MERAİM, K., AMIN, M. G., and ZOUBIR, A. M., “Joint anti-diagonalization for blind source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Salt Lake City, UT), pp. 2789–2792, May 2001.
- [17] BELOUCHRANI, A., ABED-MERAİM, K., and CARDOSO, J.-F., “A blind source separation technique using second-order statistics,” *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.
- [18] BELOUCHRANI, A. and AMIN, M. G., “Blind source separation using time-frequency distributions: Algorithm and asymptotic performance,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Munich, Germany), pp. 3469–3472, April 1997.
- [19] BELOUCHRANI, A. and AMIN, M. G., “Blind source separation based on time-frequency signal representations,” *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 2888–2897, 1998.
- [20] BROWN, J. C. and SMARAGDIS, P., “Independent component analysis for automatic note extraction from musical trills,” *Journal of the Acoustical Society of America*, vol. 115, no. 5, pp. 2295–2306, 2004.
- [21] CARDOSO, J.-F., “On the performance of orthogonal source separation algorithms,” in *Proceedings of the European Signal Processing Conference*, (Edinburgh, Scotland), pp. 776–779, 1994.
- [22] CARDOSO, J.-F., “Blind signal separation: Statistical principles,” *Proceedings of the IEEE*, vol. 9, no. 10, pp. 2009–2025, 1998.
- [23] CARDOSO, J.-F., “High-order contrasts for independent component analysis,” *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.
- [24] CARDOSO, J.-F. and SOULOUMIAC, A., “Blind beamforming for non Gaussian signals,” *IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370, 1993.

- [25] CASEY, M. and WESTNER, W., "Separation of mixed audio sources by independent subspace analysis," in *Proceedings of the International Computer Music Conference*, (Berlin), August 2000.
- [26] CASPARY, O., NUS, P., and CECCHIN, T., "The source number estimation based on Gerschgorin radii," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Seattle, WA), pp. 1993–1996, May 1998.
- [27] CEMGIL, A. T., KAPPEN, B., and BARBER, D., "Generative model based polyphonic music transcription," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), October 2003.
- [28] CICHOCKI, A., THAWONMAS, R., and AMARI, S., "Sequential blind signal extraction in order specified by stochastic properties," *Electronics Letters*, vol. 33, pp. 64–65, 1997.
- [29] CLAASEN, T. A. C. M. and MECKLENBRÄUKER, W. F. G., "The Wigner distribution - a tool for time-frequency signal analysis, part 1: Continuous-time signals," *Philips Journal of Research*, vol. 35, no. 3, pp. 217–250, 1980.
- [30] CLAASEN, T. A. C. M. and MECKLENBRÄUKER, W. F. G., "The Wigner distribution - a tool for time-frequency signal analysis, part 2: Discrete-time signals," *Philips Journal of Research*, vol. 35, no. 4/5, pp. 276–300, 1980.
- [31] CLARISSE, L. P., MARTENS, J. P., LESAFFRE, M., DE BAETS, B., DE MEYER, H., and LEMAN, M., "An auditory model based transcriber of singing sequences," in *Proceedings of the International Conference on Music Information Retrieval*, (Orlando, FL), May 2002.
- [32] COMON, P., "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.
- [33] COOPER, M. and FOOTE, J., "Summarizing popular music via structural analysis," in *Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), October 2003.
- [34] DE MULDER, T., MARTENS, J. P., LESAFFRE, M., LEMAN, M., DE BAETS, B., and DE MEYER, H., "An auditory model based transcriber of vocal queries," in *Proceedings of the International Conference on Music Information Retrieval*, (Baltimore, MD), October 2003.
- [35] DE MULDER, T., MARTENS, J. P., LESAFFRE, M., LEMAN, M., DE BAETS, B., and DE MEYER, H., "Recent improvements of an auditory model based front-end for the transcription of vocal queries," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 257–260, 2003.
- [36] ELLIS, D. P. W., *Prediction-Driven Computational Auditory Scene Analysis*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1996.

