Musical Source Separation by Coherent Frequency Modulation Cues

Elliot Creager

Department of Music Research
Schulich School of Music
McGill University
Montreal, Canada

December 2015

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of the Arts.

2015/12/11

Abstract

This thesis explores the extraction of vibrato sounds from monaural excerpts of polyphonic music using the coherent frequency modulation (CFM) of component partials as a grouping cue. Nonnegative Matrix Factorization (NMF) (Lee and Seung 1999) is currently a popular tool for musical source separation (Wang and Plumbley 2005), since it can provide a low-rank approximate factorization of the magnitude spectrogram of the analyzed sound, where the factors can be interpreted as the spectral templates and temporal activations of the notes contributing to the recording. However, NMF implicitly models each source as having a fixed spectral template and is thus ill-suited to the analysis of vibrato sounds, which are characterized by slowly varying frequency and amplitude modulations.

We first propose a useful signal parameter, expressed simply as the local ratio of frequency slope-to-frequency for a component partial, which can be extracted as a feature from an excerpt of polyphonic music via non-stationary sinusoidal model analysis (Smith and Serra 1987) followed by parameter estimation of each component sinusoid using the Distributed Derivative Method (Betser 2009). Two source separation schemes are proposed which utilize this extracted feature. The first, which we call Partial Grouping by Coherent Frequency Modulation (PG-CFM), directly employs this feature in the grouping of the partials tracked by the sinusoidal model analysis. The second, which we call Nonnegative Matrix Factorization with Coherent Frequency Modulation constraints (NMF-CFM), unifies the NMF generative model with the sinusoidal model analysis via a restructuring of the NMF decomposition. Additionally, the NMF-CFM decomposition is constrained by way of penalty terms that encourage CFM among same-source partials in the estimated sources.

Two experiments were conducted to investigate the performance of PG-CFM technique. The first investigates the behavior of the grouped partials with respect to the PG-CFM algorithm parameters. The second experiment compares the performance of PG-CFM against NMF, first for a class of synthetic vibrato sounds, and then for a small dataset of vocal audio with the vibrato effect. An application of NMF-CFM to the analysis of vibrato singing voice sounds is also provided.

These preliminary results suggest a benefit of using a CFM-based source model in a separation task where vibrato voices or instruments are the target sources.

Resumé

L'utilisation de la factorisation en matrices non-négatives (NMF) (Lee et Seung 1999) est devenue très populaire dans le cadre de la séparation de sources sonores (Wang et Plumbley 2005), car elle fournit une approximation par factorisation de rang faible des spectrogrammes d'amplitude d'un extrait musical, dont les facteurs peuvent être interprétés respectivement comme des références spectrales et des activations temporelles des notes contribuant à l'extrait musical. Cependant, la NMF considère implicitement chaque source comme étant à contenu spectral fixe, et est donc mal adaptée à l'analyse des sons vibrés qui se caractérisent par une modulation lente en fréquence et en amplitude de leurs partiels.

Dans cette thèse nous proposons tout d'abord un descripteur de signal utile dans le cadre de la séparation de sources et qui s'exprime simplement comme le rapport local entre la pente de fréquence et la fréquence d'un partiel donné, ce descripteur pouvant être estimé à partir d'un extrait de musique polyphonique selon une méthode d'analyse additive (Smith et Serra 1987), suivie d'une estimation de paramètres de chacune des sinusoïdes par la méthode dite de la distribution dérivée (DDM) (Betser 2009). Deux stratégies de séparation de sources reposant sur cette caractéristique sont ensuite proposées. La première, que nous appelons regroupement de partiels par modulations de fréquence cohérentes (PG-CFM), emploie directement ce descripteur pour regrouper les partiels préalablement extraits par une technique d'analyse sinusoïdale. La seconde, que nous appelons factorisation de matrice non-négative sous contrainte de modulations de fréquence cohérentes (NMF-CFM), unifie le modèle générique de la NMF avec le modèle de représentation sinusoïdal des sons, via une restructuration de la technique NMF de base. De plus, nous ajoutons des contraintes à la décomposition NMF-CFM sous forme de termes de pénalité qui encouragent la CFM entre partiels de même source.

Deux expériences ont été menées afin d'évaluer la performance de la technique PG-CFM. La première expérience évalue le comportement du regroupement de partiels en fonction de la paramétrisation de l'algorithme PG-CFM. La seconde expérience compare les performances de la PG-CFM et de la NMF, d'abord pour une classe de signaux synthétiques vibrés, puis sur une sous-classe de signaux vocaux réels vibrés. Une application de la NMF-CFM à l'analyse de signaux de voix chantées vibrées est aussi menée. Ces résultats préliminaires tendent à montrer l'effet bénéfique de l'introduction du descripteur CFM dans les tâches de séparation de sources audio lorsque le mélange polyphonique est constitué de

sources dont chacune est soumise à une modulation de fréquence cohérente et en particulier à une modulation de type vibrato.

Acknowledgments

I wish to express my sincerest gratitude to Professor Philippe Depalle for his mentorship and outstanding instruction beginning with first semester at McGill, for his continued support and guidance throughout the course of this research project, and for his assistance in translating the abstract. I would also like to thank Professor Roland Badeau, a visiting researcher at the Centre for Interdisciplinary Research in Music, Media, and Technology, who made invaluable contributions to the project in its final months.

I would like to acknowledge Helene Drouin and Darryl Cameron for their administrative and technical assistance in the realization of this thesis. I recognize the financial support provided by the Schulich School of Music by way of scholarships and fellowships, which facilitated my graduate studies at McGill University.

I am indebted to all of my mentors, here and at previous institutions, who have provided personal encouragement, demonstrated good research practices, and fostered qualities of inquisitiveness and persistence in my work. I wish to thank, in no particular order, David Wingate, Noah Stein, Joseph Rovan, Harvey Silverman, and David Durfee.

Lastly, I want to thank to my parents Angela and Bill Creager for their love and support in all of my endeavors.

Contents

List of Figures

List of Tables

List of Acronyms

DFT	Discrete Fourier Transform
STFT	Short-time Fourier transform
ISTFT	Inverse STFT
FFT	Fast Fourier transform
NMF	Nonnegative matrix factorization
CNMF	Complex NMF
SED	Squared Euclidean distance
KL	Generalized Kullback-Liebler divergence
IS	Itakura-Saito divergence
CASA	Computational Auditory Scene Analysis
SMA	Spectral modeling analysis
SMS	Spectral modeling synthesis
PV	Phase vocoder
DDM	Distribution derivative method
CFM	Coherent frequency modulation
PG-CFM	Partial grouping by CFM
NMF-CFM	NMF with CFM constraints
SDR	Source-to-distortion ratio
SIR	Source-to-interference ratio
SAR	Source-to-artifacts ratio
QP	Quadratic program
EM	Expectation-Maximization
MM	Majorization-Minimization

List of Selected Symbols

Symbol	Domain	Meaning
R	$\mathbb{Z}$	Number of sources in the mixture model
K	$\mathbb{Z}$	Number of component partials in the source model
F	$\mathbb{Z}$	Number of non-redundant DFT bins
N	$\mathbb{Z}$	Number of short-time hops
$\mathbb{X}$	$\mathbb{C}^{F\times N}$	STFT matrix
$\mathbf{X}$	$\mathbb{R}_{\geq 0}^{F\times N}$	STFT magnitudes matrix
$X(f,n)$	$\mathbb{R}_{\geq 0}$	Entry of $\mathbf{X}$
$\mathbf{W}$	$\mathbb{R}_{\geq 0}^{F\times R}$	NMF spectral templates matrix
$w^r(f)$	$\mathbb{R}_{\geq 0}$	Entry of $\mathbf{W}$
$\mathbf{H}$	$\mathbb{R}_{\geq 0}^{R\times N}$	NMF activations matrix
$h^r(n)$	$\mathbb{R}_{\geq 0}$	Entry of $\mathbf{H}$
$f_p(m)$	$\mathbb{R}$	p-th partial instantaneous frequency at time m
$\xi_p(m)$	$\mathbb{R}$	p-th partial local frequency slope at time m
$\boldsymbol{\Upsilon}$	$\mathbb{R}^{K\times N}$	Estimated local frequency-slope-to-frequency ratios matrix
$\Upsilon(k,n)$	$\mathbb{R}$	Entry of $\boldsymbol{\Upsilon}$
$\boldsymbol{p}$	$\mathbb{R}_{\geq 0}^{R\times K\times N}$	PG-CFM likelihoods tensor
$p^r(k,n)$	$\mathbb{R}_{\geq 0}$	Entry of $\boldsymbol{p}$; likelihood that source r produced tracked partial k at hop n
$\boldsymbol{\upsilon}$	$\mathbb{R}^{R\times N}$	PG-CFM algorithm-estimated local frequency-slope-to-frequency ratios
$\upsilon^r(n)$	$\mathbb{R}$	Entry of $\boldsymbol{\upsilon}$; estimated frequency-slope-to-frequency ratio for source r at hop n

Chapter 1

Introduction

We can understand a recording of polyphonic music as a mixture of instrumental sources. Musical source separation describes the "unmixing" of such a recording to produce an isolated track for each of the instruments present. This problem is of great interest in the Music Technology community, as a robust solution would enable the design of more capable software tools for composers and sound engineers working with recorded samples. Additionally, related tasks such as polyphonic transcription, content-based indexing, and audio coding, benefit from the individual analysis of each separated source (Lyon 2010).

This thesis concerns the analysis of monaural musical recordings in order to extract time-domain audio estimates for the individual notes present, where each note has an independent frequency vibrato effect. Two algorithms are proposed to this end, each relying on the extraction of features that represent the frequency modulation on each component partial in a non-stationary sinusoidal model. The extracted feature is shown to be common to all same-source partials and is thus useful in the separation. The first algorithm, which we call Partial Grouping by Coherent Frequency Modulation (PG-CFM), groups the partials directly by estimating feature vectors, one for each source, which most likely explain the observed data. The second, which we call Nonnegative Matrix Factorization with Coherent Frequency Modulation constraints (NMF-CFM), is an extension of Nonnegative Matrix Factorization (NMF) (Lee and Seung 1999), which correctly decomposes the magnitude spectrogram for some musical source separation tasks (Wang and Plumbley 2005) but is ill-suited to the analysis of vibrato sounds (Barker and Virtanen 2013).

1.1 Motivation

This work is motivated on the algorithmic side by the data-driven decomposition of Non-negative matrix factorization (NMF) (Lee and Seung 1999) (Paatero and Tapper 1994). NMF provides an efficient model which approximates the observed nonnegative data by means of a nonnegative factorization, where individual factor-pairs in the model can be meaningfully interpreted in a variety of contexts (Wang and Zhang 2012). The decomposition provided by NMF is fundamentally data-driven, as no a priori knowledge of the signal characteristics are required, and is often referred to as a *blind* source separation technique for this reason.

Although NMF has successfully been applied to musical source separation tasks via a factorization of the spectrogram of the musical mixture (Wang and Plumbley 2005) (Virtanen 2007), it models vibrato sounds—characterized by slowly-varying amplitude and frequency modulations (Maher and Beauchamp 1990)—incorrectly in the sense that the factor-pairs do not capture the characteristic modulation (Li, Woodruff, and Wang 2009). For the analysis of this class of sounds, we appeal to sinusoidal signal modeling (Smith and Serra 1987) (McAulay and Quatieri 1986) and the perceptual theory of Auditory Scene Analysis (Bregman 1990), which respectively provide means for a proper analysis of the subaudible modulation and subsequent perceptual grouping of partials according to the analysis.

It is certainly worth recognizing the large body research in audio analysis—including on the topic of source separation—that draws inspiration from Auditory Scene Analysis, and is called Computational Auditory Scene Analysis (CASA) (Wang and Brown 2006). Labeling a source separation method as CASA-based typically implies the use of (a) a physically-inspired signal representation, e.g., the cochlear model of (Lyon 1982), and (b) some high-level segmenting and grouping scheme which results in a so-called "auditory stream", e.g., a talker or instrument, present in the recording. The algorithms presented in this thesis are not *CASA-based* in this regard, since they work on a lower-level to resolve overlapping musical notes from short excerpts of music and use only one of the grouping cues presented in Auditory Scene Analysis.

1.2 Scope

Broadly speaking, source separation describes the task of isolating component signals contributing to an observed mixture. We focus here on the problem of single-channel source separation, where the observed mixture is a recording of musical sounds. For the sake of brevity, we avoid an exhaustive review of the broader source separation literature. Instead, we focus on the problem at hand, while providing the necessary background information and abridged history relating to the monaural musical source separation problem as needed in the subsequent chapters. For a more comprehensive study of the source separation problem in general, including a taxonomy of possible problems according to their specifications, the interested reader should refer to (Jutten and Comon 2010), or (Vincent and Deville 2010) for the chapter on audio applications.

The purpose of these investigations is to produce an appropriate signal model and accompanying algorithm for the extraction of sounds characterized by a slowly-varying frequency modulation, e.g., singing voice with vibrato. This represents a significant yet small subtask associated with the more broad and difficult problem of developing a robust musical source separation system capable of analyzing entire pieces of music. Correspondingly, throughout this thesis the source separation problem is constrained along several dimensions in order to permit a thorough investigation of the novel techniques proposed. For example, identification of the number of sources present in a mixture is a difficult problem for which no robust solution currently exists (Virtanen 2007). As such, we assume the number of sources to be known a priori in the experiments and applications presented.

The algorithms presented are evaluated within a limited scope, namely, the analysis of short (one to three seconds) monaural recordings produced as the linear mixture of two sources, where a "source" is defined as a single note, which is oftentimes synthetically produced as a simple waveform (e.g., square wave) with a vibrato effect added via a coherent frequency modulation on the component partials used in the additive synthesis. The use of well-calibrated synthetic signals facilitates a thorough investigation of the algorithm behaviors. By generating signals synthetically, we have access to the "ground truth" signal parameters such as instantaneous amplitude and frequency of the component partials. Access to ground truth is especially relevant to the novel source separation algorithms presented, since several levels of feature extraction are carried out prior to the separation and source estimation. In the experiments presented we use the features extracted from the

synthetic signal mixtures in the separation, but hold the ground truth signal parameters aside for evaluation as needed. Use of synthetic signals also permits the quick prototyping and evaluation using large datasets of vibrato sounds, possibly with random parameters.

In general, however, we should be wary of experimental evaluations using data which fit the generative model by design. To this point, evaluation of mixtures of vibrato vocal sounds are provided that demonstrate a relevance of the proposed feature extraction and separation techniques to the analysis of natural vibrato sounds.

The analysis of vibrato source separation on very short musical excerpts does seem appropriate as the effect is generally applied per-note. However, the unification of the vibrato-specific separation techniques presented in this thesis would require additional work. In particular, the coherent frequency modulation cues are not appropriate for the analysis of sounds with stationary fundamental frequency since they rely on estimates of frequency slope which are zero for this class of sounds. The results presented in here hopefully represent a contribution towards the development of a more complete and robust source separation scheme which appropriately handles vibrato sources.

Throughout this thesis we take "vibrato" to mean frequency vibrato, since we focus on the use of local frequency slope estimates as a grouping cue in the source separation. It should be noted that elsewhere in the literature, vibrato typically refers to the combined effect of subaudible modulations in both frequency and amplitude (also called tremolo). The vibrato effect may be characterized, depending on the instrumental source, by a predominance of either frequency modulations (e.g., singing voice, stringed instruments) or amplitude modulations (e.g., brass and woodwind instruments) (Verfaille, Guastavino, and Depalle 2005). While we focus on the analysis of vibrato sounds in this thesis, the underlying model can be applied more generally to the analysis of any sound with a coherent frequency modulation across partials, e.g., glissando, pitch bending.

1.3 Signal representations

The source separation of audio signals is rarely carried out in the time domain. Rather, the mixture is typically processed in some other domain following the application of an invertible transformation to the input signal or analysis by a signal model for which a corresponding synthesis scheme exists. Two signal representations are of particular interest in this thesis. The first is given by the short-time Fourier transform (STFT) of the analyzed

sound, which can be interpreted as a sliding-window Fourier transform or as the output from a bank of equispaced band-pass filters (Allen and Rabiner 1977). This complex-valued matrix encodes both magnitude and phase information at the output of each filter. NMF can analyze only the STFT magnitudes (called the spectrogram) due to the nonnegativity constraint. Thus the STFT phase information is discarded prior to the NMF analysis. This step, however, prohibits the NMF from properly analyzing vibrato sounds, since frequency modulations are represented by the phase of the STFT.

In this research we appeal to sinusoidal model analysis to capture the frequency modulation information. This signal representation more directly captures the frequency modulation in the analyzed sound by modeling it as a sum of sinusoids with slowly varying amplitudes and phase, and has been studied extensively for the analysis of a single source of both a musical and speech nature (Serra and Smith 1990) (McAulay and Quatieri 1986). While this signal model is appropriate for most pitched musical sounds, percussive sounds are notably ill-fit by this model.

1.4 Evaluation

The quantitative evaluation of algorithm performance remains a key issue in the research community, since the assessment of perceived quality of an estimated source is fundamentally a subjective task, and the subjective assessment of results by human listeners is often prohibitively expensive and discourages an easy comparison and cross validation of results across groups of researchers. Moreover, estimated sources can suffer from distortions of several varieties, and perfect separation is "rarely achieved in practice" (Vincent and Deville 2010). The general paradigm for objective evaluation of performance involves a comparison of the estimated source to the actual source by some metric which quantifies levels of distortion in the estimated source. This mode of evaluation necessitates a possession of the source signals in isolation and thus favors the use of simulated mixtures in experimentation. We use simulated mixtures in the experiments and applications presented to facilitate quantitative performance evaluations, which reflects a common practice in the literature (Vincent and Deville 2010). We use the metrics proposed by (Vincent, Gribonval, and Févotte 2006), termed `BSS_EVAL`, which were proposed to assess the variety of distortion measures characteristic of source separation algorithms. `BSS_EVAL` is widely used in the contemporary source separation literature, having been used to evaluate recent

separation challenges in source separation challenges focused on both speech and musical signals (Vincent, Watanabe, Barker, Roux, Nesta, and Matassoni 2013) (Yen, Luo, and Chi 2014), so the use of these metrics should yield results which are easy to interpret and compare against competing techniques. Details of these metrics are provided alongside the experimental procedure in section 4.3.5.

In evaluating the performance of PG-CFM and NMF-CFM, large experiments were performed on simulated mixtures of synthetic signals. The use of synthetic signals provides the sufficient control over the experimental conditions for the investigation the behavior of the proposed algorithms with respect to their parameters. Moreover, a large and well-organized database of frequency vibrato sounds was unavailable. However, an evaluation of the feature extraction and source separation techniques described is provided using a small dataset of vibrato singing voice sounds in order to show that the coherent frequency modulation model is relevant to the analysis of natural sounds. This database, which is discussed in further detail in section 4.3.2, was assembled by simulating the mixture of pairs of vibrato vocal excerpts retrieved from several existing databases of musical recordings (not specific to frequency vibrato).

1.5 Outline

Chapter 2 provides background on NMF and its application to musical source separation, including the presentation of several basic algorithms from the literature and a discussion of possible extensions. Chapter 3 discusses the analysis of musical sources by a non-stationary sinusoidal model, and provides some background relevant to the perceptual motivations of the research. Additionally, a source separation algorithm for grouping component partials according to coherent frequency modulation cues is proposed. Chapters 4 and 5 present the procedure and results from experiments carried out to evaluate the performance of the proposed partial grouping algorithm in the analysis of vibrato sound mixtures. Synthetic vibrato sounds and excerpts of vibrato vocal sounds are both considered in the evaluation. Chapters 6 proposes an algorithm which unifies the NMF with the partial grouping method and presents an application to the analysis of synthetic vibrato signals. Finally, chapter 7 summarizes the thesis, provides concluding remarks, and offers suggestions for future research. Details of the analysis and synthesis tools used in the experiments are provided in appendix A. Appendices B and C provide a derivation of the iterative updates for the

algorithms presented, and shows an equivalence of several optimization problems used in the PG-CFM and NMF-CFM to well-known optimization problems which permit the use of efficient solvers.

Chapter 2

Nonnegative matrix factorization

Nonnegative Matrix Factorization (NMF) is a dimensionality reduction technique motivated by the desire to represent observed nonnegative data approximately but efficiently as a linear mixture of nonnegative elements, where each element has a low-rank factorization. This amounts to a search for approximate matrix factors due to linearity in the matrix multiplication and mixture model. An individual element contributing to the approximation may be interpreted in isolation due to a nonnegativity constraint on its factors. NMF is often called a *parts-based* decomposition in recognition of this interpretability, which is often meaningful in the sense that each element in the approximate factorization seems to represent a *part* or *source* contributing to the observed data. This apparent representation is the key asset of NMF as a data reduction tool, as the approximate factors themselves are computed *blindly*, i.e., without a priori knowledge of the data (aside from their nonnegativity). NMF provides a simultaneously economical and meaningful signal representation in a variety of applications where nonnegative data is analyzed and has recently become a popular analysis tool in many fields of computational science including musical source separation.

2.1 History

NMF was popularized by (Lee and Seung 1999) via an application to a facial recognition task where the resulting nonnegative components were shown to be more readily inter-

pretable than those yielded by established dimensionality reduction techniques.[1] NMF was used to analyze 2,049 low-resolution greyscale images of faces in order to produce a (low-rank factorizable) basis of 49 elements. Both the observed image pixels and the NMF components are nonnegative, so individual NMF basis elements permit the interpretation as parts (e.g., a mouth or pair of eyes) contributing to the analyzed image. This seminal paper also contributed algorithms for deriving approximate factors via iterative multiplicative updates, which are discussed in section 2.7. Theoretical developments in matrix factorization with a nonnegativity constraint were introduced several years earlier (Paatero and Tapper 1994). NMF has also been extended to analyze data in higher dimensions as Nonnegative Tensor Factorization (NTF) (Cichocki, Zdunek, and Amari 2008).

(Cichocki, Zdunek, Phan, and Amari 2009) provides a more detailed history of NMF alongside a comparison to established data analysis tools such as Independent Component Analysis (ICA) (Comon 1994). (Wang and Zhang 2012) presents a comprehensive review of recent extensions to the basic NMF.

2.2 Audio applications

In audio processing, the factor-pairs resulting from the NMF decomposition of a spectrogram can be interpreted as pairs of spectral templates and temporal envelopes that approximately produce the spectrogram of the analyzed sound. In the analysis of a music recording, the elements in the modeled mixture correspond to the notes which constitute the piece or excerpt, each of which is characterized spectrally by pitch and timbre, and temporally by onset and offset. (Plumbley, Abdallah, Bello, Davies, Monti, and Sandler 2002) and (Smaragdis and Brown 2003) identified the mid-level parts-based signal representation provided by NMF as useful in a music transcription task, the goal of which is to determine names and durations of notes present in a music recording.

(Wang and Plumbley 2005) applied NMF analysis to a source separation task, where instrumental sources in the recording must be estimated in the time domain. The time-domain estimation of sources was identified as a key challenge since the nonnegative NMF components represent spectrograms of the sources and lack phase information required for

[1] NMF was compared with both Principal Component Analysis (PCA) (Hotelling 1933) and Vector Quantization (VQ) (Gray 1984) reduced to the same rank, neither of which imposes a nonnegativity constraint.

an inverse short-time Fourier transform reconstruction. Estimation of missing phase information from a magnitude spectrogram has been the subject of research in other contexts (Griffin and Lim 1984) (Achan, Roweis, and Frey 2003), particularly in the single-source case, but (Wang and Plumbley 2005) used a basic masking approach which assumes only one dominant source per time-frequency tile. This masking scheme implicitly assumes that the sum of the spectrograms of the sources is the spectrogram of the sum of the sources (equivalent to assuming no time-frequency overlap between sources), which is known to be violated by many real audio mixtures, particularly with musical mixtures (Li, Woodruff, and Wang 2009). (Kamoeka, Ono, Kashino, and Sagayama 2009) addressed this via the inclusion of a the source phases in a NMF-like decomposition (although strictly not a factorization) termed Complex Nonnegative Factorization (CNMF). (Bronson and Depalle 2014) extended CNMF via a phase model for harmonic sources. (Badeau and Plumbley 2014) applied a so called "high-resolution" NMF model, which accounts for both the STFT phases and local correlations within each frequency band, to the estimation of sources within a convolutive mixture model.

2.3 Problem formulation

The nonnegative matrix $\mathbf{X} \in \mathbb{R}_{\geq 0}^{F \times N}$ is approximated as the product of two nonnegative factor matrices as

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{WH} = \sum_{r=1}^{R} \mathbf{w}_r \mathbf{h}_r^T \tag{2.1}$$

where the $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_R) \in \mathbb{R}_{\geq 0}^{F \times R}$ is the basis matrix whose columns are the basis vectors and $\mathbf{H} = (\mathbf{h}_1, ..., \mathbf{h}_R)^T \in \mathbb{R}_{\geq 0}^{R \times N}$ is the activation matrix whose rows are the activations vectors. $\mathbf{W}$ and $\mathbf{H}$ are hereafter referred to as the NMF *factors* for convenience and consistency with the literature, despite the fact that they only approximately factorize the observed data. R is chosen so that the factorized representation is smaller then the observed data, i.e., $F \times R + R \times N \ll F \times N$. The value of R is an important problem-specific design consideration as it sets the number of basis vectors in the approximation, which should ideally match the dimension of the subspace in which the observed data lie so that individual basis vectors can be meaningfully interpreted.

$\mathbf{W}$ and $\mathbf{H}$ are chosen to minimize some scalar cost function expressed as sum of scalar costs per element, with an element-wise nonnegativity constraint on the factors, expressed

as

$$\begin{aligned} \underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad & D(\mathbf{X}|\mathbf{WH}) = \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} d(\mathbf{X}_{f,n}|(\mathbf{WH})_{f,n}) \\ \text{subject to} \quad & \mathbf{W},\mathbf{H} \geq 0. \end{aligned} \tag{2.2}$$

$\mathbf{W}$ and $\mathbf{H}$ are often referred to in the literature as the factors of the observed data although their product is in fact the low-rank approximation $\hat{\mathbf{X}}$.

For the analysis of audio mixtures $\mathbf{X}$ is chosen as some nonnegative time-frequency representation of the input audio. The constant-Q transform (Brown 1991) has also been examined for NMF-based audio analysis (Smaragdis, Raj, and Shashanka 2008), but is less desirable for source separation as the resynthesis of time-domain audio from a factored constant-Q transform is not straightforward. In this thesis we approximately factorize the spectrogram (STFT magnitudes) of the observed data, i.e.,

$$\mathbf{X}_{f,n} = \left|\mathbb{X}_{f,n}\right| \tag{2.3}$$

where the STFT $\mathbb{X}_{f,n}$ is defined in appendix A.1.

We interpret the r-th outer product in the sum of (2.1) as the spectrogram of the r-th *part* of the observed mixture spectrogram $\mathbf{X}$. $\mathbf{w}_r$ and $\mathbf{h}_r$ then correspond to the spectral template and temporal activation corresponding to the r-th part.

2.4 Separation

A reconstruction stage is required to estimate time-domain audio $\hat{\mathbf{x}}^{(r)}$ for source r given its NMF decomposition $\hat{\mathbf{X}}^{(r)} = \mathbf{w}_r\mathbf{h}_r^T$, which lacks phase information. A common approach is to apply a time-frequency mask derived from $\hat{\mathbf{X}}^{(r)}$ to the input STFT $\mathbb{X}$. The approximate STFT for source r is then expressed as

$$\hat{\mathbb{X}}^{(r)} = \frac{\hat{\mathbf{X}}^{(r)}}{\hat{\mathbf{X}}} \circ \mathbb{X} \tag{2.4}$$

where — and $\circ$ denote element-wise division and multiplication, respectively. The ratios $\frac{\hat{\mathbf{X}}^{(r)}}{\hat{\mathbf{X}}}$ correspond to per-hop Wiener filter gains for each of the estimated sources (Févotte, Bertin, and Durrieu 2009). Source r is then reconstructed in the time-domain by ISTFT

as

$$\hat{\mathbf{x}}^{(r)} = [\text{ISTFT}\{\hat{\mathbb{X}}^{(r)}\}]. \tag{2.5}$$

This separation scheme implicitly assumes a single dominant source within each time-frequency bin (i.e., non-overlapping partials). Musical noise may occur in the separated audio when this assumption is violated.