- [37] EVERY, M. R. and SZYMANSKI, J. E., “A spectral-filtering approach to music signal separation,” in *Proceedings of International Conference on Digital Audio Effects*, (Naples, Italy), pp. 197–200, October 2004.
- [38] FELLER, W., *An Introduction to Probability Theory and Its Applications*. New York: Wiley, 1971.
- [39] FÉVOTTE, C. and DONCARLI, C., “A unified presentation of blind separation methods for convolutive mixtures using block-diagonalization,” in *Proceedings of International Symposium on Independent Component Analysis and Blind Signal Separation*, (Nara, Japan), pp. 349–354, April 2003.
- [40] FÉVOTTE, C. and DONCARLI, C., “Two contributions to blind source separation using time-frequency distributions,” *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 386–389, 2004.
- [41] FITZGERALD, D., COYLE, E., and LAYLOR, B., “Sub-band independent subspace analysis for drum transcription,” in *Proceedings of International Conference on Digital Audio Effects*, (Hamburg, Germany), pp. 65–69, September 2002.
- [42] FITZGERALD, D., CRANITCH, M., and COYLE, E., “Non-negative tensor factorisation for sound source separation,” in *Proceedings of Irish Signals and Systems Conference*, (Dublin, Ireland), September 2005.
- [43] FITZGERALD, D., CRANITCH, M., and COYLE, E., “Sound source separation using shifted non-negative tensor factorisation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Toulouse, France), May 2006.
- [44] FOOTE, J., “Visualizing music and audio using self-similarity,” in *Proceedings of ACM Multimedia*, (Orlando, FL), pp. 77–80, November 1999.
- [45] FOOTE, J., “ARTHUR: Retrieving orchestral music by long-term structure,” in *Proceedings of the International Conference on Music Information Retrieval*, (Plymouth, MA), October 2000.
- [46] FOOTE, J. and COOPER, M., “Visualizing musical structure and rhythm via self-similarity,” in *Proceedings of the International Computer Music Conference*, (Göteborg, Sweden), September 2002.
- [47] FOOTE, J. and COOPER, M., “Media segmentation using self-similarity decomposition,” in *Proceedings of SPIE*, 2003.
- [48] FRITTS, L., “University of Iowa Musical Instrument Samples Database,” 1997. available online at <http://theremin.music.uiowa.edu>.
- [49] GODSILL, S. and DAVY, M., “Bayesian harmonic models for musical pitch estimation and analysis,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Orlando, FL), pp. 1769–1772, May 2002.

- [50] GODSMARK, D. and BROWN, G. J., “A blackboard architecture for computational auditory scene analysis,” *Speech Communication*, vol. 27, pp. 351–366, 1999.
- [51] GRIFFIN, D. and LIM, J., “Signal estimation from modified short-time Fourier transform,” *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 32, pp. 236–243, 1984.
- [52] GRÖCHENIG, K., *Foundations of Time-Frequency Analysis*. Boston: Birkhäuser, 2001.
- [53] HAYTER, A. J., *Probability and Statistics for Engineers and Scientists*. Pacific Grove, CA: Duxbury, 2002.
- [54] HLAWATSCH, F. and BOUDREAUX-BARTELS, G. F., “Linear and quadratic time-frequency signal representations,” *IEEE Signal Processing Magazine*, vol. 9, pp. 21–67, April 1992.
- [55] HOLOBAR, A., FÉVOTTE, C., DONCARLI, C., and ZAZULA, D., “Single autoterms selection for blind source separation in time-frequency plane,” in *Proceedings of the European Signal Processing Conference*, (Toulouse, France), 2002.
- [56] HOYER, P. O., “Non-negative sparse coding,” in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, (Martigny, Switzerland), pp. 557–565, 2002.
- [57] HOYER, P. O., “Non-negative matrix factorization with sparseness constraints,” *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
- [58] HYVÄRINEN, A., *Independent Component Analysis*. New York: Wiley, 2001.
- [59] HYVÄRINEN, A., HOYER, P. O., and INKI, M., “Topographic independent component analysis,” *Neural Computation*, vol. 13, no. 7, pp. 1527–1558, 2001.
- [60] IKEDA, S. and MURATA, N., “A method of ICA in time-frequency domain,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Aussois, France), pp. 365–371, January 1999.
- [61] JEHAN, T., “Perceptual segment clustering for music description and time-axis redundancy cancellation,” in *Proceedings of the International Conference on Music Information Retrieval*, (Barcelona, Spain), pp. 124–127, October 2004.
- [62] JOURJINE, A., RICKARD, S., and YILMAZ, O., “Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Istanbul, Turkey), pp. 2985–2988, June 2000.
- [63] KILIAN, J. and HOOS, H. H., “Voice separation - a local optimization approach,” in *Proceedings of the International Conference on Music Information Retrieval*, (Paris, France), pp. 281–282, October 2002.

- [64] Klapuri, A., “Multiple fundamental frequency estimation by harmonicity and spectral smoothness,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.
- [65] Klapuri, A. P., “Multipitch estimation and sound separation by the spectral smoothness principle,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Salt Lake City, UT), pp. 3381–3384, May 2001.
- [66] Klapuri, A. P., Eronen, A., Seppänen, J., and Virtanen, T., “Automatic transcription of music,” in *Proceedings of Symposium on Stochastic Modeling of Music*, (Ghent, Belgium), October 2001.
- [67] Klingseisen, J. and Plumbley, M. D., “Towards musical instrument separation using multiple-cause neural networks,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Helsinki, Finland), pp. 447–452, June 2000.
- [68] Konstantinides, K. and Yao, K., “Statistical analysis of effective singular values in matrix rank determination,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, pp. 757–763, May 1988.
- [69] Krongold, B. S. and Jones, D. L., “Blind source separation of nonstationary convolutively mixed signals,” in *Proceedings of the IEEE Workshop on Statistical Signal and Array Processing*, (Pocono Manor, PA), pp. 53–57, August 2000.
- [70] Lambert, R. H., *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*. PhD thesis, University of Southern California, 1996.
- [71] Lambert, R. H. and Bell, A. J., “Blind separation of multiple speakers in a multipath environment,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, (Munich, Germany), pp. 423–426, April 1997.
- [72] Lee, D. D. and Seung, H. S., “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.
- [73] Lee, D. D. and Seung, H. S., “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems 13*, pp. 556–562, MIT Press, 2001.
- [74] Li, D. and Levinson, S. E., “A Bayes-rule based hierarchical system for binaural sound source localization,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 521–524, 2003.
- [75] Liao, X. and Carin, L., “Constrained independent component analysis of DNA microarray signals,” in *Workshop on Genomic Signal Processing and Statistics*, (Raleigh, NC), October 2002.

- [76] LIAO, X. and CARIN, L., “A new algorithm for independent component analysis with or without constraints,” in *Proceedings of the IEEE Sensor Array and Multichannel Signal Processing Workshop*, (Rosslyn, VA), pp. 413–417, August 2002.
- [77] LOU, S.-T. and ZHANG, X.-D., “Blind source separation for changing source number: A neural network approach with a variable structure,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (San Diego, CA), pp. 384–389, December 2001.
- [78] LU, W. and RAJAPAKSE, J. C., “Constrained independent component analysis,” in *Advances in Neural Information Processing Systems 13*, pp. 570–576, MIT Press, 2001.
- [79] LU, W. and RAJAPAKSE, J. C., “ICA with reference,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (San Diego, CA), December 2001.
- [80] MARTIN, K. D., “Automatic transcription of simple polyphonic music,” Tech. Rep. 399, Media Laboratory Perceptual Computing Section, Massachusetts Institute of Technology, Cambridge, Massachusetts, December 1996.
- [81] MARTIN, K. D., “A blackboard system for automatic transcription of simple polyphonic music,” Tech. Rep. 385, Media Laboratory Perceptual Computing Section, Massachusetts Institute of Technology, Cambridge, Massachusetts, July 1996.
- [82] MASTER, A. S., “Bayesian two source modeling for separation of N sources from stereo signals,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Montreal, Canada), pp. 281–284, May 2004.
- [83] MATSUOKA, K., OHYA, M., and KAWAMOTO, M., “A neural net for blind separation of nonstationary signals,” *Neural Networks*, vol. 8, no. 3, pp. 411–419, 1995.
- [84] MICHEL, P., TOURNERET, J.-Y., and DJURIĆ, P. M., “On-line model selection of nonstationary time series using Gerschgorin disks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Salt Lake City, UT), pp. 3189–3192, May 2001.
- [85] NGUYEN, L.-T., BELOUCHRANI, A., ABED-MERAIM, K., and BOASHASH, B., “Separating more sources than sensors using time-frequency distributions,” in *Proceedings of the International Symposium on Signal Processing and its Applications*, (Kuala Lumpur, Malaysia), pp. 583–586, August 2001.
- [86] O’GRADY, P. D. and PEARLMUTTER, B. A., “Convolutional non-negative matrix factorisation with sparseness constraint,” in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing*, September 2006.
- [87] PARASCHIV-IONESCU, A., JUTTEN, C., and IONESCU, A. M., “Estimation of the source number using array discrete wavelet transform,” in *Proceedings of the IEEE Annual*