2.5 Scalar costs

The qualities of the factors solutions to NMF depend on the choice of an element-wise distance or divergence measure $d(\mathbf{X}_{f,n}|\hat{\mathbf{X}}_{f,n})$ in equation 2.1. (Lee, Hill, and Seung 2001) examined two popular choices of scalar cost: the squared Euclidean distance (SED), defined as

$$d_{SED}(a|b) = (a-b)^2, \tag{2.6}$$

and the generalized Kullback-Liebler (KL) divergence, defined as

$$d_{KL}(a|b) = a \log \frac{a}{b} - a + b. \tag{2.7}$$

Another popular choice is the Itakura-Saito (IS) divergence (Févotte, Bertin, and Durrieu 2009).

$$d_{IS}(a|b) = \frac{a}{b} - \log \frac{a}{b} - 1 \tag{2.8}$$

SED-NMF refers to NMF using the SED as the per-element scalar cost. Likewise, KL-NMF and IS-NMF respectively refer to NMF using KL divergence and IS divergence as the per-element scalar cost.

The IS divergence was conceived as measure of fit for comparing power spectra of speech signals (Itakura and Saito 1968), and is thus well-suited to audio applications where the observed data are power spectrograms. Furthermore the IS divergence is scale invariant, a desirable quality in the analysis of audio, which has a large dynamic range and scales logarithmically in perceptual loudness. Note that while each of the measures is nonnegative with a unique minimum at $a = b$, d_{KL} and d_{IS} are not distances in the statistical sense because they are asymmetric and violate the triangle inequality, i.e., $d(a|b+c) \leq d(a|b) + d(a|c)\ \forall\ a, b, c$ does not hold. d_{SED} is derived by squaring the ℓ_2 norm (a true distance metric) for ease of evaluating the derivatives.

$d_{SED}(a|b)$ and $d_{KL}(a|b)$ are each separately convex with respect to a and b. The overall scalar cost $D(\mathbf{X}|\mathbf{WH})$ is thus separately convex in each of the factors, which permits for an optimal solution in each step of the alternating minimization. $d_{IS}(a|b)$, however, is separately non-convex in the individual variables, so each step in the alternating minimization iteration yields a locally optimal solution by Majorization-Minimization (MM). (Févotte, Bertin, and Durrieu 2009) provides such MM algorithmic updates for IS-NMF.

The β-divergence generalizes the three aforementioned cost functions to a single divergence measure, defined as

$$d_\beta(a|b) = \begin{cases} \frac{1}{\beta(\beta-1)}(a^\beta + (\beta-1)b^\beta - \beta a b^{\beta-1}) & \beta \in \mathbb{R}\backslash\{0,1\} \\ a\log\frac{a}{b} - a + b & \beta = 1 \\ \frac{a}{b} - \log\frac{a}{b} - 1 & \beta = 0 \end{cases} . \tag{2.9}$$

where β is a tunable design parameter. NMF using the β-divergence is called β-NMF. This generalization of the scalar cost allows for a flexible algorithm design as the cost function can be tailored to the application at hand or can even be learnt from the data given an appropriate evaluation criterion[2]. The β-divergence is equivalent to SED, KL divergence, and IS divergence in the special cases of β equal to 2, 1, and 0, respectively. The value of β can in fact be understood as affecting the shape of the noise in the NMF implicit generative model, e.g., Gaussian additive noise for $\beta = 2$ and Poisson multiplicative noise for $\beta = 1$. It is interesting to note that the tuning of β controls a tradeoff of influence between high- and low-energy components in the decomposition, with increasing β prioritizing high-energy components and $\beta = 0$ weighting all components equally, i.e., scale-invariance as mentioned above. For applications to audio analysis this can be interpreted as a tradeoff of influence of the low- (e.g., fundamental frequency) and high-frequency (e.g., upper overtones) components, since the spectral energy of musical sounds tends to fall off in the higher registers.

$d_\beta(a|b)$ is separately convex in a and b only for $\beta \in [1, 2]$. (Hennequin, David, and Badeau 2011) proved the β-divergence to be a Bregman divergence, which permits the derivation of an appropriate majorizing auxiliary function for an MM-based β-NMF algorithm for the non-convex regions of β as in (Dhillon and Sra 2006). (Févotte and Idier

[2]Application-specific evaluation criteria are required to determine the optimal value of β as the scalar costs $d_\beta(a|b)$ cannot be directly compared for differing values of β.

2011) provided such an algorithm for β-NMF with MM updates and demonstrated an improvement in convergence speed when compared with multiplicative update and heuristic update β-NMF algorithms.

2.6 Alternating factor updates

The acquisition of a globally optimal solution is intractable in practice as the cost function to be minimized is generally non-convex with respect to both matrix factors. Instead we seek a locally optimal solution by an iterative alternating factor update scheme, where the inner loop first optimizes $\mathbf{W}$ while $\mathbf{H}$ is held fixed, i.e.,

$$\begin{aligned} &\text{choose} && \mathbf{W}^{i+1} \\ &\text{such that} && D(\mathbf{X}|\mathbf{W}^{i+1}\mathbf{H}^{i}) \leq D(\mathbf{X}|\mathbf{W}^{i}\mathbf{H}^{i}) \end{aligned} \tag{2.10}$$

where $\mathbf{W}^i$ and $\mathbf{H}^i$ denote the basis and activation matrices at the i-th iteration. We then in turn update $\mathbf{H}$ while fixing $\mathbf{W}$ to the value from previous computation, i.e.,

$$\begin{aligned} &\text{choose} && \mathbf{H}^{i+1} \\ &\text{such that} && D(\mathbf{X}|\mathbf{W}^{i+1}\mathbf{H}^{i+1}) \leq D(\mathbf{X}|\mathbf{W}^{i+1}\mathbf{H}^{i}) \end{aligned} \tag{2.11}$$

This optimization scheme is motivated by the separate convexity with respect to the individual factors of the several popular cost functions, including those examined in (Lee and Seung 1999). Whenever the cost function is separately convex we can alternatively update each factor to its optimum value given the fixed value of the other factor. The alternating minimization scheme remains applicable to choices of cost function that are non-convex in the individual factors. In this case each optimization step in the inner loop (e.g., finding the optimal factor update $\mathbf{W}^{i+1} \leftarrow \mathbf{W}^i$ given a fixed $\mathbf{H}^i$) returns a locally optimal solution via Majorization-Minimization (MM) given an appropriate convex majorizing auxiliary function as in (Hunter and Lange 2004).

In practice the factors are often updated with a descent method that does not update the independent factor optimally but hopefully provides a sufficient reduction in cost. Descent methods require choice of descent direction and step size. Gradient descent methods choose the descent direction as the gradient of the cost function with respect to the independent factor. A subsequent projection step is required, as updating the factors in the direction of

the gradient descent does not guarantee preservation of their nonnegativity. In the gradient descent paradigm each factor update is inexpensive and the cost decreases monotonically but may converge slowly.

The outer loop of the alternating minimization terminates when the stopping criteria are met, which is typically chosen as a maximum number of iterations (Wang and Zhang 2012), but could be defined alternatively as a minimum scalar cost to be achieved, or more rigorously as the satisfaction of the Karush-Kuhn-Tucker (KKT) conditions, which guarantee the local optimality of a solution and are defined for NMF by (Berry, Browne, Langville, Pauca, and Plemmons 2006).

2.7 Algorithms

NMF algorithms are described by their alternating factor updates (and possibly their initialization). They are assessed in terms of their speed of convergence and cost corresponding to their final solution. An overview of significant NMF algorithms can be found in (Berry, Browne, Langville, Pauca, and Plemmons 2006), where solution techniques from the literature are categorized in terms of their iterative update rules of the model parameters. Several canonical NMF algorithms are discussed here. (Berry, Browne, Langville, Pauca, and Plemmons 2006) offers a more comprehensive review of each of the following category of factor update, including an examination of the KKT conditions for local optimality of solutions and comparison of convergence speeds.

2.7.1 Alternating Least Squares NMF

(Paatero and Tapper 1994) devised SED-NMF as a two-step alternating least-squares problem. Indeed, the SED cost function defined in equation 2.6 is a least-squares problem with one fixed factor and permits a closed from solution. Thus SED-NMF can be solved by alternating least-squares updates to the factors, plus a projection step to ensure nonnegativity. The basis matrix $\mathbf{W}$ is updated as

$$\mathbf{W} \leftarrow [\mathbf{X}\mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}]_+ \tag{2.12}$$

where $[\cdot]_+$ denotes an element-wise nonnegative projection, i.e., projection onto the nonnegative orthant. Likewise, the activations matrix $\mathbf{H}$ is updated as

$$\mathbf{H} \leftarrow [(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{X}]_+. \tag{2.13}$$

The Alternating Least Squares (ALS) framework supports extensions to NMF by the inclusion of certain penalty terms that preserve the separate convexity of the cost in the factors, e.g., the inclusion of a ℓ_2 norm constraint on one factor corresponds to a Tikhonov regularization of the least-squares problem. The regularized SED-NMF of this form may not permit a closed-form factor update as in equations 2.12 and 2.13 but can be solved efficiently as a quadratic program followed by a nonnegative projection (Boyd and Vandenberghe 2009). (Cichocki, Zdunek, and Amari 2008) suggested that the speed of convergence properties of ALS make it better-suited to large-scale problems and nonnegative tensor factorizations.

2.7.2 Projected Gradient Descent NMF

Factors can be updated by additive gradient descent, a numerical optimization technique suitable for non-convex problems whereby the model parameter is updated additively in a direction opposite the local gradient of the scalar cost with respect to the parameter.

The step size is set by a growth factor ϵ. A subsequent projection step is required as the additive gradient descent factor update may produce negative values in the factor. The factors in Projected Gradient Descent NMF (PGD-NMF) are updated as

$$\mathbf{W} \leftarrow \left[\mathbf{W} - \epsilon_{\mathbf{W}} \nabla_{\mathbf{W}} D(\mathbf{X}|\mathbf{W}\mathbf{H})\right]_+, \tag{2.14}$$

and

$$\mathbf{H} \leftarrow \left[\mathbf{H} - \epsilon_{\mathbf{H}} \nabla_{\mathbf{H}} D(\mathbf{X}|\mathbf{W}\mathbf{H})\right]_+ \tag{2.15}$$

where $\epsilon_{\mathbf{W}}$ and $\epsilon_{\mathbf{H}}$ are the step sizes for the factor updates. Linearity of the global cost $D(\mathbf{X}|\hat{\mathbf{X}})$ with respect to the element-wise scalar costs $d(\mathbf{X}_{f,n}|\hat{\mathbf{X}}_{f,n})$ permits element-wise updates of the factors.

The choice of step size is crucial in PGD-NMF as it affects speed of convergence and also the final solution due to the projection step. Convergence to a local minimum $\{\mathbf{W}^*, \mathbf{H}^*\}$ is difficult to prove due to the projection step but is often empirically observed.

2.7.3 Multiplicative Update NMF

Multiplicative update NMF (MU-NMF) algorithms have been pervasive since their introduction in (Lee and Seung 1999) and (Lee, Hill, and Seung 2001), likely due to the simplicity of implementation. The updates are derived by (non-uniquely) expressing the gradient of the cost function as the difference of two positive terms, expressed as

$$\nabla_\theta D(\mathbf{X}|\hat{\mathbf{X}}(\theta)) = G_\theta - F_\theta \tag{2.16}$$

where $\theta \in \{\mathbf{W}, \mathbf{H}\}$ is one of the two factors. The selected factor matrix is updated via multiplication by the ratio of the two terms, i.e.,

$$\theta \leftarrow \theta \times \frac{F_\theta}{G_\theta}, \tag{2.17}$$

where the multiplication and division are both element-wise. Nonnegativity is assured as the multiplicative ratio must be positive. Coefficients will grow when the evaluated partial derivative is negative and shrink towards zero otherwise. Each factor update is guaranteed not to increase the cost, so MU-NMF with a sufficient number of iterations converges to a stationary point, which can be either a local minimum or a saddle point (Berry, Browne, Langville, Pauca, and Plemmons 2006), despite the initial claim by (Lee, Hill, and Seung 2001) of convergence to a local minimum. Multiplicative factor updates for SED-NMF are expressed as

$$\mathbf{W}_{f,r} \leftarrow \mathbf{W}_{f,r} \frac{(\mathbf{X}\mathbf{H}^T)_{f,r}}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)_{f,r}}, \tag{2.18}$$

and

$$\mathbf{H}_{r,n} \leftarrow \mathbf{H}_{r,n} \frac{(\mathbf{W}^T\mathbf{X})_{r,n}}{(\mathbf{W}^T\mathbf{W}\mathbf{H})_{r,n}} \tag{2.19}$$

where $\theta_{i,j}$ denotes the $\{i,j\}$-th entry of the factor matrix θ. Multiplicative factor updates for KL-NMF are expressed as

$$\mathbf{W}_{f,r} \leftarrow \mathbf{W}_{f,r} \frac{\sum_n \mathbf{H}_{r,n} \frac{\mathbf{X}_{f,n}}{(\mathbf{WH})_{f,n}}}{\sum_\sigma \mathbf{H}_{r,\sigma}} \tag{2.20}$$

$$\mathbf{H}_{r,n} \leftarrow \mathbf{H}_{r,n} \frac{\sum_f \mathbf{W}_{f,r} \frac{\mathbf{X}_{f,n}}{(\mathbf{WH})_{f,n}}}{\sum_f \mathbf{W}_{f,r}} \tag{2.21}$$

MU-NMF is often interpreted as a additive gradient descent method where the projection step is unnecessary and the step size is chosen adaptively and element-wise, expressed as

$$[\epsilon_\theta]_{i,j} = \left[\theta \frac{1 - \frac{F_\theta}{G_\theta}}{G_\theta - F_\theta}\right]_{i,j}. \tag{2.22}$$

While the above equation is mathematically correct, this interpretation is technically incorrect since a proper additive gradient descent method must have a uniform (i.e., not element-wise) step size in order to maintain that the step is in the direction of the gradient.

Each multiplicative update is simple to implement, but MU-NMF generally has slow convergence properties. Moreover, it is prone to suboptimal stationary solutions. For example, once an entry of θ goes to zero it must stay there so the algorithm initialized near a poor local optimum is likely to converge to that point as a solution. The inclusion of a small noise term (e.g., 10^{-9}) in the denominator is often included the avoid numerical issues.

2.8 Initialization

The choice of initialization affects both the effectiveness of the approximate factorization, i.e., the final cost, and the speed of convergence to the final value. The random nonnegative initializations presented in (Lee and Seung 1999) remain prevalent in practice, as the initialization tends to be "situation dependent" (Wang and Zhang 2012). (Berry, Browne, Langville, Pauca, and Plemmons 2006) suggested a Monte Carlo approach for problems requiring a near-global optimum, whereby the best NMF solution is chosen from amongst many trials with random initializations. Some initialization strategies investigated in the literature include Singular Value Decomposition (Langville, Meyer, and Albright 2006)

(Boutsidis and Gallopoulos 2008) and k-means clustering (Wild, Curry, and Dougherty 2004). (Bryan, Mysore, and Wang 2013) provides an interesting example of initialization in an audio source separation problem, where a stationary but perceptually unsatisfactory NMF solution was reinitialized with user annotations to the approximate spectrogram to produce a better solution.

2.9 Uniqueness

NMF is known to be an ill-posed problem in the sense that it has no unique global minimum, since $D(\mathbf{X}|\mathbf{WH}) = D(\mathbf{X}|\mathbf{WAA}^{-1}\mathbf{H})$ for any invertible matrix $\mathbf{A}$ such that $\mathbf{WA}$ and $\mathbf{A}^{-1}\mathbf{H}$ are nonnegative, e.g., scaling or permuting the factors. Local minima are non-unique by the same argument (Donoho and Stodden 2003). NMF solutions are also ambiguous under rotation transformations for which $\mathbf{A}$ need not be element-wise nonnegative. A common approach for addressing this scaling ambiguity is to normalize[3] columns of $\mathbf{W}$ following the factor updates and multiply rows of $\mathbf{H}$ by the inverse norms so that $\hat{\mathbf{X}} = \mathbf{WH}$ is unchanged by the normalization.

Whereas scaling and permutation ambiguities do not affect the qualitative properties of the NMF solution, rotation ambiguities are likely to do so. (Klingenberg, Curry, and Dougherty 2009) provided a geometric interpretation of SED-NMF and showed that the ability of NMF to produce a satisfactory basis whose elements correctly describe the components of the implicit NMF generative model is data-dependent. (Smaragdis and Brown 2003) provides an example of such dependence in an application to source separation, where NMF was shown to extract "unique events" rather than "unique notes". In particular, if two notes always occur in harmony, they are grouped together as one basis element in the NMF decomposition. This is related to the non-uniqueness of NMF solutions and can be understood through a geometric interpretation of SED-NMF as a Semidefinite Programming problem (Klingenberg, Curry, and Dougherty 2009).

2.10 Extensions

The basic NMF yields low-rank factorizable elements that are interpretable alongside the nonnegative observed data. This interpretability is purely data-driven since it results from

[3] typically by ℓ-1 or ℓ-2 norm

the nonnegativity constraint on the factors in the decomposition and not from any a priori knowledge about the structure of the data. Any such knowledge about the data can be included via an extension of the basic NMF, and is "essentially necessary" for the decomposition to correctly identify the underlying components for most applications (Wang and Zhang 2012). Extensions to the basic NMF can be broadly categorized as Constrained NMF, which alters the cost function to guide its minimization towards desirable factor solutions, and Structured NMF, which alters the generative model directly. These extensions are often realized by the inclusion of a regularized penalty term in the decomposition which is a function of the model parameters, and encourages certain behaviors in the resulting parts. For example, (Hoyer 2004) added a penalty term similar to the ℓ_1-norm to encourage sparsity in the factors. A similar penalty constraint was used in the NMF-based music transcription algorithm of (Cont 2006). Alternatively, NMF can be extended structurally by changing the form of the approximation. For example, (Smaragdis 2004) expressed the approximation as the convolution of nonnegative factors, which allowed for time-varying spectral templates in the decomposition. The structural extension from (Hennequin, Badeau, and David 2010) permits frequency-dependent time activations which are parameterized by an Autoregressive Moving Average (ARMA) model.

2.10.1 Constrained NMF

Constrained NMF includes additive penalty terms, which are functions of the factors, in the scalar cost function to be minimized. The optimization is then expressed as

$$\begin{aligned} \underset{\mathbf{W},\mathbf{H}}{\text{minimize}} \quad & D_c(\mathbf{X}||\mathbf{WH}) = D(\mathbf{X}||\mathbf{WH}) + \lambda_1 J_1(\mathbf{W}) + \lambda_2 J_2(\mathbf{H}) \\ \text{subject to} \quad & \mathbf{W}, \mathbf{H} \geq 0. \end{aligned} \tag{2.23}$$

where $J_1(\mathbf{W})$ and $J_2(\mathbf{H})$ are penalty terms constructed to encourage certain qualities in the resulting factors $\{\mathbf{W}, \mathbf{H}\}$. For example, a constrained NMF decomposition favors a sparse-basis solution when $J_1(\mathbf{W})$ is designed to be minimized by a sparse $\mathbf{W}$. λ_1 and λ_2 are regularization parameters which control the influence of the penalty terms in the optimization, and in particular control the tradeoff in priority between the fit of the model to the data and the prevalence of the desired factors qualities in the solution. Penalty terms should in general be smooth and differentiable to permit gradient-based methods in the

solution to equation 2.23. Penalty terms with an expression as the difference of positive terms as in equation 2.16 are also desirable as they permit multiplicative factor updates in the Constrained NMF.

(Hoyer 2002) and (Hoyer 2004) presented constrained NMF with penalty terms encouraging sparsity in the factors. (Cont 2006) deployed NMF with sparseness constraints to a polyphonic fundamental frequency estimation task. (Virtanen 2007) implemented NMF with sparsity and temporal smoothness constraints for an application to audio source separation. (Choi 2008) presented NMF with orthogonality constraints. (Rigaud, Falaize, David, and Daudet 2013) constructed an inharmonicity constraint for NMF applied to the transcription of piano music.

2.10.2 Structured NMF

Prior knowledge of the structure of the data can alternatively be incorporated into the NMF decomposition directly by changing the structure of the approximate factorization. Extensions to the basic NMF of this class are called Structured NMF. The basic NMF problem formulation as in equation 2.1 implicitly defines a generative model for the observed data of the form

$$\mathbf{X} = \hat{\mathbf{X}} + \mathbf{E} \tag{2.24}$$

where $\hat{\mathbf{X}} = \mathbf{WH}$ is the approximate factorization and $\mathbf{E} \in \mathbb{R}^{F \times N}$ is the residual error matrix. The optimization from equation 2.1 is then interpreted as minimizing the size of the residual with the 'size' of a matrix evaluated by the scalar cost function. Structural NMF assume a generative model of the form

$$\mathbf{X} = f(\mathbf{WH}) + \mathbf{E} \tag{2.25}$$

where the function $f(\mathbf{WH})$ reflects the structure of the data, which may involve a further parameterization of the factors. Structured NMF algorithms resemble those from basic NMF. Gradient-based structured NMF algorithms require an expression for the gradient of $f(\mathbf{WH})$ with respect to any independent variables in the model.

(Smaragdis 2004) introduced Nonnegative Matrix Deconvolution, a Structured NMF algorithm that permitted time-varying bases in the generative model. (Hennequin, Badeau, and David 2010) presented a more involved parameterization of the NMF factors using an

Autoregressive Moving Average model to allow for frequency-dependent time activations. (Bertin, Badeau, and Vincent 2010) applied NMF to a piano transcription task and imposed a harmonic structure on $\mathbf{W}$ via their composition as a linear sum of harmonic spectral envelopes tuned to fundamental frequencies of the piano keys.

2.11 Example NMF spectrogram decomposition

The 2-source NMF decomposition of a simulated mixture of synthetic sounds is shown in figure 2.1. The analyzed mixture comprises two overlapping notes synthesized at 16 kHz sampling rate. The first note has a triangle waveform and pitch value C4 (fundamental frequency 261 Hz), while the second note has a square waveform and pitch value E4 (fundamental frequency 330 Hz). The first note begins at $t = 0$ s and the second begins at $t = 0.25$ s. Both notes have a duration of 1 s, so the mixture has a total duration of 1.25 s with 0.75 s of note overlap

NMF was performed on the STFT magnitudes of the mixture, using a 64 ms Hann window for the analysis, hop of 8 ms, and the fast Fourier transform (FFT) size equal to the window length (i.e., $\mathcal{N} = M$). SED was used as the divergence measure and alternating least squares was used to update the factors in each iteration, with $R = 2$ (i.e., number of sources correctly defined a priori).

NMF provides a reasonable low-rank approximation in this case, since the columns of $\mathbf{W}$ each capture the spectral contours of one of the notes, while the rows of $\mathbf{H}$ likewise correspond to the temporal envelopes of the notes. The satisfactory analysis by NMF can be attributed to the fact that the implicit assumptions of the signal model are met, i.e., the sources are spectrally stationary and no more than one source dominates any particular time-frequency bin.

The sources are then estimated by ISTFT with the NMF factor-pairs, i.e., $\mathbf{w}_1\mathbf{h}_1^T$ and $\mathbf{w}_2\mathbf{h}_2^T$, used as time-frequency masks on the observed STFT $\boldsymbol{S}$. The estimated sources are shown alongside the true sources and mixture, in the time domain in figure 2.2, and in the spectrogram domain in figure 2.3.

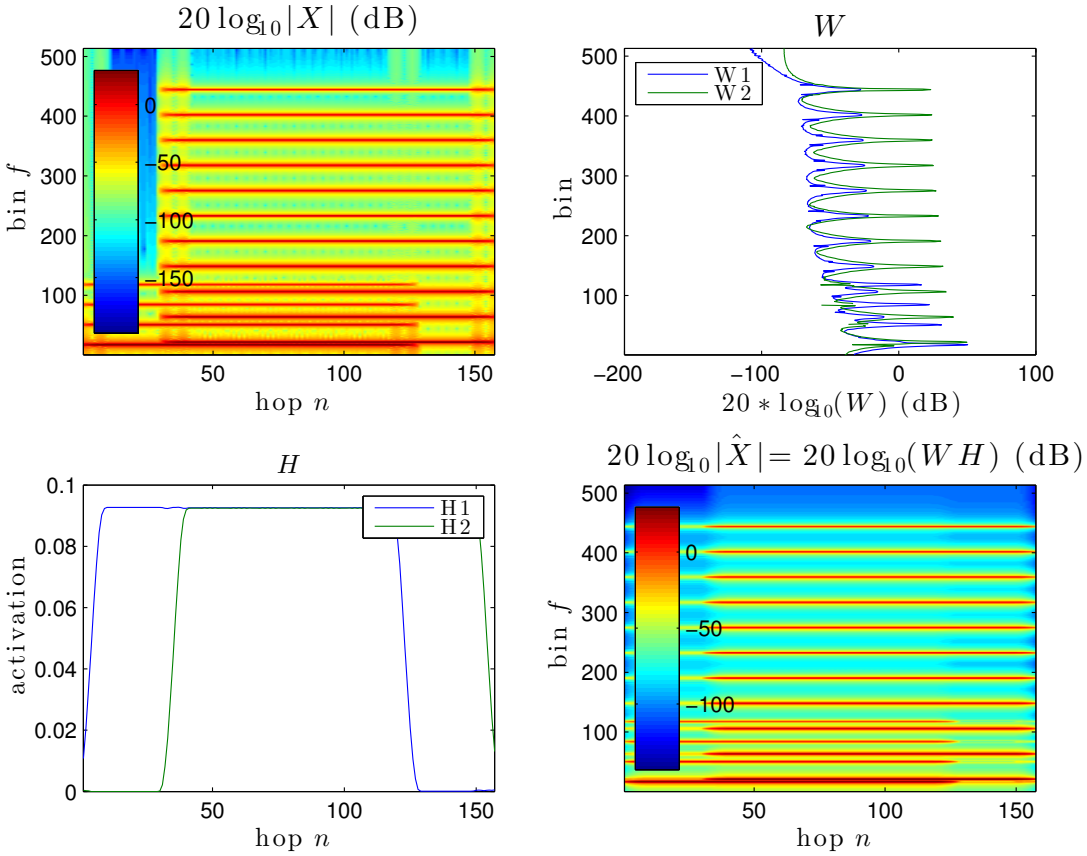

Fig. 2.1 NMF decomposition of synthetic mixture Tri C4 + Sqr E4. Specifications of the synthetic mixture are given in the text.

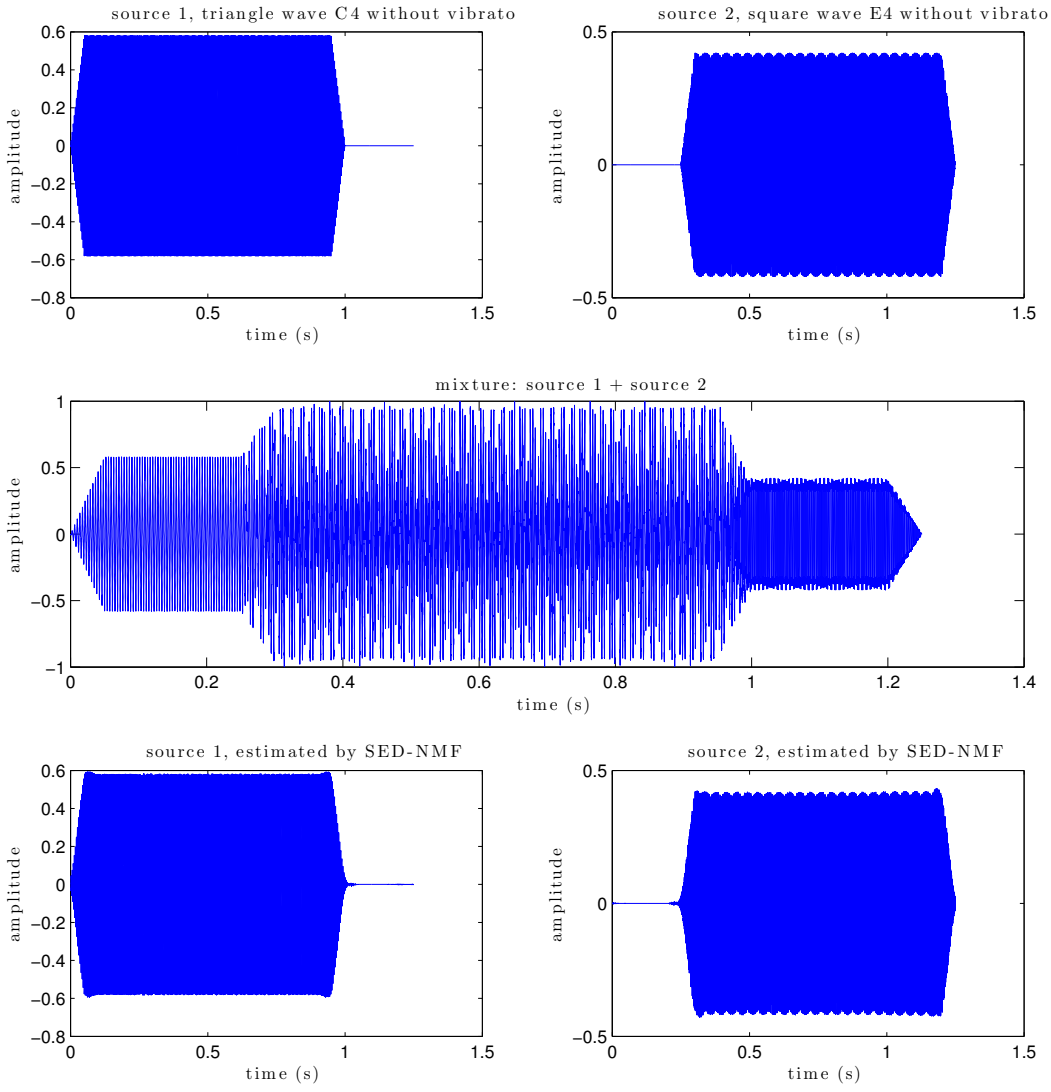

Fig. 2.2 Source estimation from NMF decomposition

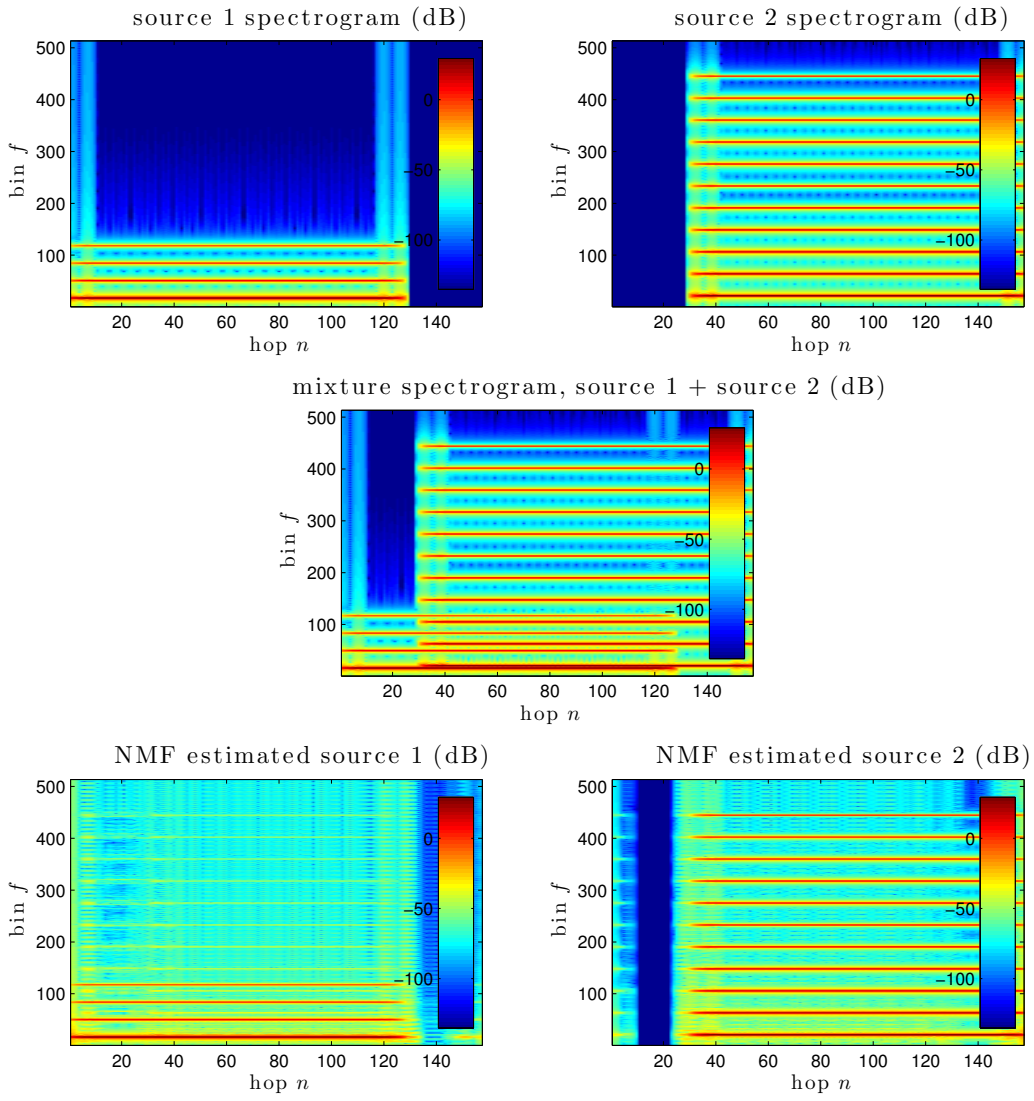

Fig. 2.3 Source spectrogram estimation from NMF decomposition

2.12 Conclusion

This chapter discussed NMF, which is considered to be a state-of-the-art source separation technique. The general problem formulation was given, along with several algorithmic approaches to finding a locally optimal solution within this framework. Some extensions to the basic NMF were then discussed. An example application to the separation of non-modulated synthetic sounds from a monaural recording was presented, and the roles of the spectral templates and temporal activations in the separation were discussed.

Chapter 3

Grouping partials by coherent frequency modulation

This chapter presents motivations for the formulation of a Coherent Frequency Modulation (CFM) source model for use in a source separation task where NMF is unsatisfactory, and discusses techniques for estimating the signal parameters relevant to such a model. We first discuss the inability of NMF, which assumes fixed spectral templates, to properly analyze vibrato sounds. We then present some motivations from the auditory perception literature for the use of frequency modulation cues in a source separation task, which precipitate the use of a non-stationary sinusoidal model to track the component partials and their frequency modulation. Methods are presented for estimating the amplitudes, frequencies, and frequency slopes of the components partials in the observed mixture under an additive sinusoidal model. We present an additive sinusoidal-source model where same-source partials are subject to a CFM, and show that same-source partials under this model have a common ratio of local frequency slope to local frequency. Finally, we propose novel source separation algorithm for the grouping of partials under an additive sinusoidal- and CFM-source model, called Partial Grouping by Coherent Frequency Modulation (PG-CFM).

3.1 Motivation

3.1.1 Analysis of vibrato sounds by NMF

The implicit signal model of NMF specifies one fixed spectral template per source, and is thus ill-suited to the analysis of vibrato sounds, which are characterized by slow frequency modulations. This is demonstrated by way of example in figures 3.1, 3.2, and 3.3, which show the NMF analysis of a simulated mixture[1] of synthetic vibrato sounds. The analyzed sound was produced by the per-source application of a vibrato effect to the sound analyzed in the previous chapter, the NMF decomposition of which was given in figure 2.2. The synthetic notes in the mixture are a triangle wave with note value C4 and a square wave with note value E4.

Unlike the non-vibrato case, which was shown in figure 2.1, NMF fails to correctly capture the spectral contours and temporal envelopes of the notes in the mixture when the notes are subject to the frequency vibrato effect. The spectral templates of the estimated sources, i.e., the columns of $\mathbf{W}$, apparently attempt to capture the full range of frequencies present in the vibrato. As a result, the two estimated sources have overlapping spectra which is problematic in the application of the time-frequency masks in the separation. The temporal activations are also incorrectly captured by the NMF, perhaps to an even more disastrous effect. The modulation seen in the observed STFT magnitudes $\mathbf{X}$, which is truly a frequency modulation and is perceived as such upon listening to the recording, is apparently captured as a dramatic amplitude modulation in the rows of $\mathbf{H}$. This amplitude modulation is can be seen in the time-domain representation of the estimated sources, shown in figure 3.2. The estimated source spectrograms are shown in 3.3.

3.1.2 Auditory scene analysis

Human listeners are excellent source separators in the sense that they are able to "follow along" with a particular instrument throughout a musical recording or live performance. The perceptual theory of Auditory Scene Analysis postulates the importance of shared frequency or amplitude modulations among partials as a perceptual cue in their grouping. (Bregman 1990) offers the following heuristic to explain the relationship: "If different parts of the spectrum change in the same way at the same time, they probably belong to the

[1] Details of the mixing process are provided in section 4.3.2.

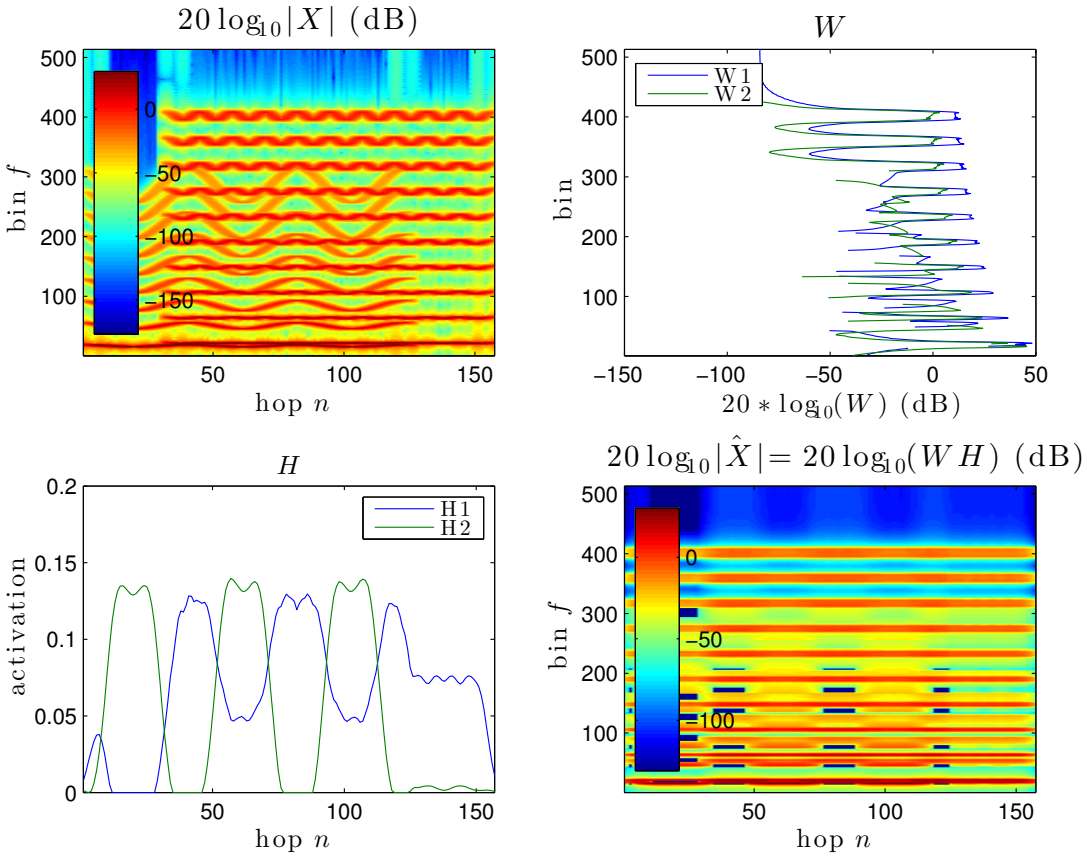

Fig. 3.1 NMF decomposition of synthetic mixture Tri C4 + Sqr E4 with vibrato

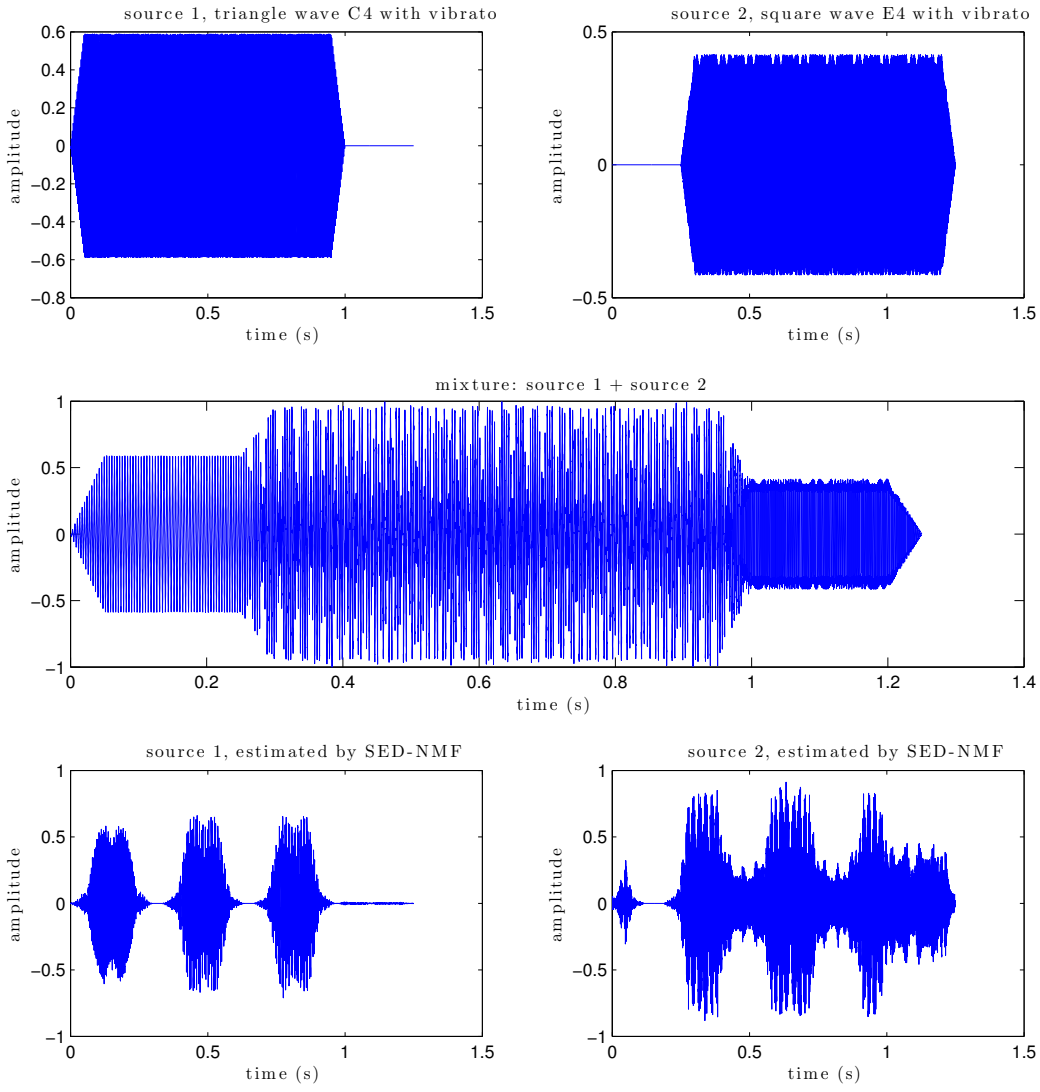

Fig. 3.2 Source estimation from NMF decomposition of vibrato sounds

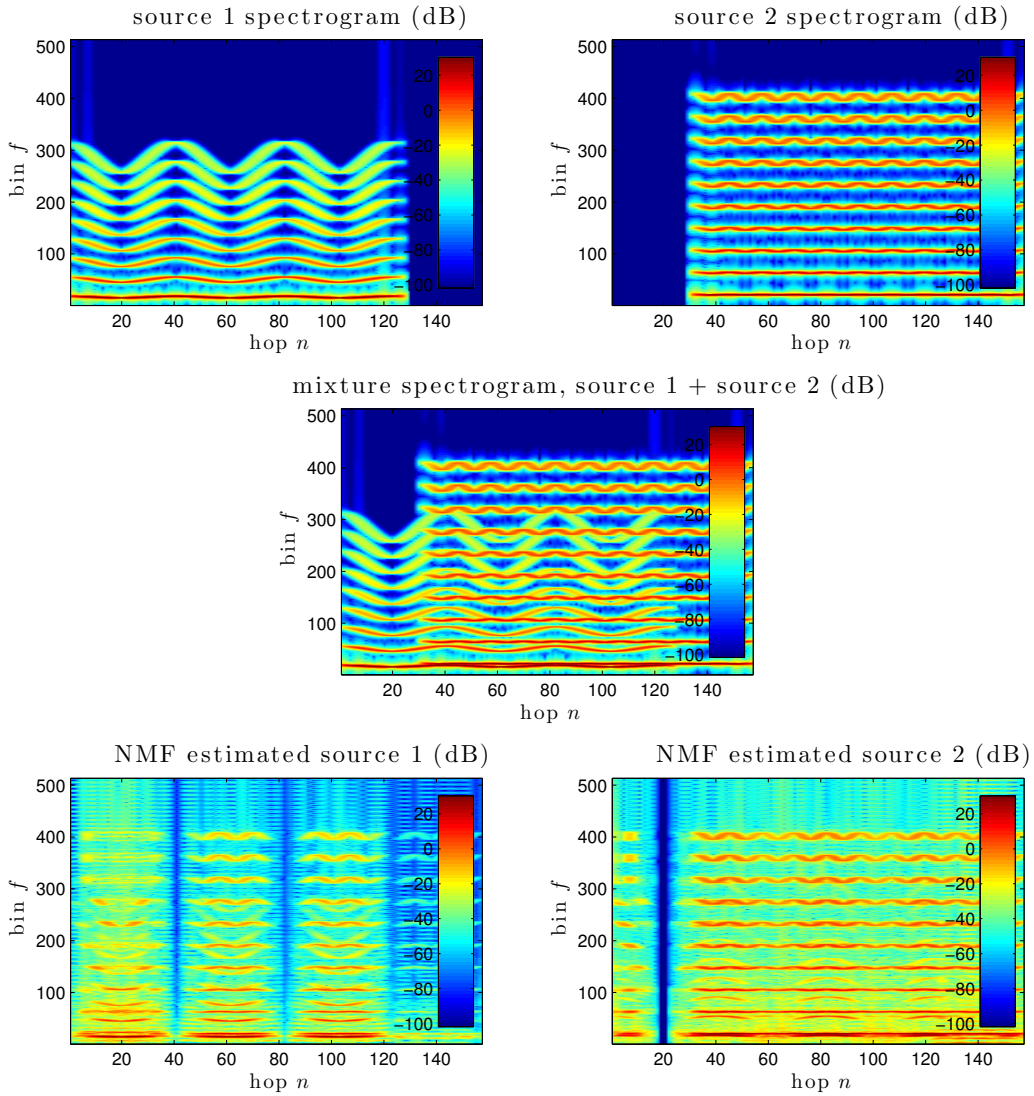

Fig. 3.3 Source spectrogram estimation from NMF decomposition of vibrato sounds

same environmental sound." This theory of perceptual grouping is based on the principle of *common fate* and stems from the assumption that components are unlikely to undergo identical frequency or amplitude modulation by chance, but are more likely "parts of the same sound, that is, that they have arisen from the same physical disturbance in the environment."

Some early empirical evidence supporting this hypothesis was presented by (Chowning 1980) in the context of synthesizing sung vowel sounds. A soprano voice was synthesized as a complex tone comprising partials at the fundamental and formant frequencies of the desired vowel sound at the desired pitch. This tone was said to "fuse" into "a unitary percept" only following the application of a coherent (i.e., shared and harmonic) frequency modulation to all partials. Application of the CFM across all partials emulates the vibrato effect, which is characterized in singing voice by a frequency modulation of the component partials at subaudible rates (Maher and Beauchamp 1990). (McAdams 1989) and (Marin and McAdams 1991) provide a more formal investigation into the effect of CFM in the perceptual grouping of component partials. Listening experiments were carried out where subjects listened to a mixture of three synthesized sung vowel sounds with different pitches and vowel types and were asked to rate the prominence of a specific vowel type in the mixture. Component partials for each vowel were subjected to CFM at a subaudible (i.e., vibrato-like) rate, with the modulation rate and depth varying according on the experiment condition. Listeners scored the target vowel as being more prominent when it was generated with the CFM.

Several techniques in the source separation literature use notions of coherent modulation in the observed partials to inform the separation. (Wang 1995) isolated individual partials with shared CFM by linear filtering following a pre-processing step to demodulate the mixture by an appropriate 'frequency warp' factor, which is derived from a maximum likelihood estimation of the instantaneous frequency. (Li, Woodruff, and Wang 2009) uses correlations in amplitude modulations to separate overlapping harmonics within a CASA framework. (Barker and Virtanen 2013) performs a Nonnegative Tensor Factorization (multi-way extension of NMF) on the so-called *modulation spectrogram*, a nonnegative tensor that encodes the inter-channel correlations in amplitude envelope modulation between channels in an auditory model.

3.2 Partial tracking

The parameter estimation of sinusoidal signal components has a rich history in computer music research, which is closely tied to developments in additive synthesis. The typical analysis framework involves the estimation of the model parameters from a short-time spectral representation, e.g., the STFT. (Portnoff 1976) provided an efficient digital implementation of the Phase Vocoder (PV), which is closely related to the STFT and provides estimates of the frequencies of sinusoidal components near the center frequencies of the FFT bins. The PV permits an exact resynthesis of the analyzed sound via additive synthesis. Alternatively the model parameters can be manipulated to achieve a pitch-shifting or time-stretching effect in the resynthesis (Moorer 1977). While the PV is able to analyze and synthesize vibrato sounds with no reconstruction error, it is ill suited to their representation since the vibrato is encoded as a coherent phase modulation across many frequency bins. Moreover, component partials of vibrato sounds are likely to cross multiple FFT bins as the fundamental frequency varies. The PV analysis of a vibrato sound essentially does not capture the underlying signal characteristics, and the application of PV-based audio effects such as time stretching results in unwanted and audible artifacts for this class of signals.

(Smith and Serra 1987) proposed PARSHL, a system for the analysis and synthesis of musical sounds under an additive sinusoidal model. Like PV, the analysis consisted of short-time signal parameter estimation at intervals equal to one hop size, i.e., "per-frame", but unlike PV, PARSHL modeled the sinusoidal signal components explicitly and without fixed pre-allocated frequencies. For each component partial in the model, the slow variations in local frequency and amplitude were tracked over the duration of the sound. (McAulay and Quatieri 1986) concurrently developed a similar system for the analysis and synthesis of speech signals. (Serra and Smith 1990) extended PARSHL to include a stochastic component (equivalent to filtered white noise) in the signal model that permits the effective analysis of non-deterministic musical sounds such as percussive transients, note onsets, and contributions of breath to the sound of singing and woodwind instruments.

3.2.1 Analysis

Here we discuss the estimation of partials in an additive sinusoidal model framework similar to (Smith and Serra 1987) and (McAulay and Quatieri 1986). It is equivalent to the deter-

ministic portion of the analysis/synthesis tool described in (Serra and Smith 1990), which was called Spectral Modeling Synthesis (SMS). In this thesis we refer signal analysis under a (purely deterministic) additive sinusoidal model as Spectral Modeling Analysis (SMA), since it fundamentally provides an analysis of the signal via an extraction of component features, and to distinguish it from the analysis portion of the work described in (Serra and Smith 1990), which technically included a stochastic component.

The input signal is modeled as the sum of sinusoidal components plus a residual, expressed as

$$x(m) = \sum_{k=1}^{K} A_k(m) \cos(\phi_k(m)) + e(m) \tag{3.1}$$

where $A_k(m)$ and $\phi_k(m)$ are the instantaneous amplitude and phase of partial k. The instantaneous phase $\phi_k(m)$ is initialized by the specification of an initial phase $\phi_k(m_0)$ and subsequently computed as the integral of the instantaneous frequency, expressed as

$$\phi_k(m) = \frac{2\pi}{f_s} f_k(m) + \phi_k(m-1) \tag{3.2}$$

where $f_k(m)$ is the instantaneous frequency of partial k and f_s is the sampling rate.

The partial frequencies are estimated per-frame via peak detection on the STFT magnitudes. A spectral interpolation scheme is required for the likely scenario that the partial frequencies do not align to the center frequencies of the FFT bins. This can be accomplished by zero-padding the windowed signal and using a larger FFT size. Although this does not increase the spectral resolution, it yields a finer spectral sampling of the signal, as well as an increase in the number of data required to store the short-time spectral representation. (Smith and Serra 1987) suggested parabolic interpolation as a more economical interpolation method, whereby the partial frequency is estimated as the peak of a parabola approximating the spectral shape of the main lobe given three data points near the peak FFT magnitude. The per-frame detected spectral peaks are subsequently sorted into of frequency guides (i.e., partial components), which is equivalent to a line detection problem (Serra and Smith 1990). This stage of the analysis is parameterized by minimum amplitude threshold, minimum duration, and maximum frame-to-frame frequency deviation of the frequency guides.

3.2.2 Synthesis

The analyzed sound can be resynthesized from the instantaneous amplitudes and phases of the tracked partials via additive synthesis. The synthesis resembles the generative model expressed by equation 3.1 excluding the error term, and is implemented using an overlap-add scheme, the details of which are provided in appendix A.2 on page 100.

3.2.3 SMA Implementation

For the experiments and applications presented in this thesis, the partial tracking analysis under an additive sinusoidal model is provided by `sms-tools`[2], a Spectral Modeling Synthesis toolbox implemented in Python and distributed with the GNU Affero General Public License. The analysis is parameterized by the standard STFT parameters: analysis window $\boldsymbol{w}$, FFT size $\mathcal{N}$, and hop size $\mathcal{H}$. Additionally, the partial detection is parameterized by the maximum number of partials in the model K and the minimum amplitude threshold for detection Thr. Partial continuity is determined according to the maximum allowable frequency deviation between successive frames of a tracked partial. This is parameterized in `sms-tools` by FDO (stands for "frequency deviation offset"), which correspond to the maximum allowable frequency deviation in Hz for a partial tracked near the 0-th bin, and by FDS (stands for "frequency deviation slope"), a Hz-per-bin ratio which sets how the maximum allowable deviation scales for higher frequencies[3]. The minimum duration of a tracked partial is parameterized by Dur.

3.3 Distributed derivative method

The local frequency slope for a given partial can be estimated by a first-order Taylor series approximation given the SMA frequency estimates at successive hops, expressed as

$$\xi_p(m) \approx \frac{f_p(m+\mathcal{H}) - f_p(m)}{\mathcal{H}}. \tag{3.3}$$

[2] `https://github.com/MTG/sms-tools`

[3] The use of both a 0-Hz offset and a per-bin slope in the parameterization of the maximum allowable frequency deviation seems appropriate since frequency vibrato modulates the higher partials proportional to their frequency.

In practice, however, these estimates are prone to spurious peaks, particularly during the onset and offset of the partial. The Distributed Derivative Method (DDM) (Betser 2009) provides an alternative frequency slope estimator, which is more robust than the first-order Taylor series approximation when partials in the analyzed sound do not overlap. DDM was developed for parameter estimation of a continuous-time monochrome analytic signal modeled as a complex exponential with time-varying polynomial amplitude and frequency laws. We implement DDM for parameter estimation of component partials of a polychrome real signal in discrete time by appropriate sampling (e.g., $s_{DT}(m) \triangleq s_{CT}(\frac{m}{f_s}) \; \forall \, m \in \mathbb{Z}$), replacing integrals and derivatives with their discrete counterparts with appropriate chain-rule scale factors.