Conference of the Industrial Electronics Society, vol. 2, (Sevilla, Spain), pp. 1520–1525, November 2002.

- [88] PARRY, R. M. and ESSA, I., “Blind source separation using repetitive structure,” in *Proceedings of International Conference on Digital Audio Effects*, (Madrid, Spain), pp. 143–148, September 2005.
- [89] PARRY, R. M. and ESSA, I., “Estimating the spatial position of spectral components in audio,” in *Independent Component Analysis and Blind Signal Separation*, vol. 3889 of *Lecture Notes in Computer Science (LNCS)*, (Charleston, SC), pp. 666–673, Springer, March 2006.
- [90] PARRY, R. M. and ESSA, I., “Source detection using repetitive structure,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Toulouse, France), pp. 1093–1096, May 2006.
- [91] PARRY, R. M. and ESSA, I., “Incorporating phase information for source separation via spectrogram factorization,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Honolulu, HI), April 2007.
- [92] PARRY, R. M. and ESSA, I., “Phase-aware non-negative spectrogram factorization,” in *Independent Component Analysis and Signal Separation*, vol. 4666 of *Lecture Notes in Computer Science (LNCS)*, (London), pp. 536–543, Springer, September 2007.
- [93] PHAM, D.-T. and CARDOSO, J.-F., “Blind separation of instantaneous mixtures of nonstationary sources,” *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1837–1848, 2001.
- [94] PHAM, D.-T., SERVIERE, C., and BOUMARAF, H., “Blind separation of convolutive audio mixtures using nonstationarity,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Nara, Japan), pp. 975–980, April 2003.
- [95] PLUMBLEY, M. D., “Algorithms for non-negative independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 14, no. 3, pp. 534–543, 2003.
- [96] PLUMBLEY, M. D., ABDALLAH, S. A., BELLO, J. P., DAVIES, M. E., MONTI, G., and SANDLER, M. B., “Automatic music transcription and audio source separation,” *Cybernetics and Systems*, vol. 33, no. 6, pp. 603–627, 2002.
- [97] RAJ, B., SINGH, R., and SMARAGDIS, P., “Recognizing speech from simultaneous speakers,” in *Eurospeech*, September 2005.
- [98] REYES-GOMEZ, M., JOJIC, B., and ELLIS, D. P. W., “Deformable spectrograms,” in *Artificial Intelligence and Statistics*, (Barbados), January 2005.
- [99] RICKARD, S., BALAN, R., and ROSCA, R., “Blind source separation based on space-time-frequency diversity,” in *International Symposium on Independent Component Analysis and Blind Source Separation*, (Nara, Japan), pp. 493–498, April 2003.