3.3.1 Generalized sinusoidal model

We estimate the frequency slope of the k-th tracked partial via a parameter estimation on its analytic signal[4]. The monochrome analytic (complex-valued) signal is modeled in continuous time as a complex exponential with time-varying Q-th order polynomial amplitude and frequency laws, as

$$s(t) = \exp\left(\sum_{q=0}^{Q} \alpha_q t^q\right) = a(t)e^{j\phi(t)}, \tag{3.4}$$

where α_i are complex polynomial coefficients and Q is the model order. The amplitude law is given by $a(t)$ while the phase law is given by the imaginary part of the exponent argument $\phi(t) = \sum_{q=0}^{Q} \Im\{\alpha_q\} t^q$. Thus the frequency law is given by $\frac{\phi'(t)}{2\pi}$. A second-order model ($Q = 2$) is found to be sufficiently expressive in practice as music signals with vibrato exhibit a slowly varying frequency modulation which is locally well-approximated as linear.

3.3.2 Frequency slope estimation

Model parameters (amplitude and phase laws) are computed as the solution to a system of equations with each equation derived from an inner product of the signal $s(t)$ with the first derivative of a differentiable finite-support atom from the family $\{\psi_i\}$. In general, the

[4]The analytic signal can be derived via Hilbert transform or equivalently by removing negative frequencies in the spectral domain.

inner product of the signal $s(t)$ with a continuous-time function $y(t)$ is defined as

$$\langle s, y \rangle = \int_{-\infty}^{+\infty} s(t) y^*(t) dt. \tag{3.5}$$

Thus the inner product of the signal and the first derivative of the i-th atom $\psi_i'(t)$ is expressed as

$$\langle s, \psi_i' \rangle = \int_{-\infty}^{+\infty} s(t) (\psi_i')^*(t) dt. \tag{3.6}$$

We evaluate this expression using integration by parts which yields zero at the limits $t \to \pm\infty$, i.e.,

$$\Big[s(t)\psi_i(t) \Big]_{-\infty}^{+\infty} = 0 = \langle s', \psi_i \rangle + \langle s, \psi_i' \rangle. \tag{3.7}$$

This can be rewritten as this as

$$-\langle s, \psi_i' \rangle = \langle s', \psi_i \rangle \tag{3.8}$$

from which a system of equations can be derived given a family of atoms $\{\psi_i\}$.

The left- and right-hand terms in equation 3.8 can each be expressed in terms of the signal model given by equation 3.4, with the time derivative of the generalized sinusoidal model given by

$$s'(t) = s(t) \sum_{q=1}^{Q} q \alpha_q t^{q-1}. \tag{3.9}$$

Signal parameters α_q (for $q > 0$) are not time-dependent and thus factor out of the integral expressed by equation 3.6. We can thus rewrite (3.8) as:

$$-\langle s, \psi_i' \rangle = \sum_{q=1}^{Q} \alpha_q \langle s p_q', \psi_i \rangle \tag{3.10}$$

where $p_q(t) = t^q$ and thus $p_q'(t) = q t^{q-1}$. This represents the q-th equation in a system which can be written as the following matrix product

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b} \tag{3.11}$$

where $\mathbf{A} \in \mathbb{R}^{L \times Q}$ has elements $[\mathbf{A}]_{l,q} = \langle sp'_q, \psi_l \rangle$ from the signal derivative inner products, $\boldsymbol{\alpha} \in \mathbb{R}^Q$ are the model parameters $[\boldsymbol{\alpha}]_q = \alpha_q$ for $q > 0$, and $\mathbf{b} \in \mathbb{R}^L$ has elements $[\mathbf{b}]_l = -\langle s, \psi'_i \rangle$.

3.3.3 DDM Implementation

In practice we are interested in determining the per-frame frequency slopes of the component partials, whose frequency, amplitude, and phase parameters have previously been estimated via SMA. Within the n-th time frame, we estimate the frequency slope of a particular partial using DDM on the (possibly zero-padded) signal buffer $\bar{\mathbf{x}}^{(n)}$ with the selection of the L DDM atoms informed by the SMA estimates. In discrete time, the generalized sinusoidal model for the n-th signal buffer is expressed as:

$$\bar{\mathbf{x}}_m^{(n)} = \cos\left(\mathbf{a}_0 + \sum_{q=1}^{Q} \mathbf{a}_q m^q\right) \tag{3.12}$$

The model is linear in the coefficients and thus permits a more compact expression as the matrix multiplication

$$\bar{\mathbf{x}}^{(n)} = \boldsymbol{P}\mathbf{a} \tag{3.13}$$

where the polynomial matrix $\boldsymbol{P} \in \mathbb{R}^{M \times Q+1}$ is defined as

$$\boldsymbol{P} = \left(\boldsymbol{p}^{(0)}, \quad \boldsymbol{p}^{(1)}, \quad \cdots, \quad \boldsymbol{p}^{(Q)}\right) \tag{3.14}$$

with the q-th column defined by the polynomial vector $\boldsymbol{p}^{(q)} \in \mathbb{R}^M$ as

$$\boldsymbol{p}_m^{(q)} = m^q. \tag{3.15}$$

DDM atoms are chosen from the family of $\mathcal{N}$ windowed complex exponentials centered at the bin frequencies of the Discrete Fourier Transform (DFT), which we refer to as the family of DFT atoms. The f-th-bin analysis atom in this family is expressed as

$$\boldsymbol{\psi}_m^{(f)} = \bar{\boldsymbol{w}}_m e^{\frac{j2\pi f(m-1)}{\mathcal{N}}} = \bar{\boldsymbol{w}}_m \boldsymbol{\gamma}_m^{(f)} \tag{3.16}$$

where $\bar{\boldsymbol{w}}_m$ is the possibly zero-padded STFT analysis window, expressed by equation A.4 on page 99, and $\boldsymbol{\gamma}^{(f)}$ is the f-th-bin complex exponential, expressed as

$$\boldsymbol{\gamma}_m^{(f)} = e^{\frac{j2\pi f(m-1)}{\mathcal{N}}}. \tag{3.17}$$

This permits an interpretation of the inner product $\langle \bar{\mathbf{x}}^{(n)}, \boldsymbol{\psi}_m^{(f)} \rangle$ of the STFT of $\mathbf{x}$ evaluated at the n-th frame and f-th bin, i.e.,

$$\langle \bar{\mathbf{x}}^{(n)}, \boldsymbol{\psi}^{(f)} \rangle = [\mathrm{STFT}\{\mathbf{x}\}]_{f,n} \tag{3.18}$$

We choose L such atoms in the frequency neighborhood of the partial of interest (i.e., nearby the SMA frequency estimate for that partial). We estimate the signal parameters $\boldsymbol{\alpha} = [\mathbf{a}_1 \cdots \mathbf{a}_Q]^T$ as the solution to a system of equations where each equation in the system follows from equation 3.8 given one of the L chosen atoms. This system can be expressed compactly in matrix form as

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{b} \tag{3.19}$$

where $\mathbf{A} \in \mathbb{C}^{L\times(Q)}$ is the DDM matrix whose element $\mathbf{A}_{m,f}$ corresponds to the inner product of the first derivative of the n-th signal buffer $(\bar{\mathbf{x}}^{(n)})'$ and the f-th DDM atom $\boldsymbol{\psi}^{(f)}$, evaluated at time m, i.e.,

$$\mathbf{A}_{m,f} = \langle (\bar{\mathbf{x}}^{(n)})', \boldsymbol{\psi}^{(f)} \rangle_m. \tag{3.20}$$

Likewise, $\mathbf{b} \in \mathbb{C}^L$ is the DDM vector whose element $\mathbf{b}_m$ corresponds to the inner product of the n-th signal buffer and the first derivative of the f-th DDM atom $(\boldsymbol{\psi}^{(f)})'$, i.e.,

$$\mathbf{b}_m = -\langle \bar{\mathbf{x}}^{(n)}, (\boldsymbol{\psi}^{(f)})' \rangle_m. \tag{3.21}$$

The first derivatives for $\bar{\mathbf{x}}^{(n)}$ and $\boldsymbol{\psi}^{(f)}$ are obtained trivially from the respective definitions for the signal model and window, given by equations (3.12) and (A.4). When $L > Q$, the least-squares solution to equation 3.19 is given by the Moore-Penrose pseudoinverse as

$$\hat{\boldsymbol{\alpha}} = \mathbf{A}^{\dagger}\mathbf{b} = (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{b} \tag{3.22}$$

We choose order $Q = 2$ to model linear frequency modulations. The frequency law corresponds to the $\Im\{\boldsymbol{\alpha}\}$, thus the frequency slope of the partial is estimated as

$$\hat{\xi} = \frac{\Im\{\hat{\boldsymbol{\alpha}}_2\}}{\pi}. \tag{3.23}$$

DDM also provides a secondary estimate of the frequency of the partial as

$$\hat{f} = \frac{\Im\{\hat{\boldsymbol{\alpha}}_1\}}{2\pi} \tag{3.24}$$

As with spectral peak estimation in SMA, zero-padding may be used as a spectral interpolation technique. The larger FFT size affords the selection of DFT atoms from more points on the main lobe of the window, which can improve the DDM parameter estimates, particularly in the case of low-frequencies partials.

3.3.4 CFM source model

We now propose a non-stationary additive sinusoidal signal model for sources, which includes the CFM of all same-source partials and is thus called the CFM source model. The model makes no further assumption of structure (e.g., harmonicity) about the component partials. In the absence of any frequency modulation, the source is modeled as the sum of its component partials, each of which are parameterized by a time-varying amplitude and phase. The source is expressed as

$$s(m) = \sum_{p=1}^{P} A_p(m) \cos(\phi_p(m)) \tag{3.25}$$

where $A_p(m)$ and $\phi_p(m)$ are the instantaneous amplitude and phase (in radians) of partial p and time index m. The instantaneous phase is initialized by the specification of an initial phase $\phi_p(m_0)$ and subsequently computed as an integration of the instantaneous frequency, expressed as

$$\phi_p(m+1) = \phi_p(m) + \frac{2\pi}{f_s} f_p(m) \tag{3.26}$$

where $f_p(m)$ is the instantaneous frequency (in Hz) of partial p.

The application of CFM to all partials models the source during vibrato or glissando, expressed as

$$f_p(m) = f_p(m_0)(1 + \beta(m)) \tag{3.27}$$

where $f_p(m_0)$ is the instantaneous frequency of partial p in the initial state, and $\beta(m_0) \triangleq 0$.

We show that all partials belonging to the same source under this model share a common ratio of frequency slope to frequency at a given time. The local frequency slope is defined as the first derivative of the instantaneous frequency of the p-th partial, expressed as

$$\xi_p(m) \triangleq \frac{d}{dm} f_p(m). \tag{3.28}$$

Substituting in equation 3.27 and carrying the derivative through yields

$$\xi_p(m) = \frac{d}{dm} f_p(m_0) + \frac{d}{dm} f_p(m_0)\beta(m) = f_p(m_0)\beta'(m). \tag{3.29}$$

Dividing the local frequency slope by the instantaneous frequency yields the frequency-slope-to-frequency ratio, denoted by Υ_p, which does not depend on the instantaneous frequency $f_p(m)$ and is common to all same-source partials. The local frequency-slope-to-frequency ratio is expressed as

$$\Upsilon_p(m) \triangleq \frac{\xi_p(m)}{f_p(m)} = \frac{f_p(m_0)\beta'(m)}{f_p(m_0)(1 + \beta(m))} = \frac{\beta'(m)}{1 + \beta(m)} \tag{3.30}$$

Υ_p is a potentially useful feature in grouping observed partials from a mixture of sources when instantaneous frequencies and frequency slopes are available (or can be reliably estimated) for all the partials present in the mixture are available. Grouping using the coherent frequency modulation cue Υ_p does not require estimation of the common modulation function β.

3.3.5 Example feature extraction

We can approximate the frequency-slope-to-frequency ratio Υ by evaluating equation 3.30 using the per-partial, per-hop parameter estimates $\breve{f}_p$ and $\hat{\xi}_p$, which are provided by SMA and DDM respectively. Example features extracted from a simulated mixture of vibrato

sounds are shown in figure 3.4. The analyzed sound is the same one that produced the NMF decomposition plots in figures 3.1 and 3.2.

Two distinct curves are present when the analyzed sound is represented in the Υ feature space, which reflects the fact that same-source partials have the same local frequency-slope-to-frequency ratio under the CFM source model. The apparent visual separability in the feature space is an informal but encouraging result which suggests both the appropriateness of the proposed feature Υ in the analysis of vibrato sounds, and the potential for a partial grouping algorithm that employs these features to correctly separate vibrato sources from an observed recording. The goal of the partial grouping algorithm is essentially to estimate a unique feature curve for each source in the observed mixture, such that the observed data are best explained by the algorithm-estimated curves when expressed in the feature domain, i.e., as the matrix $\boldsymbol{\Upsilon}$ containing local frequency-slope-to-frequency ratio estimates for each of the partials tracked by SMA, where DDM is used to estimate the local frequency slopes.

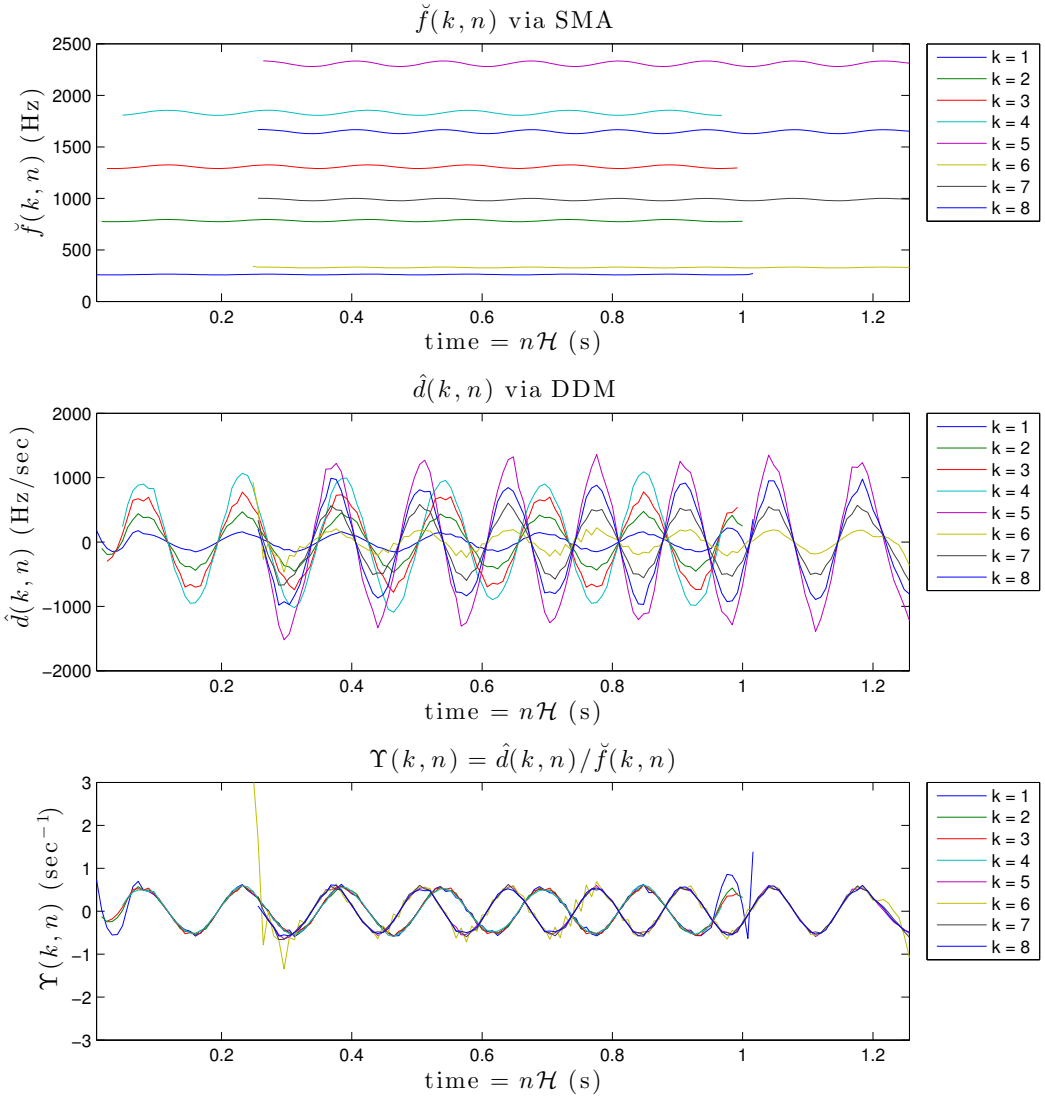

Fig. 3.4 Features extracted from synthetic mixture Tri C4 + Sqr E4. $\breve{f}(k,m)$ and $\hat{d}(k,m)$ are the respective SMA frequency estimate DDM frequency slope estimate for the k-th tracked partial at hop index n. $\Upsilon(k,n)$ is the resulting frequency-slope-to-frequency ratio feature. A coherent frequency modulation of same-source partials is apparent in the Υ feature space.

3.4 Partial Grouping by Coherent Frequency Modulation

We propose a method for grouping component partials in a non-stationary sinusoidal model via a grouping of their frequency-slope-to-frequency ratio features $\boldsymbol{\Upsilon} \in \mathbb{R}^{K \times N}$, provided by the tandem feature extraction of SMA and DDM. The number of parts R is assumed to be known. The goal is to estimate one feature vector per source such that the component partials tracked by SMA and DDM are well-explained in the feature space. Hopefully, the estimated feature vector for source r captures the CFM that uniquely characterizes the source under the CFM source model presented in section 3.3.4.

Formally, the method estimates per-source frequency-slope-to-frequency ratios $\boldsymbol{\upsilon} \in \mathbb{R}^{R \times N}$ such that $\upsilon^r(n)$ corresponds to the ratio of frequency slope to frequency for source r and time n. Concurrently, a tensor $\boldsymbol{p} \in \mathbb{R}^{R \times K \times N}$, to be interpreted as a set of "likelihoods", is estimated, such that $p^r(k, n)$ represents the likelihood that observed ratio feature[5] $\Upsilon(k, n)$ was produced by source r. We constrain the likelihoods to be nonnegative values that sum across sources to 1 at each $\{k, n\}$, i.e., each observed feature $\Upsilon(k, n)$ is well-explained by the sources $r = 1 \dots R$.

3.4.1 Cost function formulation

As with NMF, the estimation of the algorithm variables is formalized as the minimization of a scalar cost function, subject to the appropriate constraints. We formalize a cost function representing the inverse model fit of the algorithm-estimated ratios $\boldsymbol{\upsilon}$ to the observed ratio features $\boldsymbol{\Upsilon}$. The cost incurred at a time n for track k by source r comprises the squared difference of the observed ratio $\Upsilon(k, n)$ and the estimated ratio $\upsilon^r(n)$, scaled by the $p^r(k, n)$, which represents the likelihood that part r explains partial k at time n.

It is expressed as

$$J(\boldsymbol{p}, \boldsymbol{\upsilon}) = p^r(k, n)|\Upsilon(k, n) - \upsilon^r(n)|^2. \tag{3.31}$$

The cost function is minimized when $\upsilon^r(n)$ is close to the observed ratio $\Upsilon(k, n)$ if source r is likely to have explained the observation (i.e., $p^r(k, n)$ close to 1). The global cost is

[5] Here we slightly modify the previous notation for the local frequency-slope-to-frequency ratio features for consistency amongst the various matrices and tensors used in the formalization of the algorithm, and also to distinguish between "true" and "estimated" features. E.g., while $\Upsilon_p(m)$ represented the true local frequency-slope-to-frequency ratio of partial p at time m, $\Upsilon(k, n)$ represents the estimated frequency-slope-to-frequency ratio of the k-th tracked partial at hop n.

then the sum of 3.31 over $\{r, k, n\}$ plus the penalty terms

$$P_1(\boldsymbol{\upsilon}) = \sum_{r,n} (\upsilon^r(n) - \upsilon^r(n-1))^2 \tag{3.32}$$

and

$$P_2(\boldsymbol{p}) = \sum_{r,k,n} (p^r(k,n) - p^r(k,n-1))^2 \tag{3.33}$$

to encourage temporal smoothness in $\boldsymbol{\upsilon}$ and $\boldsymbol{p}$, respectively. The influences of the two penalty terms in the global cost are controlled by regularization parameters $\lambda_{\Delta\upsilon}$ and $\lambda_{\Delta p}$. The global cost is thus expressed as

$$\begin{aligned} C(\boldsymbol{p}, \boldsymbol{\upsilon}) &= \sum_{r,k,n} J(\boldsymbol{p}, \boldsymbol{\upsilon}) + \lambda_{\Delta\upsilon} P_1(\boldsymbol{\upsilon}) + \lambda_{\Delta p} P_2(\boldsymbol{\upsilon}) \\ &= \sum_{r,k,n} p^r(k,n) |\Upsilon(k,n) - \upsilon^r(n)|^2 + \lambda_{\Delta\upsilon} \sum_{r,n} (\upsilon^r(n) - \upsilon^r(n-1))^2 \\ &\quad + \lambda_{\Delta p} \sum_{r,k,n} (p^r(k,n) - p^r(k,n-1))^2 \end{aligned} \tag{3.34}$$

3.4.2 Optimization problem

The optimization problem is thus formalized as the minimization of the above cost function subject to the appropriate constraints on $\boldsymbol{p}$, and is expressed as

$$\begin{aligned} \underset{\boldsymbol{p},\boldsymbol{\upsilon}}{\text{minimize}} \quad & \sum_{r,k,n} p^r(k,n) |\Upsilon(k,n) - \upsilon^r(n)|^2 \\ & + \lambda_{\Delta\upsilon} \sum_{r,n} (\upsilon^r(n) - \upsilon^r(n-1))^2 + \lambda_{\Delta p} \sum_{r,n} (p^r(k,n) - p^r(k,n-1))^2 \\ \text{subject to} \quad & p^r(k,n) \geq 0 \,\forall\, r, k, n, \\ & \sum_{r=1}^{R} p^r(k,n) = 1 \,\forall\, \{k, n\}. \end{aligned} \tag{3.35}$$

Although $C(\boldsymbol{p}, \boldsymbol{\upsilon})$ is non-convex in both arguments, it is convex in $\boldsymbol{p}$ with fixed $\boldsymbol{\upsilon} = \boldsymbol{\upsilon}_0$, and is likewise convex in $\boldsymbol{\upsilon}$ with fixed $\boldsymbol{p} = \boldsymbol{p}_0$. As with NMF, we seek a locally optimal solution by iterating a two-step process, first fixing $\boldsymbol{\upsilon}$ and computing $\boldsymbol{p}^*$, then fixing $\boldsymbol{p}$ and computing $\boldsymbol{\upsilon}^*$.

3.4.3 Algorithm

We propose an algorithm, called Partial Grouping by Coherent Frequency Modulation (PG-CFM), which seeks a locally optimal solution to the optimization problem expressed in equation 3.35 by alternate minimization of the variables $\{\boldsymbol{p}, \boldsymbol{\upsilon}\}$, with stopping criteria set by the number of iterations η. With fixed $\boldsymbol{\upsilon}$, the optimization problem given by equation 3.35 is equivalent to a Quadratic Program (QP), a constrained optimization problem of a standard form, where the cost function to be minimized is quadratic in the independent variable. The derivation of the equivalent QP is given in appendix B. The cost function for the equivalent QP is convex in $\boldsymbol{p}$, which permits a globally optimal[6] solution $\boldsymbol{p}^*$, which can be found efficiently using existing solvers. Although the solution to such a convex QP is beyond the scope of this thesis, the interested reader should refer to (Boyd and Vandenberghe 2009) for a thorough description of solution methods for specific classes of QP.

When $\boldsymbol{p}$ is fixed to its previously assigned value, the constraints in equation 3.35 are satisfied automatically. Thus solving for the optimal $\boldsymbol{\upsilon}$ amounts to an unconstrained convex QP, which permits a closed-form solution, expressed as

$$\tilde{\boldsymbol{\upsilon}} \leftarrow (\mathbf{U}_k^T \mathbf{D}_p \mathbf{U}_k + \lambda_{\Delta\upsilon} \boldsymbol{\Lambda}_\upsilon^T \boldsymbol{\Lambda}_\upsilon)^{-1} (\mathbf{U}_k \mathbf{D}_p \mathbf{U}_r \tilde{\boldsymbol{\Upsilon}}) \tag{3.36}$$

where $\tilde{\boldsymbol{\upsilon}}$ and $\tilde{\boldsymbol{\Upsilon}}$ are the column vector representations of $\boldsymbol{\upsilon}$ and $\boldsymbol{\Upsilon}$, respectively. $\mathbf{U}_k \in \mathbb{R}^{RKN \times RN}$ and $\mathbf{U}_r \in \mathbb{R}^{RKN \times KN}$ are repeating matrices, the applications of which effectively repeat elements of $\boldsymbol{\upsilon}$ and $\boldsymbol{\Upsilon}$ along the k and r dimensions, respectively. $\in \mathbb{R}^{RKN \times RKN}$ is a smoothing matrix which effectively performs the first order difference operation expressed by the smoothness penalty on $\boldsymbol{\upsilon}$, as expressed by equation 3.32. Cf. appendix B for a full explanation of this vector notation.

The PG-CFM procedure is expressed by algorithm 3.1

[6]The solution is globally optimal in the parameter space spanned by $\{\boldsymbol{p}, \boldsymbol{\upsilon}_0\}$, since $\boldsymbol{\upsilon}$ is fixed to its previously assigned value.

Algorithm 3.1: PG-CFM

Input: $\boldsymbol{\Upsilon} \in \mathbb{R}^{K \times N}$, R, η, $\lambda_{\Delta v}$, $\lambda_{\Delta p}$

Output: $\boldsymbol{p} \in \mathbb{R}_{\geq 0}^{R \times K \times N}$, $\boldsymbol{v} \in \mathbb{R}^{R \times N}$

initialize $i = 1$ and $\boldsymbol{p}, \boldsymbol{v}$ by k-means clustering, with
$\boldsymbol{v} \in \mathbb{R}^{R \times N}$, $\boldsymbol{p} \in \mathbb{R}_{\geq 0}^{R \times K \times N}$, $\sum_{r=1}^{R} p^r(k,n) = 1 \, \forall \, k, n$;

compute $\tilde{\mathbf{p}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\Upsilon}}$ by unfolding $\boldsymbol{p}, \boldsymbol{v}, \boldsymbol{\Upsilon}$ as column vectors;

while $i \leq \eta$ **do**

- compute $\tilde{\boldsymbol{v}}, \boldsymbol{\Lambda}_p, \tilde{\mathbf{d}}$;
- $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}}^*$ by solution to equivalent QP, given by equation B.16, page 109;
- compute $\boldsymbol{\Lambda}_v, \mathbf{U}_r, \mathbf{U}_k, \mathbf{D}_p$;
- $\tilde{\boldsymbol{v}} \leftarrow (\mathbf{U}_k^T \mathbf{D}_p \mathbf{U}_k + \lambda_{\Delta v} \boldsymbol{\Lambda}_v^T \boldsymbol{\Lambda}_v)^{-1} (\mathbf{U}_k^T \mathbf{D}_p \mathbf{U}_r \tilde{\boldsymbol{\Upsilon}})$;
- $i \leftarrow i + 1$

compute $\boldsymbol{p}, \boldsymbol{v}$ by reshaping $\tilde{\mathbf{p}}, \tilde{\boldsymbol{v}}$ to original dimensions;

3.4.4 Initialization

The PG-CFM solution $\{\boldsymbol{p}^*, \boldsymbol{v}^*\}$ represents a local minimum of the scalar cost function. Both the final cost value and the qualities of solution variables depend on algorithm initialization. A random initialization strategy[7] was found to give inconsistent results with respect to the quantitative evaluation metrics, i.e., PG-CFM could not always produce a correct separation from a random initialization. We instead initialize $p^r(k,n)$ and $v^r(n)$ by a local (per-hop) k-means clustering on the observed frequency-slope-to-frequency ratios $\hat{\Upsilon}(k,n)$.

k-means clustering[8] describes the task of assigning some observed data to a pre-determined number of clusters (Bishop 2006). Each datum is assigned to only one cluster, and each cluster is parameterized by its mean χ. A scalar cost function to be minimized is formulated as the sum of all SED between each observed datum and its assigned cluster. The

[7] taking into account the appropriate constraints on $p^r(k,n)$

[8] k-means clustering is so-named because its solution provides the estimated mean for each cluster, where the number of clusters is conventionally denoted by k. In our case the number of clusters is equal to the number of sources R and should not be confused with k, the index variable for the partials tracked by SMA.

k-means clustering optimization problem is thus expressed as

$$\begin{aligned} \underset{\chi,\rho}{\text{minimize}} \quad & \rho^r(k) \sum_{r=1}^{R} \sum_{k=1}^{K} (\chi(r) - a(k))^2 \\ \text{subject to} \quad & \sum_{r}^{R} \rho^r(k) = 1 \, \forall \, k \end{aligned} \tag{3.37}$$

where $a(k)$ are the data, $\chi(r)$ is the r-th cluster mean and $\rho^r(k) \in \{0, 1\}$ is the binary-valued k-means assignment variable, which evaluates to 1 when the k-th datum is assigned to the r-th cluster, i.e.,

$$\rho^r(k) = \begin{cases} 1 & a(k) \text{ assigned to cluster } r \\ 0 & \text{else.} \end{cases} \tag{3.38}$$

The Expectation-Maximization (EM) algorithm is used to find a locally optimal solution to the k-means clustering problem (Dempster, Laird, and Rubin 1977). EM is an iterative alternating minimization algorithm resembling those discussed in chapter 2, where, within the inner loop, χ is optimized with ρ fixed and vice versa. k-means clustering is relatively efficient to implement since each optimization step represents a least-squares problem with a closed form solution.

We initialize the PG-CFM variables $p^r(k, n)$ and $\upsilon^r(n)$ by per-hop k-means clustering of the extracted features $\hat{\Upsilon}(k, n)$. For the n_0-th hop, $\{p^r(k, n_0), \upsilon^r(n_0)\}$ is initialized as the solution $\{\rho^r, \chi(r)\}$ to the k-means clustering of the extracted features $\Upsilon(k, n_0)$, i.e., the solution to equation 3.37 with $a(k) \triangleq \Upsilon(k, n_0)$. In the EM algorithm used to solve the k-means problem, $\{\rho^r, \chi(r)\}$ are themselves initialized to the solutions from the previous hop, i.e., $\{p^r(k, n_0 - 1), \upsilon^r(n_0 - 1)\}$.

k-means clustering is appropriate for the local analysis of the frequency-slope-to-frequency feature data from the K tracked partials, since, for a given hop n, the extracted features $\hat{\Upsilon}(k, n)$ are visually separable in the feature space. Locally, partial grouping resembles a clustering task. Globally satisfactory partial grouping by local clustering would require proper continuity in the cluster means across all hops. Regrettably, k-means clustering by EM is ill-suited to the global separation task for this reason, since it suffers from permutation ambiguity. Moreover, k-means clustering does not appropriately handle the case

where features from two partials belonging to separate sources cross over one another in the feature space, since they are well-explained by a single cluster during the crossing. Despite these shortcomings in the solution to the global separation problem, k-means clustering provides a reasonable and inexpensive initialization to the PG-CFM algorithm. Initialization of PG-CFM by k-means was observed to improve the consistency of the resulting separations with respect to the evaluation metrics (compared with a random initialization), although it did not, in general, improve the best-case performance.

3.4.5 Separation by masking and resynthesis

The r-th source is estimated by the application of a mask derived from the likelihoods $p^r(k,n)$, followed by an additive resynthesis. From a probabilistic interpretation, the values of the likelihoods associated with source r directly correspond to a soft mask, expressed as

$$\mu^r_{\text{soft}}(k,n) = p^r(k,n). \tag{3.39}$$

Alternatively, a hard mask is created by a rounding of the likelihoods, expressed as

$$\mu^r_{\text{hard}}(k,n) = \begin{cases} 1 & r = \underset{r}{\operatorname{argmax}}\, p^r(k,n) \\ 0 & \text{else.} \end{cases} \tag{3.40}$$

Per-source estimates for the instantaneous amplitudes of each partial are estimated by applying the mask to the observed partial amplitudes $\breve{A} \in \mathbb{R}^{K\times N}$, expressed as

$$\hat{A}^r(k,n) = \mu(k,n)\breve{A}(k,n) \tag{3.41}$$

with one of the two aforementioned masks chosen, i.e., $\mu \in \{\mu_{\text{soft}}, \mu_{\text{hard}}\}$ Each source can then be estimated via an overlap-add synthesis using the masked instantaneous partial amplitudes and the (unmasked) instantaneous phases.

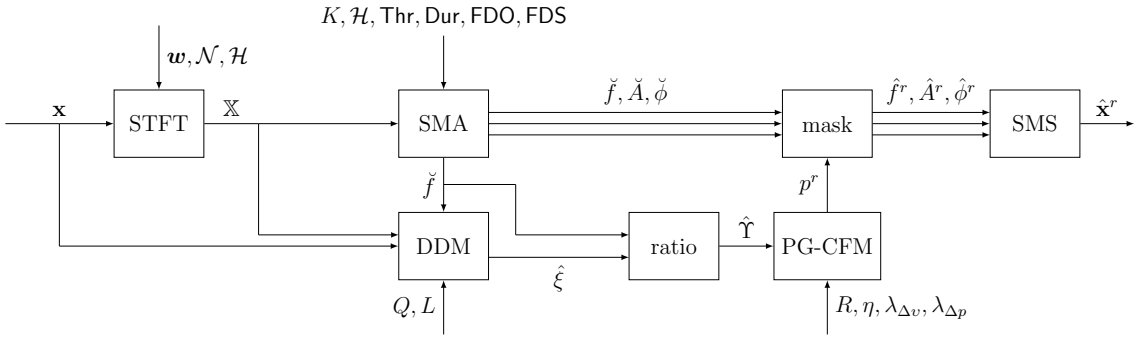

Fig. 3.5 PG-CFM block diagram

3.4.6 Application to a synthetic vibrato mixture

An application of PG-CFM to a simulated mixture of synthetic vibrato sounds is shown by figures 3.6, 3.7, and 3.8. The separation parameters were $\lambda_{\Delta v} = 215.4$, $\lambda_{\Delta v} = 1.0$, and $\eta = 6$, with hard masking used. Figure 3.6 shows the estimation of a reasonable feature vector for each of the two sources. Figure 3.7 shows the correct subsequent grouping of partials by hard masking, which results in a good separation, as shown in figure 3.8. We see in figure 3.8 a permutation ambiguity in the separation, as estimated source 2 corresponds to true source 1 and vice versa. While PG-CFM (as with NMF) does not attempt to solve the permutation problem, it can be addressed by simple projection/correlation measures if the true sources are known a priori.

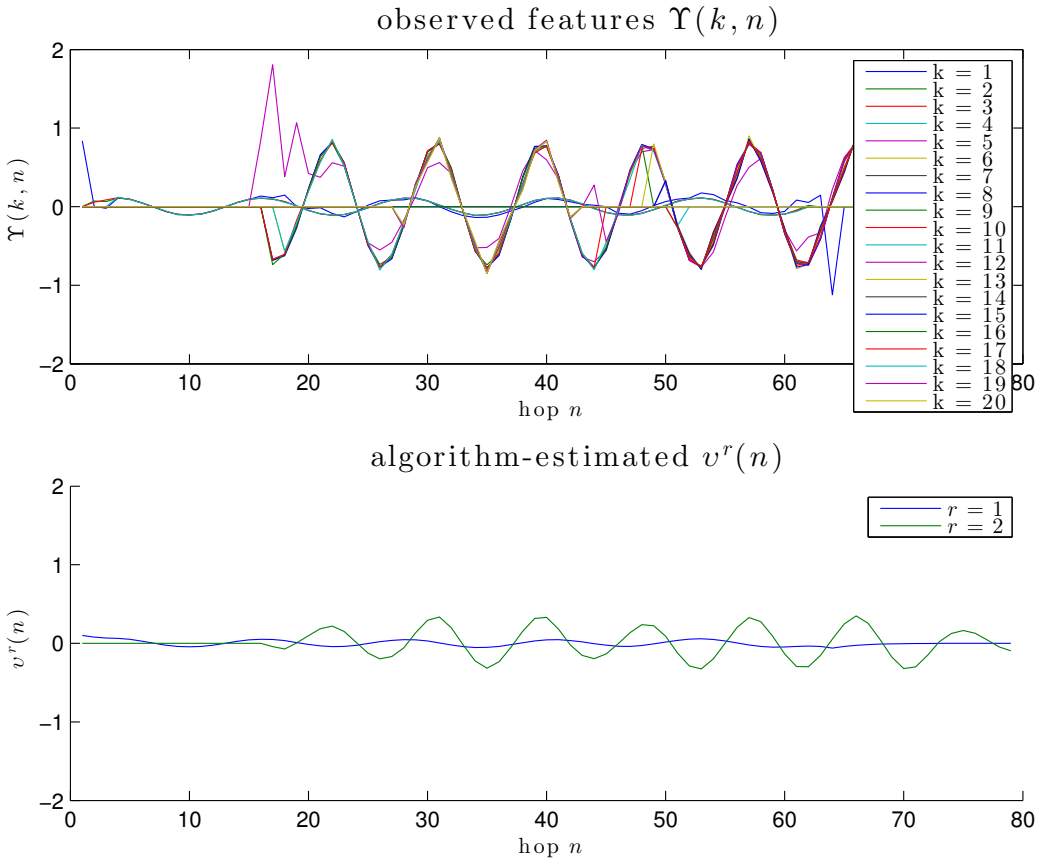

Fig. 3.6 Observed and estimated features for PG-CFM on synthetic vibrato mixture Tri C4 + Sqr E4

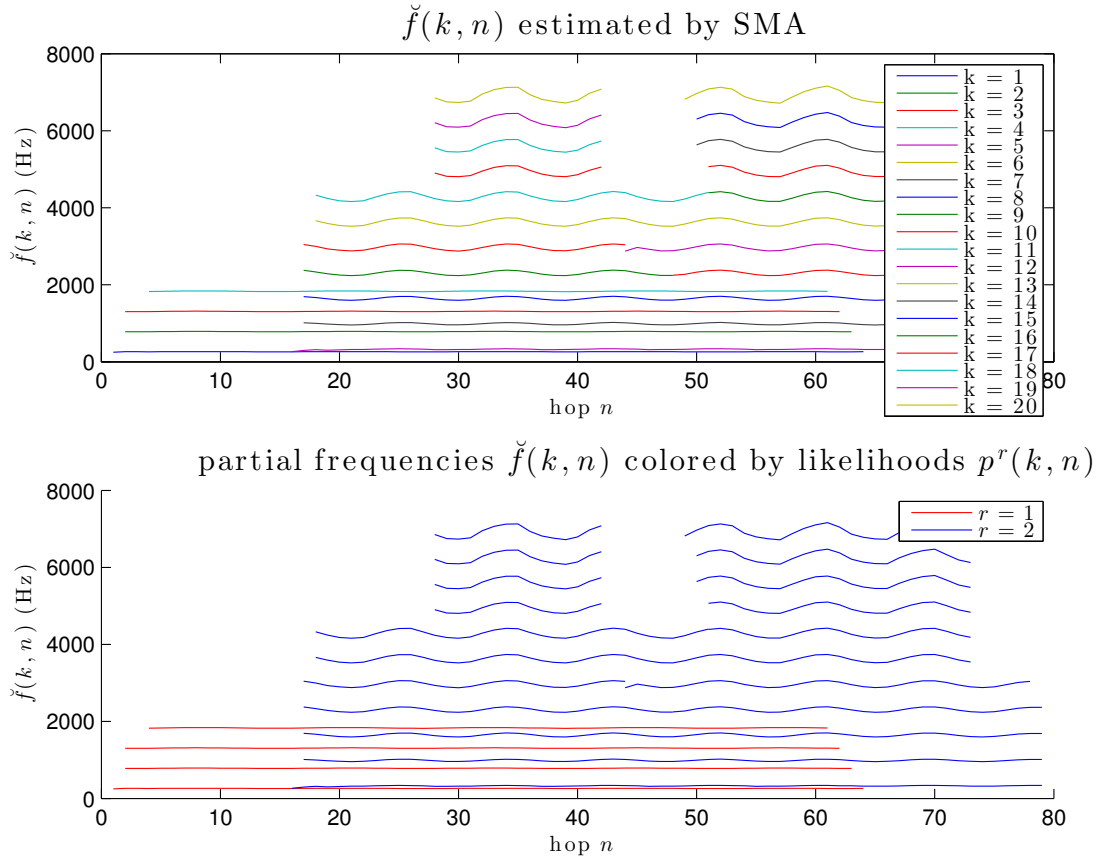

Fig. 3.7 PG-CFM synthetic vibrato mixture Tri C4 + Sqr E4: partial frequencies colored by likelihoods

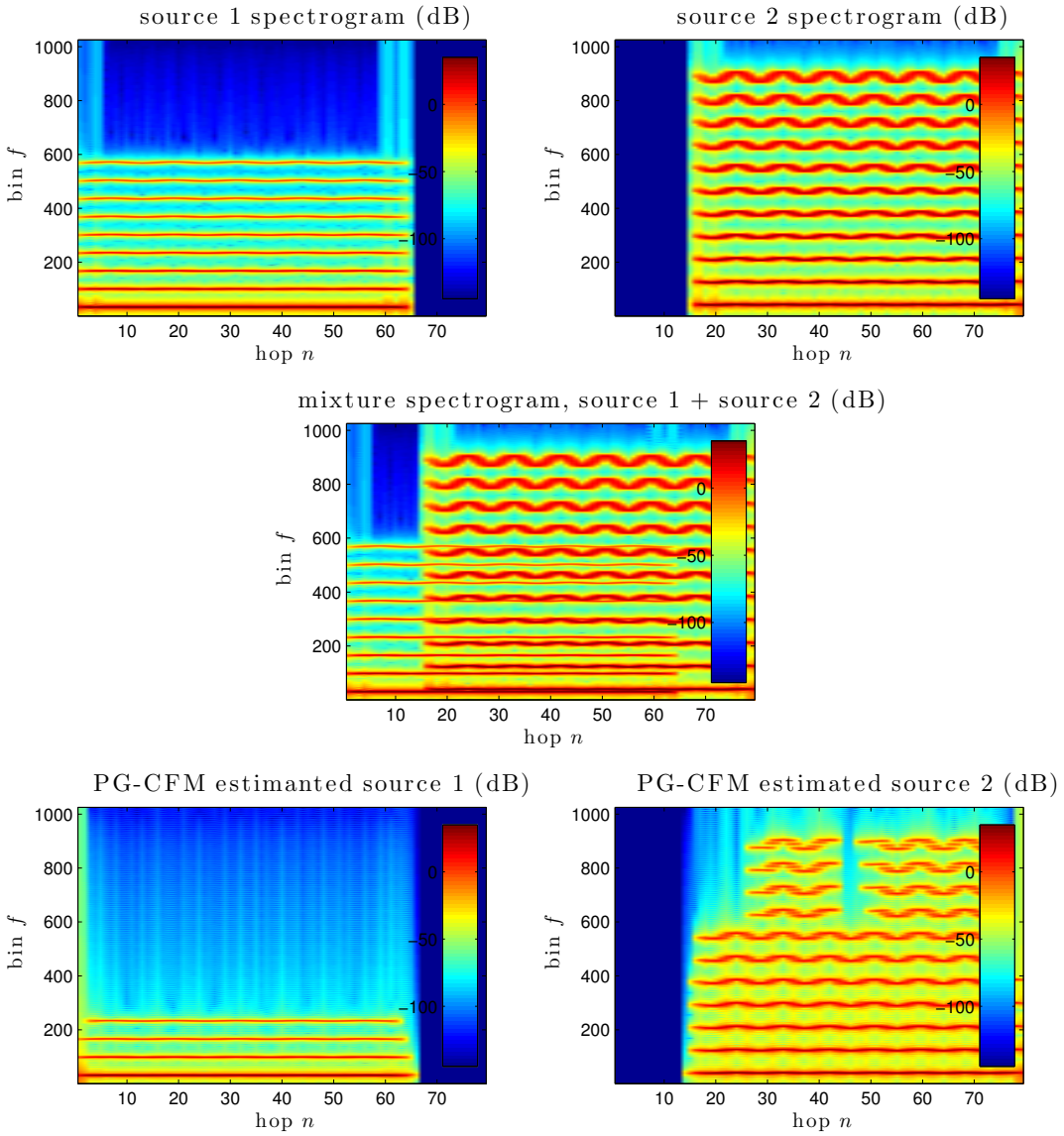

Fig. 3.8 Source spectrogram estimations from PG-CFM on vibrato mixture Tri C4 + Sqr E4

3.5 Conclusion

This chapter introduced PG-CFM, a novel algorithm for the extraction of musical sources with CFM from a monaural musical recording under an additive sinusoidal model. Motivations for this new technique were presented, namely, the shortcomings of NMF in the analysis of vibrato sounds and the proposal of CFM as a perceptual grouping cue by Auditory Scene Analysis. The CFM source model underlying PG-CFM was presented, and the per-partial local frequency-slope-to-frequency ratio feature, denoted by Υ, was introduced, along with a proposed method for estimating this signal parameter using SMA and DDM. An application of PG-CFM to the source separation of a simulated mixture of synthetic vibrato sources was then discussed, which validates our approach to analyzing this class of sounds, namely, the use of CFM cues in the grouping of the tracked partials into estimated sources.

Chapter 4

Experiment 1: PG-CFM parameter analysis

4.1 Motivation

The proposed PG-CFM source separation method, given by algorithm 3.1 necessitates the specification of several algorithm parameters. Before the performance of PG-CFM can be compared with NMF-based (or any other) source separation technique, the parameters PG-CFM parameters yielding the best performance should be determined, since the effectiveness and robustness of separation performance by PG-CFM depends on the algorithm parameter values being properly specified. This chapter discusses an experiment to determine the best-performing values for a subset of the PG-CFM parameters, evaluated using a constructed dataset of artificially mixed synthetic mixtures of single notes with frequency vibrato.

4.2 System parameters

Prior to the design of a parameter sweep experiment, the system parameters are discussed and categorized. The PG-CFM system is described by the block diagram in figure 3.5, page 50. The *analysis parameters* are those associated with STFT, SMA, and DDM blocks, and are listed in table 4.1. They must be appropriately selected to ensure meaningful features are extracted from the input audio in a robust manner. Note that parameters common to multiple analysis tools (e.g., window shape and size, FFT size) assume the same value

in each case and are thus listed only once table 4.1. The manual selection of the analysis parameter values for the experiment is further discussed in section 4.3.3.

Subsequently, the *separation parameters*, those associated with PG-CFM algorithm, must be properly set in order to ensure a partial grouping with desired qualities in the algorithm-estimated variables $p^r(k,n)$ and $v^r(n)$. In particular, a certain degree of smoothness in these variables is necessary for a good separation. These parameters are listed in table 4.2 and are further discussed in 4.3.4.

There are also a set of *synthesis parameters* associated with the resynthesis of estimated sources via SMS, given the estimated instantaneous frequencies and amplitudes for each of the sources, which result from the masking stage. These parameters are not depicted in figure 3.5 or tabulated independently, but are implicitly assigned to the same values as their corresponding analysis parameters. They are the FFT size $\mathcal{N}$, hop size $\mathcal{H}$, and sampling rate f_s.

Meaning	**Notation**	**Domain**
STFT parameters		
FFT size	$\mathcal{N}$	$\mathbb{Z}$
Window length	M	$\mathbb{Z}$
Window	$\boldsymbol{w}$	$\mathbb{R}^M$
Hop size	$\mathcal{H}$	$\mathbb{Z}$
SMA parameters		
Amplitude detection threshold	Thr	$\mathbb{R}$
Number of tracked partials	K	$\mathbb{Z}$
Frequency deviation offset	FDO	$\mathbb{R}$
Frequency deviation slope	FDS	$\mathbb{R}$
Minimum track duration	Dur	$\mathbb{R}$
DDM parameters		
Model order	Q	$\mathbb{Z}$
Number of atoms	L	$\mathbb{Z}$

Table 4.1 PG-CFM Analysis parameters

Meaning	Notation	Domain
PG-CFM separation parameters		
Number of iterations	η	$\mathbb{Z}$
Number of sources	R	$\mathbb{Z}$
v smoothness regularization	$\lambda_{\Delta v}$	$\mathbb{R}$
p smoothness regularization	$\lambda_{\Delta p}$	$\mathbb{R}$

Table 4.2 PG-CFM Separation parameters

Finally, a choice between soft and hard post-separation masking must be made. These masking schemes are expressed in equations 3.39 and 3.40, respectively. In this experiment a hard masking scheme was chosen, as it was observed empirically to produce estimated sources with fewer artifacts. The hard masking scheme implicitly assumes that each time-frequency bin contains energy from a single dominant source. It is possible that the benefits of hard masking in terms of separation results are due to the use of synthetic signals in a simulated and noiseless mixing environment, and that a soft masking scheme would generalize better to the analysis of real hi-fidelity musical mixtures.

4.3 Procedure

4.3.1 Experiment overview

An experiment was carried out to determine the set of optimal PG-CFM separation parameters, i.e., the set of regularization parameters $\{\lambda_{\Delta p}, \lambda_{\Delta v}\}$ that yields the best separation performance according to the `BSS_EVAL` metrics. Separation performance was evaluated on a dataset composed of eight simulated mixtures of synthetic vibrato notes produced according to the CFM signal model detailed in section 3.3.4. For each mixture in the dataset, a combinatorial sweep over a grid of possible separation parameter values is performed, with a PG-CFM decomposition produced for each separation parameter pair considered. The optimal separation parameter set is chosen by comparing the `BSS_EVAL` metrics for each $\{\lambda_{\Delta p}, \lambda_{\Delta v}\}$, averaged across the eight mixtures.

4.3.2 Data

A dataset of eight simulated mixtures of synthetic vibrato sources was generated for the parameter sweep experiment. The two sources to be mixed are single notes with triangle and square waveforms, respectively. Each note is one second long, with CFM vibrato applied as expressed in section 3.3.4. Each mixture is simulated by linear superposition with appropriate scaling to ensure equal power in the two sounds prior to mixing. A 50 ms linear ramp is applied to the note onsets and offsets to avoid clicking. The sources are time-shifted so that they overlap for 75% of their duration, and the resulting eight mixtures are 1.25 each seconds long.

The sources were generated by the additive synthesis model described in appendix A, with the partial amplitudes for the square and triangle waveforms specified by equations A.18 and A.22, respectively. The value of the triangle wave note was fixed to C4 (fundamental frequency 262 Hz) for all mixtures, while the square wave note took a different value in the C major scale in the octave range C4–C5 (fundamental frequencies 262–523 Hz) for each mixture in the dataset. Number of partials used to synthesize each source, i.e., P in equation A.14, was randomized to an integer in the range $[3, 11]$ and reduced as necessary to avoid aliasing.

Vibrato was applied via the application of CFM to all partials, as expressed in equation 3.27. Vibrato depth was selected randomly from the range of $[2, 19]$ % of the fundamental frequency. Vibrato rate was selected randomly from the range of $[1, 11]$ Hz. These ranges depth and rate correspond to a mild exaggeration of the guidelines for vocal vibrato synthesis provided by (Maher and Beauchamp 1990), so that the resulting sounds have reasonable but pronounced frequency vibrato. Frequency law and vibrato law initial phases were randomized in the range $[0, 2\pi]$.

Time-domain, dB spectrogram, and partial tracking representations of one such mixture are shown in figure 4.1.

4.3.3 Selection of analysis parameters

The goal of this experiment is to determine the set of ideal separation parameters, i.e., the inputs to algorithm 3.1 that are not the extracted features $\boldsymbol{v}$. The remaining system parameters, i.e., the analysis parameters, influence the system performance insofar as they produce meaningful features $\boldsymbol{v}$ as input to the separation algorithm. These parameter

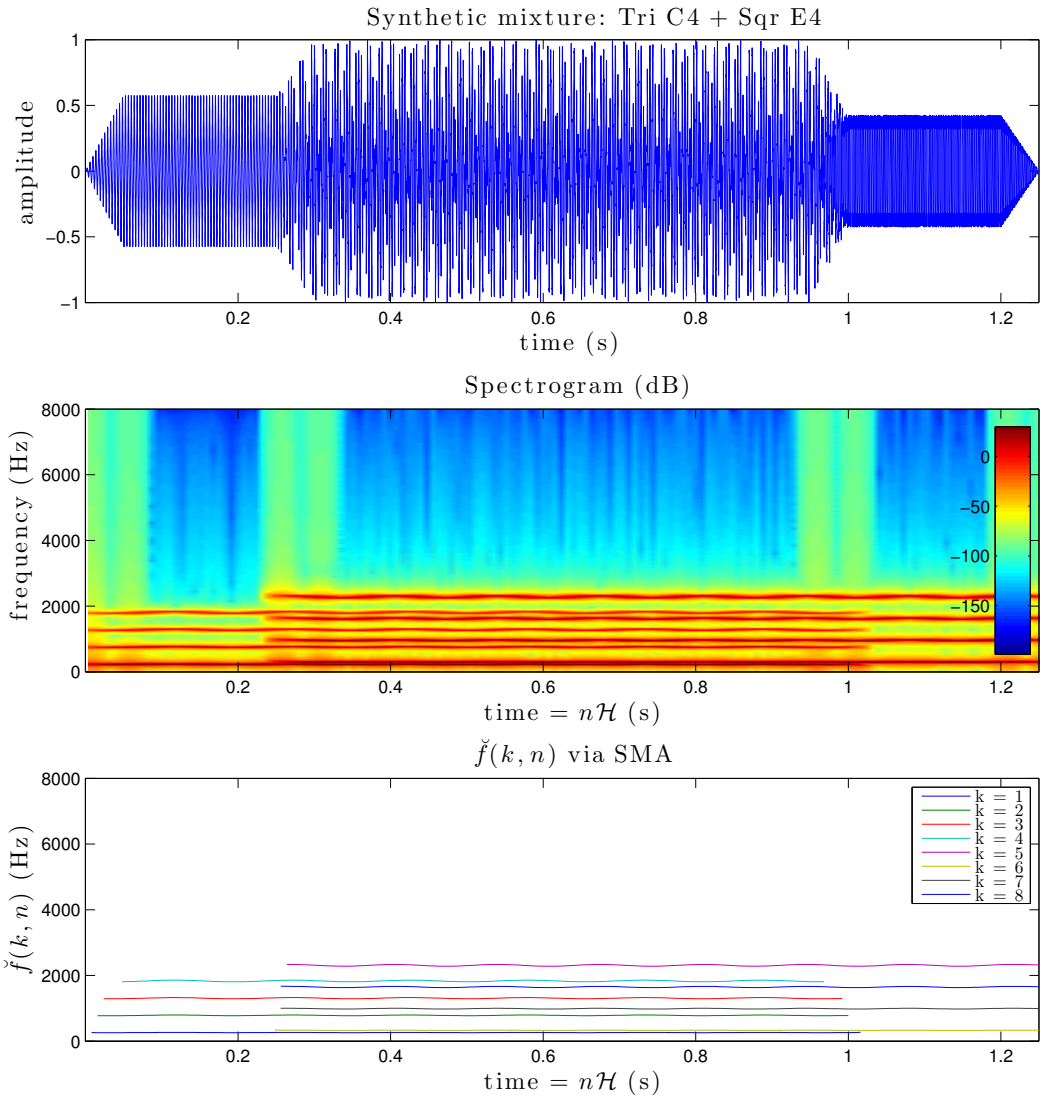

Fig. 4.1 Synthetic mixture Tri C4 + Sqr E4 in time domain, STFT magnitudes in dB, and SMA frequency estimates

values are hand-tuned prior to the experiment, with the selection of values informed by knowledge of the signal characteristics (e.g., range of fundamental frequencies, maximum vibrato rate). These considerations are discussed in this section. Fixing the analysis parameters also greatly reduces the computational resources required for the experiment since the run time of a combinatorial parameter sweep scales polynomially with the number of independent parameters.

It is well known that the choice of window size in STFT-based analysis corresponds to a tradeoff between spectral and temporal resolution, with spectral resolution proportional to window length. In this case, we want the window to span at least four periods of the lowest frequency of interest in order to properly resolve low frequency components. The window length is chosen as 1024 samples, which corresponds to 64 ms at 16 kHz sampling rate. This permits proper resolution of fundamental frequencies for notes as low as B1, while retaining an acceptable temporal resolution.