- [100] ROADS, C., *The Computer Music Tutorial*. Cambridge, MA: MIT Press, 1996.
- [101] ROWEIS, S. T., “One microphone source separation,” in *Advances in Neural Information Processing Systems 13*, pp. 793–799, MIT Press, 2001.
- [102] SAKURABA, Y., KITAHARA, T., and OKUNO, J. G., “Comparing features for forming music streams in automatic music transcription,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Montreal, Canada), pp. 273–276, May 2004.
- [103] SANCHIS, J. M., CASTELLS, F., and RIETA, J. J., “Convolutional acoustic mixtures approximation to an instantaneous model using a stereo boundary microphone configuration,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Granada, Spain), pp. 816–823, September 2004.
- [104] SANCHIS, J. M. and RIETA, J. J., “Computational cost reduction using coincident boundary microphones for convolutional blind signal separation,” *IEEE Electronics Letters*, vol. 41, pp. 374–376, March 2005.
- [105] SCHEIRER, E. D., “Bregman’s chimeras: music perception as auditory scene analysis,” in *Proceedings of the International Conference of Music Perception and Cognition*, (Montreal, Canada), August 1996.
- [106] SCHMIDT, M. N. and OLSSON, R. K., “Single-channel speech separation using sparse non-negative matrix factorization,” in *Proceedings of Interspeech*, September 2006.
- [107] SLANEY, M., NAAR, D., and LYON, R. E., “Auditory model inversion for sound separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Adelaide, SA Australia), pp. 77–80, April 1994.
- [108] SMARAGDIS, P., “Information theoretic approaches to source separation,” Master’s thesis, MAS Department, Massachusetts Institute of Technology, 1997.
- [109] SMARAGDIS, P., “Blind separation of convolved mixtures in the frequency domain,” in *International Workshop on Independence and Artificial Neural Networks*, (Tenerife, Spain), February 1998.
- [110] SMARAGDIS, P., *Redundancy Reduction for Computational Audition, a Unifying Approach*. PhD thesis, MAS Department, Massachusetts Institute of Technology, 2001.
- [111] SMARAGDIS, P., “Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Granada, Spain), pp. 494–499, September 2004.
- [112] SMARAGDIS, P. and BROWN, J. C., “Non-negative matrix factorization for polyphonic music transcription,” in *Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), pp. 177–180, October 2003.

- [113] STONE, J. V. and PORRILL, J., “Undercomplete independent component analysis for signal separation and dimension reduction,” tech. rep., Department of Psychology, University of Sheffield, Sheffield, England, October 1997.
- [114] TORKKOLA, K., “Blind separation for audio signals – are we there yet?,” in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, (Aussois, France), pp. 239–244, January 1999.
- [115] VIRTANEN, T., “Separation of sound sources by convolutive sparse coding,” in *ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing*, 2004.
- [116] VIRTANEN, T., “Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1066–1074, March 2007.
- [117] VIRTANEN, T. and KLAPURI, A., “Separation of harmonic sounds using linear models for the overtone series,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, (Orlando, FL), pp. 1757–1760, May 2002.
- [118] WANG, B. and PLUMBLEY, M. D., “Musical audio stream separation by non-negative matrix factorization,” in *Proceedings of the DMRN Summer Conference*, July 2005.
- [119] WANG, B. and PLUMBLEY, M. D., “Investigating single-channel audio source separation methods based on non-negative matrix factorization,” in *ICA Research Network International Workshop*, pp. 17–20, September 2006.
- [120] WU, H.-T., YANG, J.-F., and CHEN, F.-K., “Source number estimator using Gerschgorin disks,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, (Adelaide, SA Australia), pp. 261–264, April 1994.
- [121] WU, H.-T., YANG, J.-F., and CHEN, F.-K., “Source number estimators using transformed Gerschgorin radii,” *IEEE Transactions on Signal Processing*, vol. 43, no. 6, pp. 1325–1333, 1995.
- [122] YAMAMOTO, K., ASANO, F., VAN ROOIJEN, W. F. G., LING, E. Y. L., YAMADA, T., and KITAWAKI, N., “Estimation of the number of sound sources using support vector machines and its application to sound source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 485–488, 2003.
- [123] YE, J.-M., ZHU, X.-L., and ZHANG, X.-D., “Adaptive blind separation with an unknown number of sources,” *Neural Computation*, vol. 16, pp. 1641–1660, 2004.
- [124] YILMAZ, O. and RICKARD, S., “Blind separation of speech mixtures via time-frequency masking,” *IEEE Transactions on Signal Processing*, vol. 52, pp. 1830–1847, July 2004.