The STFT hop size must be small enough to guarantee proper subsampling of the short-time spectra, which relates to the bandwidth of the main lobe of the analysis window spectrum (Allen and Rabiner 1977). In the case of the Hann window, the hop size can be at most $\frac{1}{4}$ of the window size (Dolson 1986). The hop size is chosen as $\frac{1}{4}$ the size of the window, or 16 ms. We zero-pad the analysis window by a factor of 2 prior to the FFT computations in the STFT, SMA, and DDM blocks, i.e., $\mathcal{N} = 2M$. Zero padding was identified as being important to the satisfactory performance of DDM, since it permits the selection of more DDM atoms on the main lobe of the analysis window for the frequency slope estimation of a particular partial. This is demonstrated by figures 4.2 and 4.3, which show, for a particular hop $n = 78$[1], the selected DDM atoms overlaid on the STFT magnitudes and phases.

The DDM model order is set to $Q = 2$, which is sufficient to model the local (per-hop) frequency modulation as linear. The number of DDM atoms is set to $L = 4$.

All analysis parameters values used in experiment 1 are listed in table 4.3. Values for the separation parameters that yield the best performance ideally do not depend on the data but likely do depend on the analysis parameters. Therefore if the analysis parameters were to change (e.g., higher sampling rate, longer window, no frequency oversampling) the parameter sweep experiment would likely need to be run again to determine a new set of optimal separation parameters.

[1] Tri C4 and Sqr E4 overlap for the chosen n

Parameter	Experiment 1 value
Data parameters	
Sampling rate	16 kHz
STFT parameters	
FFT size	2048
Window length	1024 samples (64 ms)
Hop size	256 samples (16 ms)
SMA parameters	
Amplitude detection threshold	-50 dB
Maximum number of partials tracked	25
Frequency deviation offset	20 Hz
Frequency deviation slope	0.11 Hz/bin
Minimum track duration	200 ms
DDM parameters	
Model order	2
Number of atoms	4

Table 4.3 Experiment 1 PG-CFM analysis parameter values

4.3.4 Sweeping across separation parameters

We sweep over a parameter grid with logarithmic spacing, with the minimum and maximum evaluated parameter values chosen, such that the parameter grid extends over a variety of parameter pairs observed to give reasonable results in some informal initial investigations. The parameter grid values are $\lambda_{\Delta v} = 10^{\{-3,-2,-1,0,1,2,3\}}$ and $\lambda_{\Delta p} = 10^{\{1,1.67,2.33,3,3.67,4.33,5\}}$.

$\lambda_{\Delta p}$, which encourages smoothness in $\boldsymbol{p}$, was found to require a larger value (compared its counterpart $\lambda_{\Delta v}$) in order to be effective in the sense of producing visibly smooth $\boldsymbol{p}$ in the PG-CFM solution. This can likely be attributed to the domain of $\boldsymbol{p}$ being much larger than that of $\boldsymbol{v}$, and perhaps also due to the constraints on $\boldsymbol{p}$ that forbid it from assuming large values.

The number of iterations is set to $\eta = 6$, which was observed to be suffice, in general, for convergence to a locally optimal point, i.e., cost function no longer decreases with further

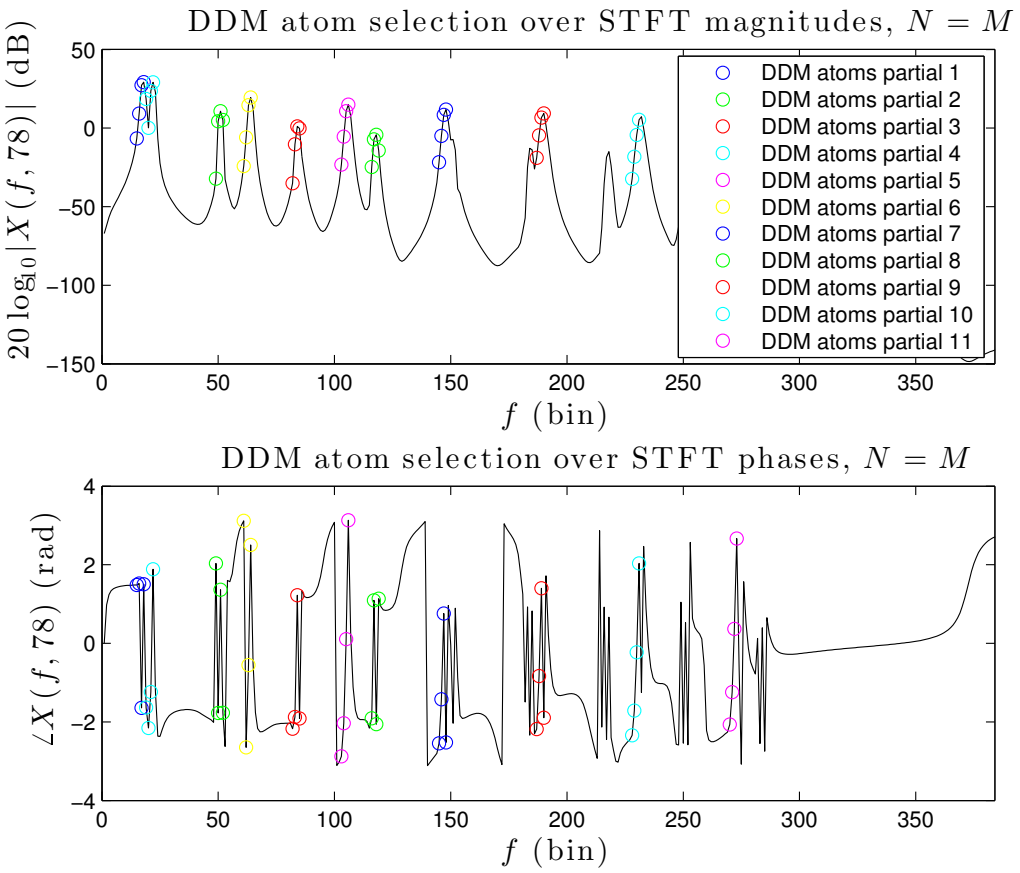

Fig. 4.2 Synthetic vibrato mixture Tri C4 + Sqr E4 selected DDM atoms; $Q = 2, L = 4, N = M$

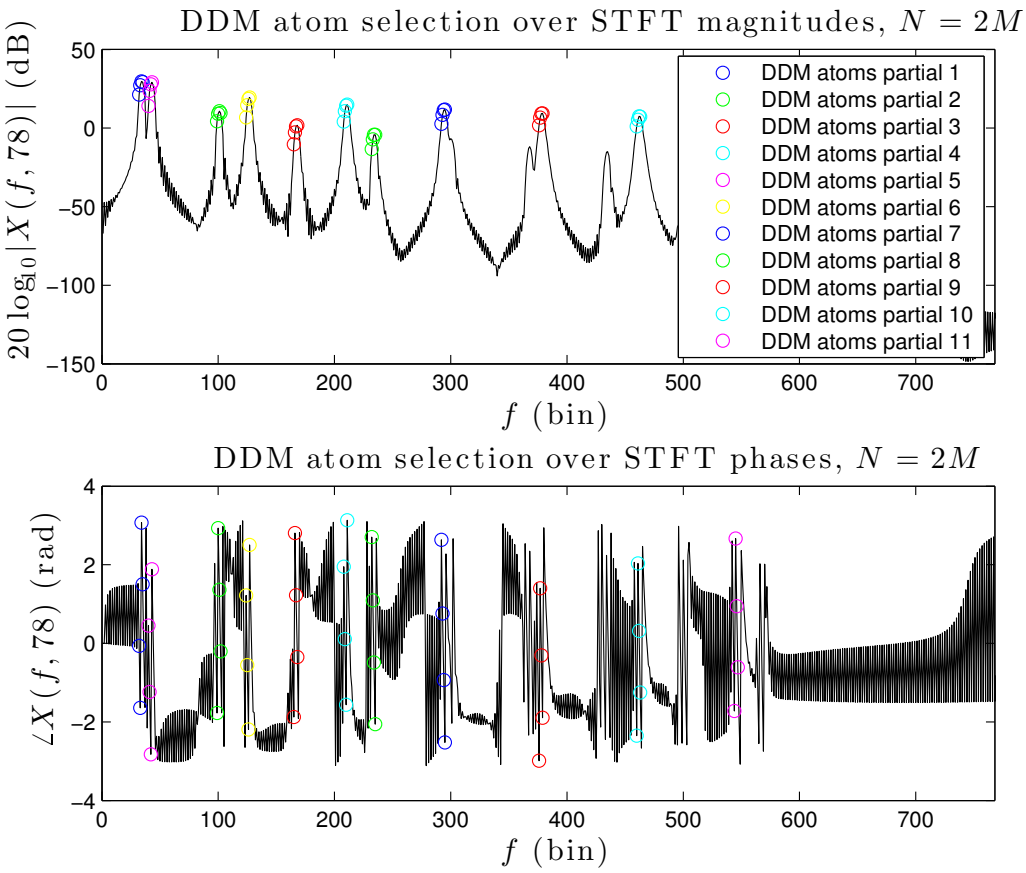

Fig. 4.3 Synthetic vibrato mixture Tri C4 + Sqr E4 selected DDM atoms; $Q = 2, L = 4, N = 2M$

iterations. The correct number of sources $R = 2$ is specified a priori. Hard masking was used in the separation.

4.3.5 Evaluation

For the quantitative evaluation of separation performance, we use metrics from (Vincent, Gribonval, and Févotte 2006), which were implemented in the `BSS_EVAL` toolbox[2], and are hereafter referred to as the `BSS_EVAL` metrics. `BSS_EVAL` quantifies several types of distortion typically of masking-based source separation algorithms, which are computed by orthogonal projection. The observed mixture $\mathbf{y} \in \mathbb{R}^S$ is modeled as the sum of R true sources $\mathbf{x} \in \mathbb{R}^{S \times R}$ according to some mixture mapping, plus some additive noise $\mathbf{n} \in \mathbb{R}^S$, as

$$\mathbf{y} = \mathcal{A}(\mathbf{x}) + \mathbf{n}. \tag{4.1}$$

The source separation algorithm produces R estimated sources $\{\hat{\mathbf{x}}_r\}$. The j-th estimated source is modeled as the sum of contributions from the target signal plus a variety of distortion terms, expressed as

$$\hat{\mathbf{x}}_j = \mathbf{x}_{\text{target}} + \mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{artif}} \tag{4.2}$$

where $\mathbf{x}_{\text{target}}$ resembles the j-th true source $\mathbf{x}_j$, possibly subject to some allowable distortions, and $\mathbf{e}_{\text{interf}}$, $\mathbf{e}_{\text{noise}}$, and $\mathbf{e}_{\text{artif}}$ are the interference, noise, and artifact error terms, respectively. The source-to-distortion (SDR) ratio is defined as

$$\text{SDR} \triangleq 10 \log_{10} \frac{\|\mathbf{x}_{\text{target}}\|^2}{\|\mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}} + \mathbf{e}_{\text{artif}}\|^2}. \tag{4.3}$$

The source-to-interference (SIR) ratio is defined as

$$\text{SIR} \triangleq 10 \log_{10} \frac{\|\mathbf{x}_{\text{target}}\|^2}{\|\mathbf{e}_{\text{interf}}\|^2}. \tag{4.4}$$

The source-to-noise (SNR) ratio is defined as

$$\text{SNR} \triangleq 10 \log_{10} \frac{\|\mathbf{x}_{\text{target}} + \mathbf{e}_{\text{interf}}\|^2}{\|\mathbf{e}_{\text{noise}}\|^2}. \tag{4.5}$$

[2] `http://bass-db.gforge.inria.fr/bss_eval/`

The source-to-artifacts (SAR) ratio is defined as

$$\text{SAR} \triangleq 10 \log_{10} \frac{\|\mathbf{x}_{\text{target}} + \mathbf{e}_{\text{interf}} + \mathbf{e}_{\text{noise}}\|^2}{\|\mathbf{e}_{\text{artif}}\|^2}. \tag{4.6}$$

The target vector $\mathbf{x}_{\text{target}}$ and each of the noise terms, $\mathbf{e}_{\text{interf}}$, $\mathbf{e}_{\text{noise}}$, and $\mathbf{e}_{\text{artif}}$, are calculated per-source by a series of orthogonal projections given the true sources $\{\mathbf{x}_j\}$ and the j-th estimated source $\hat{\mathbf{x}}_j$, as

$$\mathbf{x}_{\text{target}} \triangleq P_{\mathbf{x}_j} \hat{\mathbf{x}}_j \tag{4.7}$$

$$\mathbf{e}_{\text{interf}} \triangleq P_{\mathbf{x}} \hat{\mathbf{x}}_j - \mathbf{x}_{\text{target}} \tag{4.8}$$

$$\mathbf{e}_{\text{noise}} \triangleq P_{\mathbf{x},\mathbf{n}} \hat{\mathbf{x}}_j - P_{\mathbf{x}} \hat{\mathbf{x}}_j \tag{4.9}$$

$$\mathbf{e}_{\text{artif}} \triangleq \hat{\mathbf{x}}_j - P_{\mathbf{x},\mathbf{n}} \hat{\mathbf{x}}_j \tag{4.10}$$

where $P_{\mathbf{x}_j} \hat{\mathbf{x}}_j$ is the projection of the j-th estimated source onto the j-th true source, $P_{\mathbf{x}} \hat{\mathbf{x}}_j$ is the projection of the j-th estimated source onto the subspace spanned by all true sources, and $P_{\mathbf{x},\mathbf{n}} \hat{\mathbf{x}}_j$ is the projection of the j-th estimated source onto the union of subspaces spanned by the true sources and true noise.

For each pair $\{\lambda_{\Delta p}, \lambda_{\Delta v}\}$ in the parameter grid, PG-CFM was performed 5 times on each of the 8 mixtures in the dataset, for a total of 40 trials per parameter pair. We run the algorithm multiple times for each note pair to account for the local optimality of the PG-CFM solution.

4.4 Results

The `BSS_EVAL` metrics give one SDR, SIR, and SAR score for each source estimated by a PG-CFM solution. The SDR, SIR, and SAR scores reported are calculated by considering the evaluation score for a single trial to be the mean SDR, mean SIR, and mean SAR scores averaged across all sources, i.e., the performance is assessed based on the average quality of all estimated sources, rather than the quality of the best-estimated source.

We omit the SNR `BSS_EVAL` metric in the results since simulated mixtures presented in this thesis do not use additive noise so $\mathbf{n}$ and $\mathbf{e}_{\text{noise}}$ evaluate to 0. SIR results are notably high, reaching 40 dB in the best case. This can be attributed the class of examined synthetic signals, which are unlikely to mutually correlate.

In order to determine the optimal separation parameters, we examine the median values and variances for the 40 runs of PG-CFM which were produced for each $\{\lambda_{\Delta p}, \lambda_{\Delta v}\}$ in the parameter grid. We prioritize a parameter pair which yields both (a) high median and (b) low variance for its 40 `BSS_EVAL` scores. We thus devise a criteria of "consistent goodness", whereby a parameter pair is called consistently good if it produces median SDR and SAR in the top quartile compared with all points on the parameter grid, and likewise SDR variance and SAR variance in the bottom quartile. The `BSS_EVAL` metrics for parameter pairs with "consistent goodness", along with the parameter pairs yielding the best and worst median SDR results, are shown in table 4.4. The metrics are presented by their median values, with the lower and upper bounds of the interquartile range given in braces.

		Median [Q1, Q3]		
$\lambda_{\Delta v}$	$\lambda_{\Delta p}$	**SDR**	**SIR**	**SAR**
Parameters yielding maximum median SDR				
10^{0}	$10^{2.33}$	$19.96[3.85, 25.30]$	$41.60[8.63, 42.20]$	$20.25[14.21, 25.43]$
Parameters yielding minimum median SDR				
10^{-3}	10^{1}	$-0.97[-3.35, 4.08]$	$7.45[2.29, 12.58]$	$2.49[0.77, 9.74]$
Parameters meeting the "consistent goodness" criteria				
10^{-2}	$10^{3.66}$	$19.84[8.80, 24.23]$	$37.42[11.25, 42.18]$	$20.21[17.10, 28.42]$
10^{-2}	$10^{4.33}$	$19.84[8.80, 24.23]$	$37.42[13.23, 42.18]$	$20.21[17.10, 28.42]$

Table 4.4 PG-CFM parameter sweep selected results

4.5 Discussion

We see that the inclusion of the penalty terms in the formulated cost function is necessary in order to steer the algorithm towards a meaningful solution, since the worst-performing parameter pair corresponds to smallest values possible on the separation parameter grid. The "consistent goodness" criteria was satisfied for only two parameter pairs on the grid: $\{\lambda_{\Delta p}, \lambda_{\Delta v}\} = \{10^{-2}, 10^{3.66}\}$ and $\{\lambda_{\Delta p}, \lambda_{\Delta v}\} = \{10^{-2}, 10^{4.33}\}$. These two parameter pairs

perform nearly identically on the dataset, even across the multiple algorithm runs. This suggests that once the penalty terms are in the correct region of influence, they are not sensitive to small changes in their corresponding regularization parameters.

4.6 Conclusion

This chapter presented a parameter sweep experiment designed to determine the optimal set of PG-CFM algorithm parameters. The experiment was carried out using a dataset of simulated mixtures of synthetic vibrato signals. Two parameter pairs were identified as performing consistently well over the set of analyzed data, which represent the set of permissible PG-CFM parameters to be used in subsequent source separation tasks.

Chapter 5

Experiment 2: Evaluation of PG-CFM vs. NMF

In this chapter, we present the results of an experiment that was carried out to compare the performance of PG-CFM with basic NMF on a source separation task using a dataset of simulated mixtures of synthetic signals. We then discuss an application of PG-CFM to the analysis of simulated mixtures of vibrato singing voice sounds. The procedure for this application is less rigorous than the synthetic sound experiment since fewer sounds are analyzed, the analysis parameters must be tuned by hand according to each analyzed sound. Evaluation results are presented for the application to natural sounds and compared with similar results from an NMF-based separation.

5.1 Experiment 2a: evaluation of PG-CFM vs. NMF with synthetic data

5.1.1 Motivation

Basic NMF was shown in section 3.1.1 to be ill-suited for the analysis sounds comprising the mixtures of sources with CFM vibrato, in the sense that the estimated sources do not properly capture the frequency modulation present, which results in estimated sources of a poor quality. We here design an experiment to provide a quantitative comparison of NMF and PG-CFM in a source separation task where the analyzed sounds are mixtures of

synthetic vibrato notes. The PG-CFM algorithm parameter are informed by the results of the parameter sweep experiment, which was presented in chapter 4.

5.1.2 System parameters

Selection of the PG-CFM separation parameter values was informed by the experiment described in chapter 4. In particular, we choose the parameter pair $\{\lambda_{\Delta p}, \lambda_{\Delta v}\} = \{10^{-2}, 10^{4.33}\}$, which was observed to perform with "consistent goodness" on the dataset considered for the parameter sweep experiment. The PG-CFM analysis parameters are set to the values used in Experiment 1, which were given by table 4.3.

PG-CFM was compared with a basic NMF implementation, for which SED was used as the divergence measure and alternating least squares was used to update the factors in each iteration, with $R = 2$ (i.e., number of sources correctly defined a priori).

5.1.3 Data

The dataset comprised 100 simulated mixtures of the same class previously discussed, namely, of synthesized triangle and square wave notes with vibrato effect added. Whereas the experiment 1 examined simulated mixtures with fixed note values, here the value of each note was selected randomly from the set of notes on a piano keyboard, i.e., A0–C8 (fundamental frequencies 28–4186 Hz). The vibrato parameter values were randomly selected from the same ranges as in experiment 1, which were specified in section 4.3.2, page 58.

5.1.4 Procedure

For each simulated mixture in the dataset, PG-CFM and NMF were run 5 times. The `BSS_EVAL` metrics were computed for each of the solutions produced, for a total of 500 values of SDR, SAR, and SIR produced for each of the two algorithms.

5.1.5 Results

For each algorithm, the interquartile ranges of the 500 evaluation results are shown in table 5.1. These ranges are illustrated by the box plot shown in figure 5.1.

	Median [Q1, Q3]		
Algorithm	**SDR**	**SIR**	**SAR**
PG-CFM	20.21[19.29, 23.67]	33.85[31.00, 37.77]	25.26[22.98, 26.98]
NMF	15.91[9.29, 19.77]	22.27[14.91, 27.68]	18.56[14.09, 21.42]

Table 5.1 PG-CFM vs. NMF for source separation of randomly generated mixtures of synthetic vibrato sounds

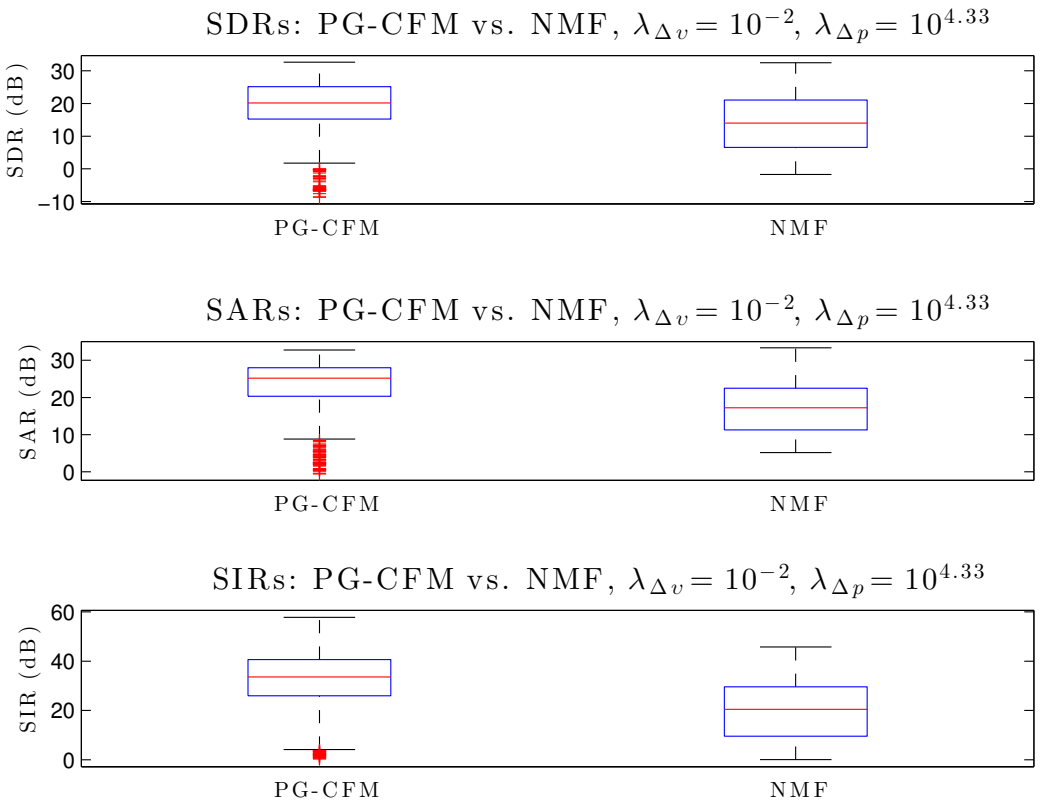

Fig. 5.1 Interquartile ranges for PG-CFM vs. NMF using synthetic dataset

5.1.6 Discussion

Results from the analysis of simulated mixtures of synthetic vibrato sounds show gains of 4.31 dB, 11.58 dB, and 6.7 dB, of PG-CFM over basic SED-NMF in the median SDR,

median SAR, and median SIR values across all 500 trials. While the experiment is non-exhaustive since it considers a class of single-note synthetic vibrato sounds, the result is an encouraging one that suggests that PG-CFM is better-suited to the source separation task where the analyzed sounds contain a vibrato effect. We also see that the lower quartile value Q1 is generally much higher for PG-CFM than NMF: 10 dB higher in the case of median SDR. This suggests that PG-CFM may solve the failure mode of NMF in analyzing vibrato sounds that was illustrated in 3.1. However, the failures of PG-CFM are more severe than those of NMF as is evidenced by the few negative SDR results which can be seen in figure 5.1.

5.2 Experiment 2b: PG-CFM analysis of vibrato singing voice mixtures

5.2.1 Motivation

The use of well-calibrated synthetic signals thus far in the experimentation has permitted a careful inspection of the behaviors of PG-CFM. However, in order to advocate for the use of these features and techniques within a more general musical source separation framework, we also would like to demonstrate their appropriateness for the analysis of real musical sounds. To this end, we describe an application of PG-CFM to a source separation task, where the analyzed sounds are simulated mixtures of recordings of (real) vibrato singing voice, and present results from using NMF on the same task for comparison.

5.2.2 Data

In the analysis of natural sounds, we must acknowledge a limit of PG-CFM in the analysis of sounds in the absence of frequency modulation, where estimated frequency slope $\hat{\xi}$ is close to zero the proposed features Υ thus suffer from numerical issues. Thus, we focus on the analysis of a dataset of musical sounds with a frequency modulation effect. We choose to examine singing vocal vibrato sounds, which are characterized by a strong frequency modulation (Maher and Beauchamp 1990) and thus should be appropriate for analysis by the proposed signal model.

A well-organized dataset of vibrato singing voice sounds was unavailable, so a small dataset of this class of sounds was complied by hand. Raw recordings were gathered

from the MIR-1K dataset[1] (Hsu and Jang 2009) and from recordings[2] of female and male opera singers found on freesound.org (Font, Roma, and Serra 2013). They were edited to isolate parts of the performance subject to a frequency vibrato. The segments of the vocal performance that met this criteria were sustained sung vowel sounds with the vocal vibrato effect. The note onsets and decayed were typically excluded from the edits since they are not well-described by the CFM signal model. The resulting edited recordings were amplitude normalized, and downsampled to 16 kHz sampling rate as needed (to reflect the sampling rate used in the previous experiment for which the regularization parameters $\{\lambda_{\Delta v}\lambda_{\Delta p}\}$ were found). The resulting dataset of isolated source sounds comprises 10 one-second recordings of sung vowel sounds with vibrato; the time-domain and spectrogram representations for one such sound are shown by figure 5.2.

5.2.3 System parameters

The system elements providing the pre-separation feature extraction of the component partials, i.e., the DDM and SMA, were found to be very sensitive to changes in the analysis parameters for the analysis of the dataset of vibrato singing voice sounds. This sensitivity presents a key obstacle in the application of PG-CFM in the analysis of natural sounds, since proper separation in the feature domain requires the extraction of reliable features from the observed data.

We found that we could encourage the extraction of more reliable features by hand tuning the analysis parameters to the analyzed sound prior to the separation. An example of such a hand tuning is shown by figures 5.3 and 5.4. Figure 5.3 shows the local frequency-slope-to-frequency ratio features extracted from a vibrato vocal source in isolation (i.e., not a simulated mixture) using the default parameters given in table 4.1, which are apparently quite noisy, which can be attributed to the incorrect tracking of noise above 2 kHz as component partials by SMA. Figure 5.4 shows features extracted from the same sound when the SMA minimum partial detection threshold has been increased from -50 dB to -40 dB. The total number of partials detected (including correctly-tracked partials) decreases

[1] MIR-1K contains recordings of untrained singers singing Chinese pop songs karayoke-style by the graduate students in the lab, who are presumably untrained singers.

[2] in particular, `http://www.freesound.org/people/digifishmusic/sounds/84243/` and `https://www.freesound.org/people/NoiseCollector/sounds/62103/`, which are recordings of female and male opera singers.

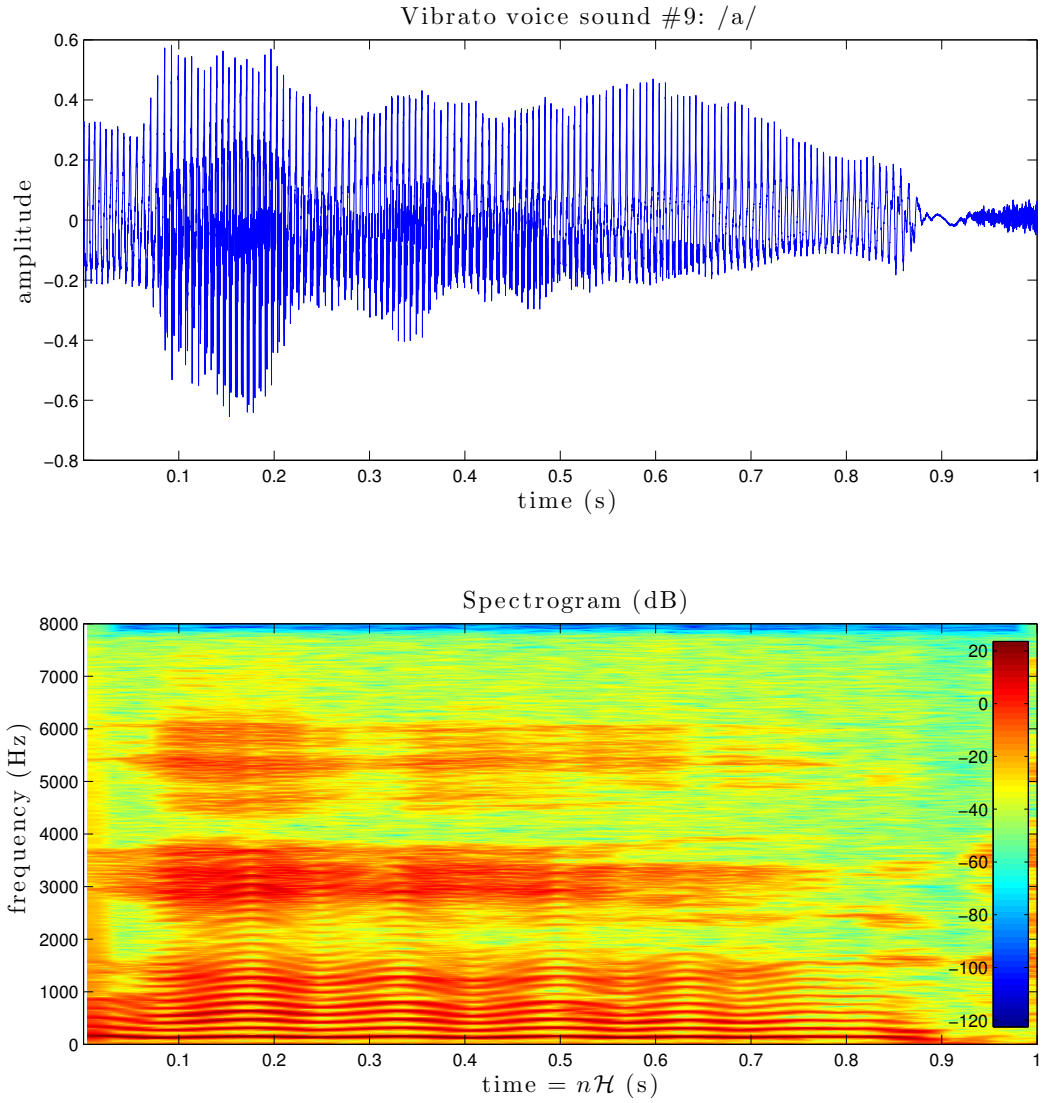

Fig. 5.2 Time-domain and spectrogram representations for vibrato voice sound in isolation: /a/

as a results of the adjustment, but the resulting features Υ are much cleaner and coherent across same-source partials.

In general we were able to produce similar "cleaned" extracted features for each of the 10 vibrato vocal sources in the dataset by hand-tuning the analysis parameters, which typically involved adjusting the minimum detection threshold but in several cases involved additional adjustments to the minimum partial duration or maximum frequency deviation parameters.

Analysis parameters related to STFT are kept the same for every analysis as those specified by table 4.1. Separation parameters are set to $\{\lambda_{\Delta p}, \lambda_{\Delta v}\} = \{10^{-2}, 10^{4.33}\}$, which were determined in experiment 1 to be "consistently good" in the analysis of simulated mixtures of synthetic vibrato sounds.

5.2.4 Procedure

Prior to the simulated mixing of the vibrato vocal sources, the hand-tuned analysis parameters, one for each source, were set aside. Mixtures of vibrato vocal sounds were simulated by the same procedure as in experiments 1 and 2, namely, by linear superposition with 0.75 seconds overlap for a resulting simulated mixture of 1.25 seconds. The data set of 10 isolated source sounds permits a unique combination of 45 unique two-source combinations. For this application we consider "source A mixed with source B" and "source B mixed with source A" to be unique source separation tasks, since we use the set-aside hand-tuned analysis parameters for source A in the first case and those for source B in the second case. Therefore, a total of $2 * 45 = 90$ simulated mixtures were considered in the source separation task.

PG-CFM and NMF were performed on for each of the 90 simulated mixtures produced from the singing voice vibrato dataset with 5 repetitions, for a total of 450 trial evaluations for each of the two algorithms considered. Unlike experiments 1 and 2, where the analysis parameters were fixed across trials, the analysis parameters in this application varied per trial, and were specified to the set of values hand tuned to the first source in the mixture. The `BSS_EVAL` metrics were computed for each of the 450 trials. As before, these metrics were averaged across sources for each trial, so that the algorithm performance is assessed during each trial on its overall average performance rather than the best-source performance.

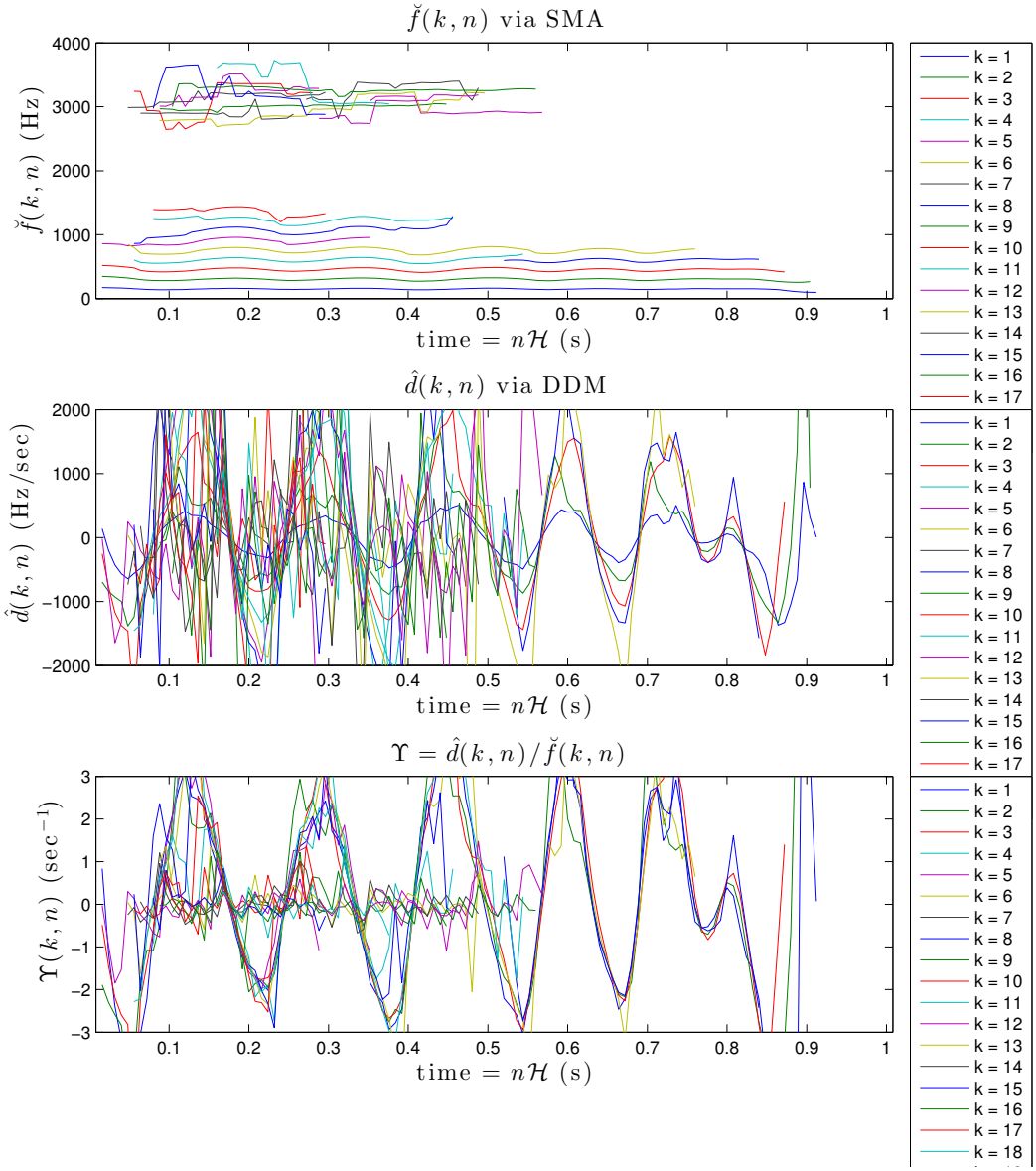

Fig. 5.3 Features extracted from /a/ sound using analysis parameters from chapter 4

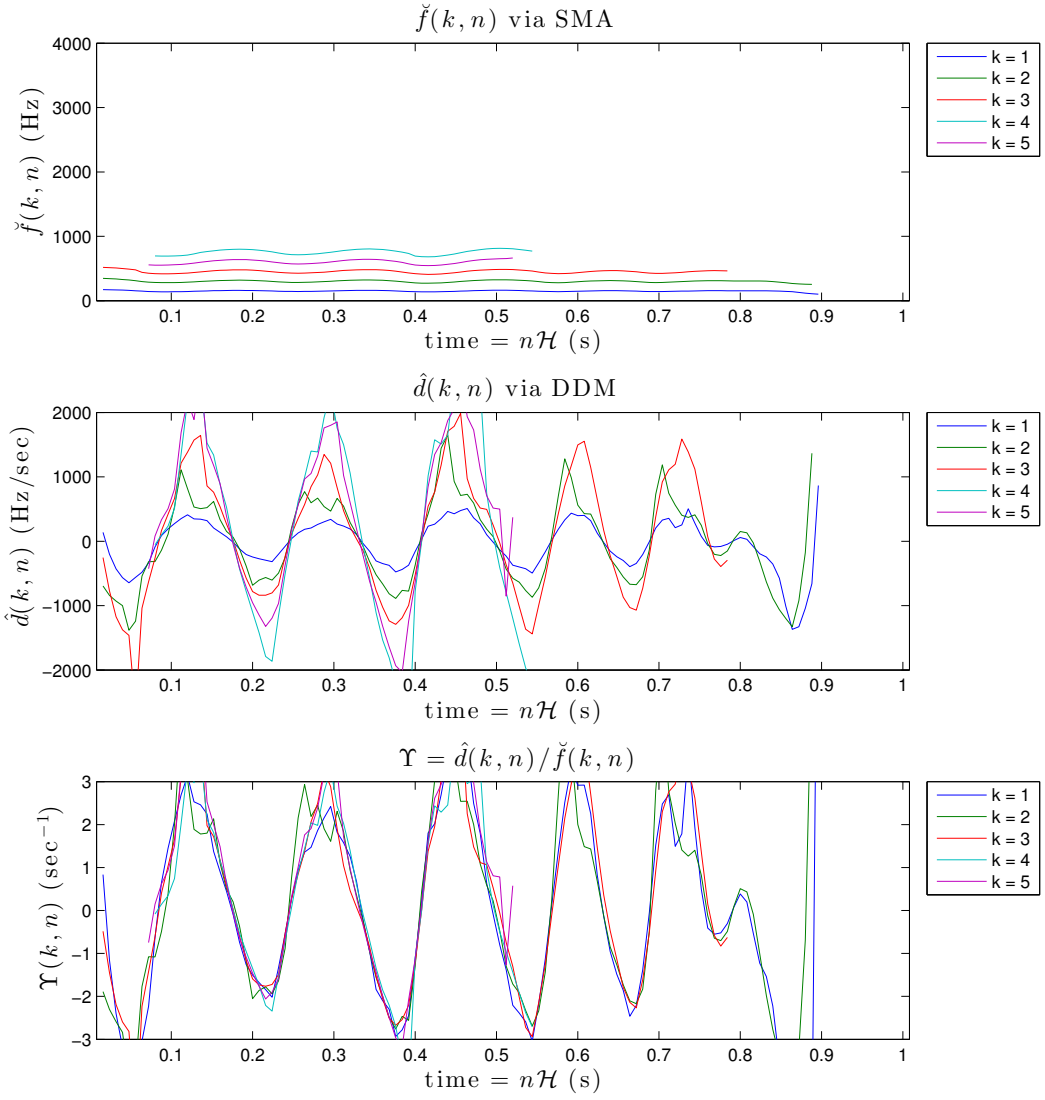

Fig. 5.4 Features extracted from /a/ sound using hand-tuned analysis parameters, partial detection threshold $\mathsf{Thr} \rightarrow$ -40 dB

5.2.5 Results

For each algorithm, the interquartile ranges of the 450 evaluation results are shown in table 5.2. These ranges are illustrated by the box plot shown in figure 5.5.

	Median [Q1, Q3]		
Algorithm	**SDR**	**SIR**	**SAR**
PG-CFM	9.61[0.89, 12.14]	19.28[9.91, 23.35]	12.30[7.17, 15.62]
NMF	7.34[2.04, 10.06]	11.73[5.27, 19.64]	9.20[6.25, 11.95]

Table 5.2 PG-CFM vs. NMF for source separation of mixtures of vibrato singing voice sounds

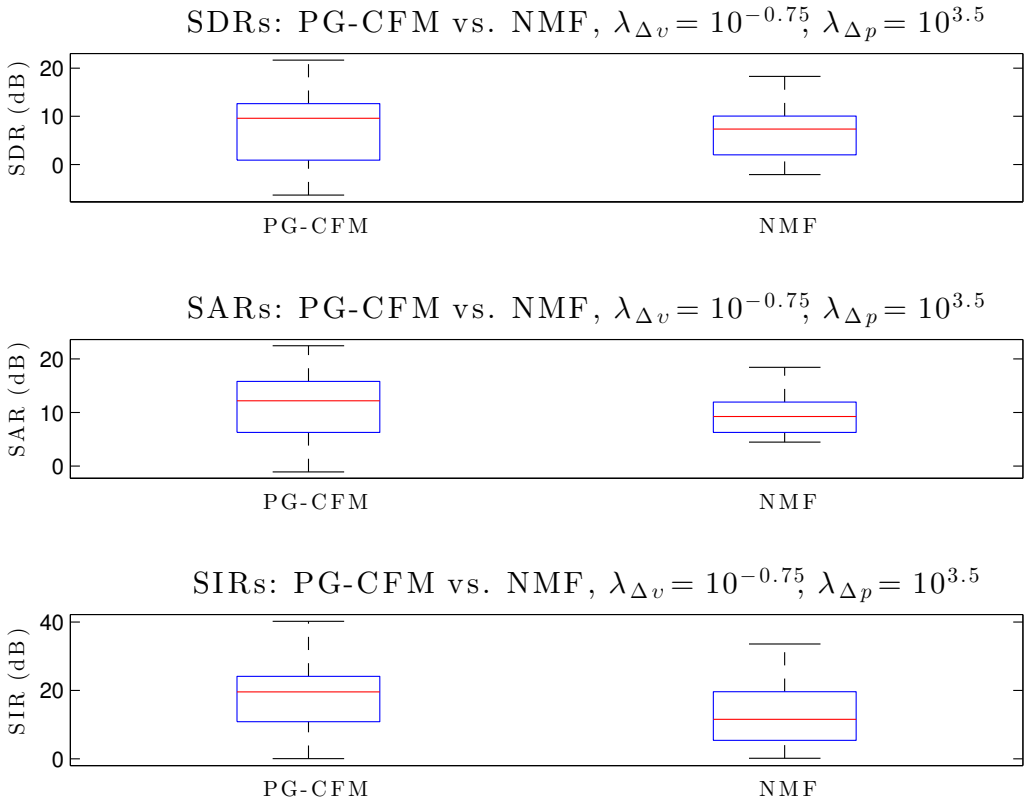

Fig. 5.5 Interquartile ranges for PG-CFM vs. NMF using vibrato singing voice dataset

5.3 Discussion

Both PG-CFM and NMF performed poorly on the application to vibrato vocal source separation, when compared with the results from experiment 2. PG-CFM apparently performs marginally better than NMF across the 450 trials, with gains of 2.26 dB, 7.53 dB, and 3.10 dB in median SDR, median SIR, and median SAR, respectively. Looking at the box plot in figure 5.5, we see that PG-CFM yields a larger range of results than NMF. Interestingly, PG-CFM yields much better best-case results than NMF, but does not perform well consistently. Unlike with experiment 2, we see that PG-CFM has a small value for its lower-quartile cutoff, which suggests a higher failure rate in this application. While the small gains of PG-CFM over NMF in the median `BSS_EVAL` results are somewhat encouraging, the application to real musical sounds is somewhat inclusive, especially considering the hand-tuning of analysis parameters that was required to facilitate the separations.

Sensitivity of the feature extraction blocks in PG-CFM to analysis parameters, which may be data-dependent, is a major problem for PG-CFM as a practical source separation tool since it reduces the robustness of the technique with regard to handling a variety of data (e.g., of different noise floors). One possible future research direction would be to attempt an automated approach to the setting of the analysis parameters, e.g., estimation of the level of stationary noise in the recording.

However, the necessity of hand-tuning analysis parameters in this application suggests a potentially more serious issue with PG-CFM, which stems from its reliance on non-stationary sinusoidal model for not only the extraction of features, but also the estimation of sources by resynthesis. This process of cleaning the features by hand essentially reduces the fidelity of signal representation, since it tends to reduces the number of partials in the non-stationary sinusoidal model. This is because, in general, the component partials of singing voice decrease in energy as their frequency increases. Therefore, by increasing the minimum threshold for detection in SMA we implicitly throw away the higher-register partials.

An example of this phenomenon is seen in figures 5.3 and 5.4, where the SMA analysis with hand-tuned features yields better features but also contains just a few partials. Since the source separation task is equivalent to a partial grouping task from the perspective of PG-CFM, and since the algorithm cannot group partials absent from the signal representation, the PG-CFM may estimate sources of a low fidelity in cases where hand-tuning of the

analysis parameters is required, since the signal model comprises only a few partials. It is possible that building a harmonicity constraint into the source model could help to discern higher-register partials that are closer in amplitude to the ambient noise, since we would know in what spectral neighborhoods to expect partials. However this would likely require the estimation or a priori knowledge of the fundamental frequency for each source, which is a challenging problem for the case of polyphonic mixtures (Yeh, Roebel, and Rodet 2010).

The application of PG-CFM to the analysis of vibrato singing voice mixtures revealed SMA and DDM as useful but fragile feature extraction tools, which provided meaningful features for real musical sounds, but are sensitive to data-dependencies such as the level of stationary noise in the recording. The attempt to separate sources by a direct grouping of partials using the extracted features demonstrated a moderate benefit of the proposed method over NMF, but also suggested that the estimation of vibrato sources in high fidelity would require source separation technique with a more robust signal model.

5.4 Conclusion

This chapter evaluated the performance of PG-CFM on two monaural musical source separation tasks, and provided a comparison to NMF in each case. First, an experiment concerning simulated mixtures of synthetic vibrato sounds was carried out, which showed moderate gains of PG-CFM over NMF. An application to the analysis of simulated mixtures of vibrato voices showed small gains of PG-CFM over NMF, but also revealed a fragility in SMA, the analysis tool used to provide local parameter estimates of the component partials under an additive sinusoidal model, which in turn affects the local frequency slope estimates provided by DDM. The sources estimated by PG-CFM were shown to be of an unsatisfactory fidelity due to the hand-tuning of analysis parameter required to address this issue.

Chapter 6

Structured NMF with CFM constraints

The proposed PG-CFM algorithm was shown in chapters 4 and 5 to provide some benefit over NMF in the analysis of vibrato signals in a source separation task. In this chapter we propose an extension of the NMF, which incorporates the local frequency slope information provided by SMA and DDM, and hopefully provides gains over basic NMF in the analysis of sounds where the CFM signal model is applicable, while proving more robust than than PG-CFM in the analysis of natural sounds. This algorithm can be categorized as both a structural NMF extension and a constrained NMF extension. An application to the analysis of a simulated mixture of synthetic vibrato sounds is then discussed.

6.1 Motivation

The proposed local frequency-slope-to-frequency ratio feature Υ, introduced in section 3.3.4, is evidently useful as a cue in the correct grouping of partials by source, as was demonstrated in previous chapters. The PG-CFM signal model, which assumes a nonzero frequency modulation applied coherently to each partial within a given source, was shown to be appropriate in the analysis of short excerpts of sung vowel sounds with vibrato. Sounds of this class occur briefly over the course of an entire vocal phrase, however, and PG-CFM is not robust to the analysis of sounds in the absence of such a modulation since the frequency slope ξ evaluates to zero or near-zero, resulting in numerical issues in the computation of the frequency-slope-to-frequency ratio Υ. Furthermore, proper separation

with PG-CFM assumes that *every* source in the mixture has some nonzero and CFM on its partials. The experiments presented in chapter 5 examined the case where two vibrato voice sounds overlap, but the more likely application of interest involves the extraction of a single vibrato voice sound from a noisy environment, or from accompanying piano or orchestral parts. PG-CFM may fail in this case since the signal model does not fit every competing source in the auditory scene.

6.2 Proposed NMF extension

Can the NMF decomposition, with its fixed spectral templates, be extended to accommodate the analysis of sounds containing frequency modulations at subaudible rates? We address this question by proposing restructured NMF decomposition, which must explain the analysis from the non-stationary sinusoidal model by construction. This approach represents a unification of the STFT-based and sinusoidal model-based signal analyses. Additionally, we include several penalty terms in the decomposition, which are designed to encourage certain qualities, including intra-source CFM, in the estimated sources.

We first present a new cost function which comprises the restructured model approximation along with the desired penalty terms. We then propose an algorithm, called Nonnegative Matrix Factorization with Coherent Frequency Modulation constraints (NMF-CFM), which minimizes the proposed cost function subject to the appropriate constraints. NMF-CFM is both a structural and constrained extension of the basic NMF.

6.3 Cost function formalization

6.3.1 Model fit

We begin with a restructuring of the model approximation. The observed spectrogram is approximated as the sum of contributions from R nonnegative sources, each of which is characterized by low-rank spectral templates and temporal activations as with NMF. Additionally, each source in the model must explain some frequency estimates of the component partials provided by SMA, where "explanation" is formalized by weighting the estimated sources (i.e., $\{w^r, h^r\}$ factor-pairs) by a tensor. This tensor incorporates (a) a notion of the "likelihood" that source r explains the SMA data of partial k at hop n that was used in PG-CFM, notated by $p^r(k, n)$, and (b) a function, derived from spectral shape of the

STFT analysis window, which permits the comparison of the STFT and SMA analyses in the same domain. The approximate spectrogram under this model is expressed as

$$\hat{X}(f,n) = \sum_{r=0}^{R-1} w^r(f) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) \tag{6.1}$$

where $\frac{f}{\mathcal{N}}$ is the f-th FFT bin in normalized frequency, and $\breve{f}(k,n)$ is the SMA frequency estimate expressed for tracked partial k at hop n, also expressed in normalized frequency. $H(\cdot)$ is the magnitude of the Fourier transform of the analysis window $\boldsymbol{w}$, normalized such that $H(0) = 1$. The spectral templates $w^r(f)$ and temporal activations $h^r(n)$ for a given source only contribute to the approximate spectrogram for time-frequency bins $\{f,n\}$ where $\sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))$ evaluates to 1. This occurs when both $p^r(k,n) \approx 1$ and $\frac{f}{\mathcal{N}} - \breve{f}(k,n) \approx 0$, which is to say that the likelihoods associated with source r explain one[1] of the SMA frequency estimates $\breve{f}(k,n)$ at time-frequency bin $\{f,n\}$.

In other words, source r contributes to the approximation $\hat{X}(f,n)$ only when both its associated factor-pair $w^r(f)h^r(n)$ and the likelihood $p^r(k,n)$ "agree". The factor-pair relates to the STFT analysis in the sense that it is a low-rank spectrogram, and the likelihood relates to the sinusoidal model analysis, so this "agreement" in an optimization sense, is a negotiation between the two modes of analysis, and the model fits the data only if it accounts for both analyses.

The normalized spectral shape of the analysis window is expressed as

$$H(\nu) = \frac{1}{\mathcal{W}} \left| \frac{1}{2} H_{rect}(\nu) - \frac{1}{4} H_{rect}(\nu + \frac{1}{M-1}) - \frac{1}{4} H_{rect}(\nu - \frac{1}{M-1}) \right| \tag{6.2}$$

where $\nu \in [0,1]$ is a normalized frequency, $\mathcal{W}$ is the analysis window normalization term, which for the the Hann window is given by A.5, and $H_{rect}(\nu)$ is the spectral shape of the length-M rectangular window (Harris 1978), expressed as

$$H_{rect}(\nu) = e^{-j\pi(\nu-1)} \frac{\sin(M\pi\nu)}{\sin(\pi\nu)}. \tag{6.3}$$

[1]We implicitly assume no more than one dominant partial per time frequency patch, whose width is given by the main lobe width of the analysis window magnitude spectrum, so that $H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) \leq 1$. This is a stronger assumption than that of basic NMF, which assumes no more than one dominant source per time-frequency bin. However, it is the same as the implicit assumption made by SMA since the spectral resolution is set by the main lobe width of the analysis window.

This model approximation is not a true factorization as with NMF. Some practical consequences of this results are discussed in section 6.7.4, page 92.

We use the SED as a measure of fit of the approximate spectrogram $\hat{X}(f,n)$ to the observed spectrogram $X(f,n)$, expressed as

$$d_{SED}(\mathbf{X}_{f,n}|\hat{\mathbf{X}}_{f,n}) = (X(f,n) - \hat{X}(f,n))^2. \tag{6.4}$$

This distance measure is chosen for its convexity and simplicity with regards to gradient evaluation. The scalar cost is the sum of element-wise SED of the observed spectrogram to its model approximation, expressed as

$$D(\mathbf{W},\mathbf{H},\boldsymbol{\upsilon},\boldsymbol{p}) \triangleq D_{SED}(\mathbf{X}|\hat{\mathbf{X}}) = \sum_{f=0}^{F-1}\sum_{n=0}^{N-1} d_{SED}(\mathbf{X}_{f,n}|\hat{\mathbf{X}}_{f,n}). \tag{6.5}$$

6.3.2 Coherent frequency modulation constraint

We formulate a scalar penalty function to encourage CFM for each of the estimated sources, which takes the same form as equation 3.31, page 44, and is expressed as

$$C_{CFM}(\boldsymbol{p},\boldsymbol{\upsilon}) = \sum_{r=0}^{R-1}\sum_{k=0}^{K-1}\sum_{n=0}^{N-1} p^r(k,n)(\Upsilon(k,n) - \upsilon^r(n))^2. \tag{6.6}$$

6.3.3 Spectral templates smoothness constraint

We formulate a scalar penalty function to encourage smoothness in the spectral templates $w^r(f)$, which is expressed as

$$C_{\Delta w}(\mathbf{W}) = \lambda_{\Delta w}\sum_{r=0}^{R-1}\sum_{f=1}^{F-1}(w^r(f) - w^r(f-1)).^2 \tag{6.7}$$

This penalty formulation is motivated by the apparent inability of the non-statinonary sinusoidal model to properly estimate the high frequency components of sources, which was discussed in section 5.3. In the absence of this penalty function, NMF-CFM is encouraged to choose factor pairs $\{w^r(f), h^r(n)\}$ for the r-th estimated source with energy only in time-frequency bins where a partial likely explain that source (formalized by $p^r(k,n)$) was tracked. In other words, unpenalized NMF-CFM will tend to prefer source estimates

resemble those of PG-CFM. By including this penalty term we discourage NMF-CFM from setting its spectral templates to zero in the high frequency bins, particularly when a partial was tracked in a nearby (usually lower) bin. This hopefully allows NMF-CFM to model signal components in the higher frequencies where partial tracks from SMA are unavailable, e.g., due to a high noise floor.

6.3.4 Likelihoods smoothness constraint

We formulate a scalar penalty function to encourage smoothness in the likelihoods $p^r(k,n)$, which takes the same form as 3.33 and is expressed as

$$C_{\Delta p}(\boldsymbol{p}) = \sum_{r=0}^{R-1}\sum_{k=0}^{K-1}\sum_{n=1}^{N-1}(p^r(k,n) - p^r(k,n-1))^2. \tag{6.8}$$

6.3.5 Ratio features smoothness constraint

We formulate a scalar penalty function to encourage smoothness in the estimated ratio features $\upsilon^r(n)$, which takes the same form as 3.32 and is expressed as

$$C_{\Delta \upsilon}(\boldsymbol{\upsilon}) = \sum_{r=0}^{R-1}\sum_{n=1}^{N-1}(\upsilon^r(n) - \upsilon^r(n-1))^2. \tag{6.9}$$

6.3.6 Cost function formulation

The overall scalar cost is expressed as

$$\begin{aligned} J(\mathbf{W},\mathbf{H},\boldsymbol{\upsilon},\boldsymbol{p}) = \;& D(\mathbf{W},\mathbf{H},\boldsymbol{\upsilon},\boldsymbol{p}) \\ &+ \;\; \lambda_{CFM}C_{CFM}(\boldsymbol{p},\boldsymbol{\upsilon}) + \lambda_{\Delta w}C_{\Delta w}(\mathbf{W}) \\ &\quad + \lambda_{\Delta p}C_{\Delta p}(\boldsymbol{p}) + \lambda_{\Delta \upsilon}C_{\Delta \upsilon}(\boldsymbol{\upsilon}). \end{aligned} \tag{6.10}$$

6.4 Optimization problem

The optimization problem for finding the best model approximation of the observed spectrogram amounts to a minimization of the cost function expressed by 6.10, subject to the appropriate constraints. In particular, the spectral templates $w^r(f)$ and temporal activations $\upsilon^r(n)$ must be nonnegative, and $p^r(k,n)$ is constrained as in PG-CFM, so that it

is nonnegative and always sums across parts to one. The optimization problem is thus expressed as

$$\begin{aligned} \underset{\mathbf{W},\mathbf{H},\boldsymbol{\upsilon},\boldsymbol{p}}{\text{minimize}} \quad & J(\mathbf{W},\mathbf{H},\boldsymbol{\upsilon},\boldsymbol{p}) \\ \text{subject to} \quad & p^r(k,n) \geq 0 \,\forall\, r,k,n, \\ & \sum_r^R p^r(k,n) = 1 \,\forall\, \{k,n\}, \\ & w^r(f) \geq 0 \,\forall\, \{r,f\}, \\ & h^r(n) \geq 0 \,\forall\, \{r,n\}. \end{aligned} \tag{6.11}$$

6.5 Algorithm

We propose an algorithm, called NMF-CFM, which finds a locally optimal solution to the optimization problem expressed by equation 6.11 by the familiar routine of alternating minimization of the variables $\{p,\upsilon,w,h\}$. During the update to each of these four variables, the other three are held fixed to their previously assigned value. As with PG-CFM, p and υ are updated by the solution to an equivalent QP. For the update to p, the equivalent QP is convex, which permits and efficient solution using an existing solver. For the update to υ, the equivalent QP is convex and unconstrained, which permits a closed-form solution. The w and h are updated multiplicatively. The multiplicative updates are derived as in (Lee, Hill, and Seung 2001), which was discussed in section 2.7.3, with details provided in appendix C NMF-CFM is expressed by algorithm 6.1.

6.5.1 Multiplicative update of w

The spectral templates are updated multiplicatively as

$$w^r(f) \leftarrow w^r(f)\frac{\sum_{n=0}^{N-1} X(f,n)h^r(n)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n)) + \lambda_{\Delta w}\varkappa(w^r(f))}{\sum_{n=0}^{N-1} \hat{X}(f,n)h^r(n)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n)) + \lambda_{\Delta w}\varrho(w^r(f))} \tag{6.12}$$

with

$$\varrho(w^r(f)) = \begin{cases} 2w^r(f) & f = 0 \\ 4w^r(f) & 0 < f < F-1 \\ 2w^r(f) & f = F-1 \end{cases} \tag{6.13}$$

and

$$\varkappa(w^r(f)) = \begin{cases} 2w^r(f+1) & f = 0 \\ 2w^r(f-1) + 2w^r(f+1) & 0 < f < F-1 \\ 2w^r(f-1) & f = F-1. \end{cases} \tag{6.14}$$

6.5.2 Multiplicative update of h

The temporal activations are updated multiplicatively as

$$h^r(n) \leftarrow h^r(n) \frac{\sum_{f=0}^{F-1} X(f,n) w^r(f) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n))}{\sum_{f=0}^{F-1} \hat{X}(f,n) w^r(f) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n))}. \tag{6.15}$$

cf. appendix C for a derivation of the multiplicative updates to both w and h.

6.5.3 Scaling

We include a normalization step since it prevents the spectral templates from blowing up proportional to the number of iterations. In particular, we rescale the factors so that $w^r(f)$ sums to unity across all bins in each part, i.e.,

$$w^r(f) \leftarrow \frac{w^r(f)}{\sum_{\nu=0}^{F-1} w^r(\nu)}, \tag{6.16}$$

$$h^r(n) \leftarrow h^r(n) \sum_{\nu=0}^{F-1} w^r(\nu). \tag{6.17}$$

Some consequences of this decision are discussed in section 6.5.4

6.5.4 Convergence

Updates to p and v are local optima (with all other variables fixed), while the multiplicative updates to w and h guarantee nonincreasing cost and satisfaction of the constraints. In the absence of a post-update normalization step, we can guarantee convergence to a locally stable solution in the limit $\eta \to \infty$. However, we sacrifice a convergence guarantee by including the scaling step. This is not the case with the basic NMF, but in our case the cost may increase slightly as a result of the scaling, which cannot be applied to the

likelihoods without violating their constraints. In practice we observe that such increases due to the normalization do not affect the overall convergence behavior.

6.5.5 Initialization

In practice, NMF-CFM performance was found to be quite sensitive to initialization of the algorithm variables $\{\mathbf{H}, \mathbf{W}, \boldsymbol{p}, \boldsymbol{v}\}$. For example, the update of $\boldsymbol{p}$ on the first iteration, when $\mathbf{W}$ and $\mathbf{H}$ were initialized to random nonnegative values, the first updates to $\boldsymbol{p}$ and $\boldsymbol{v}$ would sometimes yield values for $\{\boldsymbol{p}, \boldsymbol{v}\}$ that, unlike a good PG-CFM solution, did not well explain the observed features $\boldsymbol{\Upsilon}$, and moreover could not be "undone" in the following iterations.

The remaining algorithm factors $\{\mathbf{W}, \mathbf{H}\}$ are then updated according to gradients of variables which do not properly encode the CFM of the sources, which leads to a NMF-CFM solution after η that, while representing a stationary point of the cost function, yields a poor separation. To address this issue, we propose an initialization scheme called NMF-CFM-init, which is expressed by algorithm 6.2. The idea is to initialize $\{\boldsymbol{p}, \boldsymbol{v}\}$ to a PG-CFM solution and run several iterations of multiplicative updates to $\mathbf{W}$ and $\mathbf{H}$ to allow the spectral templates and temporal activations to reach values which reasonably-well explain the observed data. From here we begin the NMF-CFM procedure.

Some observations on algorithm behavior with and without initialization by NMF-CFM-init are discussed in section 6.7.

6.5.6 Block diagram

A block diagram for the proposed PG-CFM system is given by figure 6.1

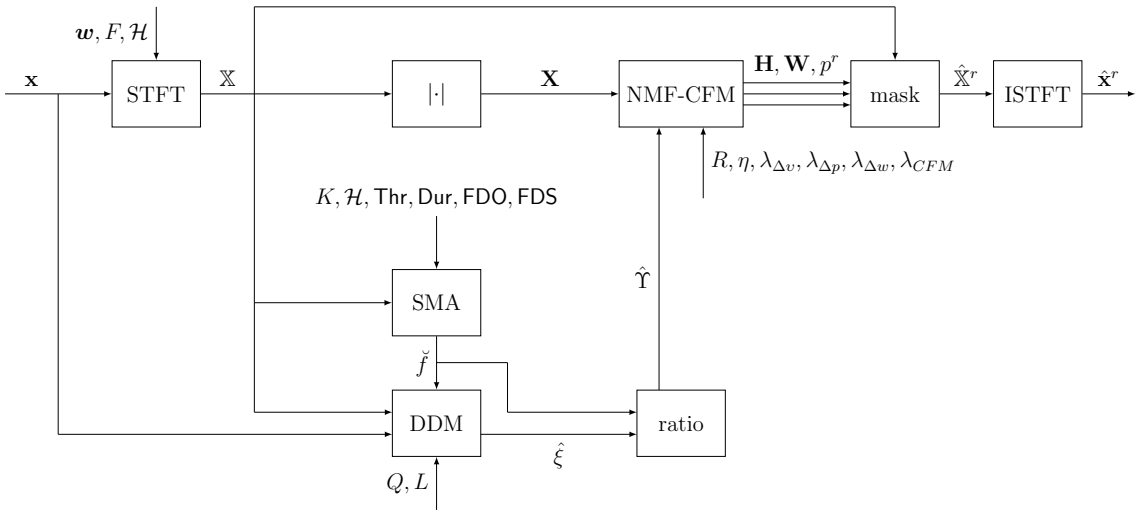

Fig. 6.1 NMF-CFM block diagram

Algorithm 6.1: NMF-CFM with multiplicative updates

Input: $\boldsymbol{\Upsilon}, \breve{\boldsymbol{f}} \in \mathbb{R}^{K\times N}$, R, η,$\lambda_{CFM}, \lambda_{\Delta p}, \lambda_{\Delta\upsilon}, \lambda_{\Delta w}$

Output: $\boldsymbol{p} \in \mathbb{R}^{R\times K\times N}_{\geq 0}$, $\boldsymbol{\upsilon} \in \mathbb{R}^{R\times N}$, $\mathbf{W} \in \mathbb{R}^{F\times R}_{\geq 0}$, $\mathbf{H} \in \mathbb{R}^{R\times N}_{\geq 0}$

initialize $j = 1$ and $\mathbf{W}, \mathbf{H}, \boldsymbol{p}, \boldsymbol{\upsilon}$ such that

$\mathbf{W} \in \mathbb{R}^{F\times R}_{\geq 0}, \mathbf{H} \in \mathbb{R}^{R\times N}_{\geq 0}, \boldsymbol{\upsilon} \in \mathbb{R}^{R\times N}$, $\boldsymbol{p} \in \mathbb{R}^{R\times K\times N}_{\geq 0}$, $\sum_{r=1}^{R} p^r(k,n) = 1 \; \forall \, k, n$;

compute $\tilde{\mathbf{p}}, \tilde{\boldsymbol{\upsilon}}, \tilde{\boldsymbol{\Upsilon}}$ by unfolding $\boldsymbol{p}, \boldsymbol{\upsilon}, \boldsymbol{\Upsilon}$ as column vectors;

compute $\boldsymbol{H} \in \mathbb{R}^{R\times F\times N\times K}$ by equation 6.2;

while $j \leq \eta$ **do**

- compute $\tilde{\boldsymbol{\upsilon}}, \boldsymbol{\Lambda}_p, \tilde{\mathbf{d}}, \mathbf{D}_H, \mathbf{D}_\Xi, \boldsymbol{\Sigma}_r, \boldsymbol{\Sigma}_k, \mathbf{U}_f$;
- $\tilde{\mathbf{p}} \leftarrow \tilde{\mathbf{p}}^*$ by solution to equivalent QP, given by equation C.27 ;
- compute $\boldsymbol{\Lambda}_\upsilon, \mathbf{U}_r, \mathbf{U}_k, \mathbf{D}_p$;
- $\tilde{\boldsymbol{\upsilon}} \leftarrow (\mathbf{U}_k^T \mathbf{D}_p \mathbf{U}_k + \lambda_{\Delta\upsilon} \boldsymbol{\Lambda}_\upsilon^T \boldsymbol{\Lambda}_\upsilon)^{-1} (\mathbf{U}_k^T \mathbf{D}_p \mathbf{U}_r \tilde{\boldsymbol{\Upsilon}})$;
- compute $\boldsymbol{\varrho} \in \mathbb{R}^{F\times R}, \boldsymbol{\varkappa} \in \mathbb{R}^{R\times N}, \boldsymbol{\upsilon} \in \mathbb{R}^{R\times F\times N}$;
- $w^r(f) \leftarrow w^r(f) \frac{\sum_{n=0}^{N-1} X(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n)) + \lambda_{\Delta w} \varkappa(w^r(f))}{\sum_{n=0}^{N-1} \hat{X}(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n)) + \lambda_{\Delta w} \varrho(w^r(f))}$;
- compute $\boldsymbol{\varrho} \in \mathbb{R}^{F\times R}, \boldsymbol{\varkappa} \in \mathbb{R}^{R\times N}, \boldsymbol{\upsilon} \in \mathbb{R}^{R\times F\times N}$;
- $h^r(n) \leftarrow h^r(n) \frac{\sum_{f=0}^{F-1} X(f,n) w^r(f) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n))}{\sum_{f=0}^{F-1} \hat{X}(f,n) w^r(f) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{N} - \breve{f}(k,n))}$;
- normalize $\mathbf{W}$, $\mathbf{H}$ as appropriate;
- $j \leftarrow j + 1$

compute $\boldsymbol{p}, \boldsymbol{\upsilon}$ by reshaping $\tilde{\mathbf{p}}, \tilde{\boldsymbol{\upsilon}}$ to original dimensions;

Algorithm 6.2: NMF-CFM-init with multiplicative updates

Input: $\boldsymbol{\Upsilon}, \breve{\boldsymbol{f}} \in \mathbb{R}^{K\times N}$, R, η_j,$\lambda_{\Delta p}, \lambda_{\Delta v}, \lambda_{\Delta w}$

Output: $\boldsymbol{p}_i \in \mathbb{R}^{R\times K\times N}_{\geq 0}$, $\boldsymbol{v}_i \in \mathbb{R}^{R\times N}$, $\mathbf{W}_i \in \mathbb{R}^{F\times R}_{\geq 0}$, $\mathbf{H}_i \in \mathbb{R}^{R\times N}_{\geq 0}$

initialize $j = 1$ and $\mathbf{W}_i, \mathbf{H}_i$ such that $\mathbf{W}_i \in \mathbb{R}^{F\times R}_{\geq 0}, \mathbf{H}_i \in \mathbb{R}^{R\times N}_{\geq 0}$, and $\boldsymbol{p}_i \in \mathbb{R}^{R\times K\times N}, \boldsymbol{v}_i \in \mathbb{R}^{R\times N}$ by PG-CFM;

while $i \leq \eta_j$ **do**

compute $\boldsymbol{\varrho} \in \mathbb{R}^{F\times R}, \boldsymbol{\varkappa} \in \mathbb{R}^{R\times N}$;

$$w^r(f) \leftarrow w^r(f)\frac{\sum_{n=0}^{N-1} X(f,n)h^r(n)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n))+\lambda_{\Delta w}\varkappa(w^r(f))}{\sum_{n=0}^{N-1} \hat{X}(f,n)h^r(n)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n))+\lambda_{\Delta w}\varrho(w^r(f))};$$

compute $\boldsymbol{\varrho} \in \mathbb{R}^{F\times R}, \boldsymbol{\varkappa} \in \mathbb{R}^{R\times N}$;

$$h^r(n) \leftarrow h^r(n)\frac{\sum_{f=0}^{F-1} X(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n))}{\sum_{f=0}^{F-1} \hat{X}(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{N}-\breve{f}(k,n))};$$

normalize $\mathbf{W}_i$, $\mathbf{H}_i$ as appropriate;

$j \leftarrow j + 1$

6.6 Comments on the implementation

The NMF-CFM approximate model is not a product of low-rank matrix factors as with the NMF. Although the spectral templates and temporal factorizations are low rank, the overall model representation is not since we must store both the SMA and DDM extracted features in memory. Efficiency of the model representation was a prime motivation in the development of basic NMF, while the resulting interpretability of the estimated sources was secondary, essentially amounting "good news" (Bertin, Badeau, and Vincent 2010). Here we prioritize interpretability and correctness in the estimated sources, and sacrifice efficiency of the model representation in so doing.

NMF-CFM represents an enormous computational requirement compared with PG-CFM, which translates to a burden in either memory or speed depending on the implementation. In particular, the memory-speed tradeoff hinges on whether the tensor $p^r(k,n)H(\frac{f}{N} - \breve{f}(k,n)) \in \mathbb{R}^{R\times F\times N\times K}$ is stored in memory as a pre-processing step, or re-computed in the inner loop of each iteration. In our implementation, we chose to compute this tensor in the pre-processing, which facilitates the examination of convergence behaviors of the algorithm after many (on the order of 50) iterations. This, however, limited the analysis to audio sampled at 16 kHz.

As discussed in chapter 3, transient or breathy sounds are not well-modeled by the sum of non-stationary sinusoids (Serra and Smith 1990). In theory, these sounds are more appropriately modeled by NMF with flat spectral templates. Since the restructuring of the NMF model in the formulation of NMF-CFM incorporates the non-stationary sinusoidal model in the decomposition, it is unlikely to analyze this class of sounds well. This issue could potentially be addressed by incorporating a stochastic model of the sinusoidal model residual, as in (Serra and Smith 1990), into the decomposition.

NMF-CFM involves four separation parameters, $\lambda_{CFM}, \lambda_{\Delta p}, \lambda_{\Delta v}$, and $\lambda_{\Delta w}$, which span a considerably larger parameter space than the two separation parameters of PG-CFM. This consideration, along with the increased computation and memory requirements of NMF-CFM over PG-CFM and the lack of a large dataset of real sounds containing frequency modulations, ultimately deterred from the design of a full parameter sweep experiment to determine the optimal separation parameters, of the type described in chapter 4. Such an experiment could be the focus of future work.

6.7 Experiment 3: CFM-NMF analysis of synthetic vibrato sound mixtures

We here demonstrate some of the merits and limitations of NMF-CFM by presenting an application to a source separation task, where the algorithm parameters are selected by hand. We focus on the analysis of simulated mixtures of (real) vocal vibrato sounds, which was identified in section 5.3 as a case where PG-CFM provides gains over NMF but is ultimately unsatisfactory due to an inability to model partials in the higher frequency ranges. In particular, we examine a single simulated mixture of two vibrato signing voice sources, taken from the dataset described in section 5.2.2. This permits a more thorough examination of the NMF-CFM behaviors with respect to its algorithm parameters (in the absence of a parameter sweep experiment). Moreover, the examination of a single mixture from the dataset limits the influence of noise on the extracted features, which was previously identified as an issue in the analysis of real musical recordings in our dataset. We compare these results with those produced by NMF and PG-CFM. While this application is non-exhaustive, it provides some useful insight into the quality of the sources estimated by NMF-CFM.

6.7.1 Data

Sources 1 and 2 are recordings of an alto voice with singing a "oo" at B3 (fundamental frequency 247 Hz) and "oo" at A4 (fundamental frequency 440 Hz), respectively, with moderate vibrato in both cases. Spectrogram representations (in dB) are given for the two vibrato voice sources in isolation and in simulated mixture by figure 6.2. We display only the frequency bins in the range $f \in [1, 175]$, which corresponds to frequencies less than 2.75 kHz, to enhance the clarity of the presented results, as the partials of interest exist in this range for the sounds considered.

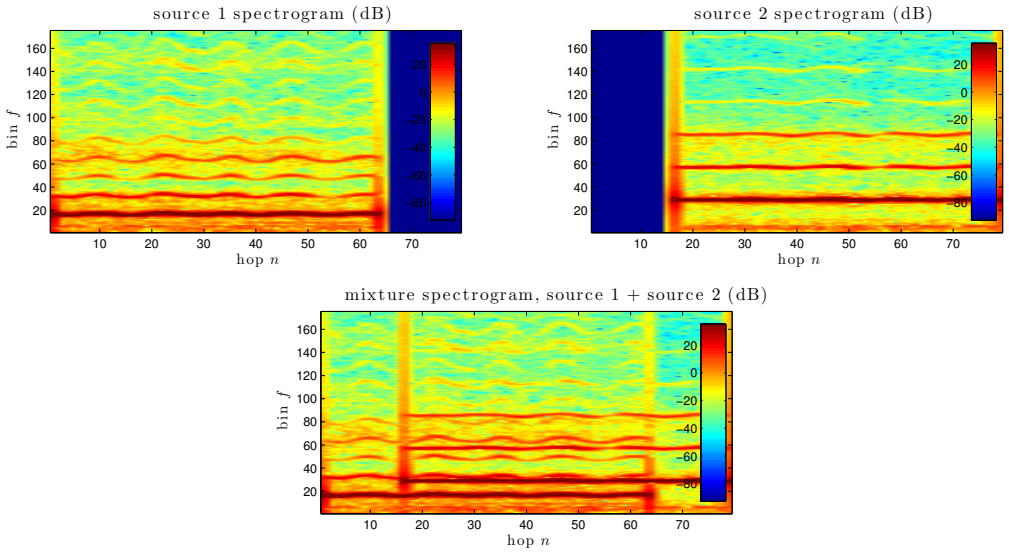

Fig. 6.2 Spectrograms (dB) for vibrato singing voice sources in isolation and mixture

6.7.2 Procedure

Analysis parameters for STFT and DDM were set to the same values as used for previous experiments, which were given in table 4.3. The SMA analysis parameters were specified to set of values hand-tuned to source 1, which in this case were identical to the SMA parameters specified by table 4.3, except for the amplitude detection threshold, which was set to -47 dB.

Four separations were performed using the following algorithms:

- PG-CFM with $\eta = 6, \lambda_{\Delta v} = 10^{-2}, \lambda_{\Delta p} = 10^{4.33}$
- SED-NMF with $\eta = 25$, updates by alternating least-squares
- NMF-CFM-init with $\eta_i = 100, \lambda_{\Delta w} = 10^{7}$, initialized by the previous PG-CFM solution
- NMF-CFM with $\eta = 5, \lambda_{CFM} = 10^{4}, \lambda_{\Delta v} = 10^{-2}, \lambda_{\Delta p} = 10^{4}, \lambda_{\Delta w} = 10^{7}$, initialized by the previous NMF-CFM-init solution

6.7.3 Results

The `BSS_EVAL` metrics for each of the four separations are give by table 6.1. Unlike the results for the previous experiments, which averaged the metrics across sources for each trial, we here present the per-source metrics. Spectrogram representations (in dB) for both estimated sources produced by each of the four algorithms are shown by figure 6.3.

	(Source 1, Source 2)		
Algorithm	**SDR**	**SIR**	**SAR**
PG-CFM	(21.25, 22.03)	(43.25, 37.12)	(21.28, 22.17)
NMF	(15.47, 17.87)	(34.88, 24.31)	(15.53, 19.00)
NMF-CFM-init	(29.08, 22.28)	(41.79, 23.34)	(29.32, 28.98)
NMF-CFM	(21.75, 18.41)	(40.62, 20.78)	(21.8, 22.22)

Table 6.1 Source separation performance for vibrato voice mixture: PG-NMF, NMF, NMF-CFM, and NMF-CFM-init

6.7.4 Discussion

The spectrograms of the sources estimated by PG-CFM illustrate the fidelity problem of the non-stationary sinusoidal model, which was discussed in section 5.3. For example, PG-CFM estimates source 1 as a single partial. Looking at the spectrograms of the NMF-CFM and NMF-CFM-init, we can see the effect of the new signal model in the reproduction; sources estimated by this method are much more capable than PG-CFM of representing the higher partials present in the sources. This is reflected by a subjective evaluation of (i.e., listening to) of the estimated sources;

The NMF-CFM signal model seems more capable than that of PG-CFM, but does not yield a perfect separation in this application, as is illustrated by the presence of some

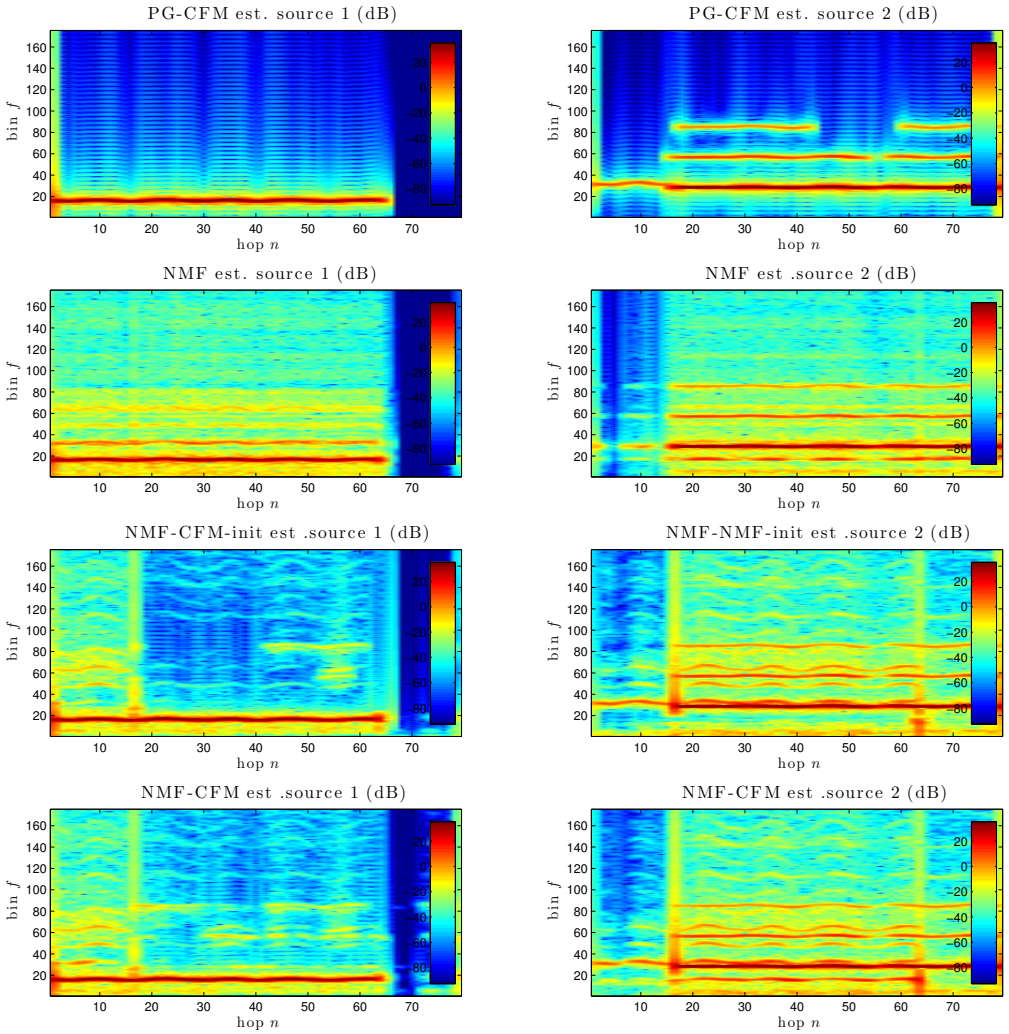

Fig. 6.3 Source 1 and 2 estimated spectrograms (dB) by PG-CFM, NMF, NMF-CFM, and NMF-CFM-init

source-1 overtones above bin 100 in source-2 estimations from both NMF-CFM and NMF-CFM-init (we also observe this behavior in the NMF estimated sources).

It is very interesting to note that, in terms of the `BSS_EVAL` metrics, NMF-CFM-init performs better than any other algorithm on this task, with gains of 13.61 dB and 7.83 dB in SDR over NMF and PG-CFM, respectively. The NMF-CFM iterations updating the full set of algorithm parameters $\{\mathbf{W}, \mathbf{H}, \boldsymbol{p}, \boldsymbol{v}\}$ apparently provide no extra separation benefit after the multiplicative factor updates to $\{\mathbf{W}, \mathbf{H}\}$ provided by NMF-CFM-init.

Although it is unwise to generalize about algorithm behavior from a single example, one hypothesis is we do not benefit from a concurrent separation in both the Υ feature domain and the spectrogram domain. In other words, perhaps the partial grouping provided by PG-CFM best explains the observed sound in the feature domain, and while NMF-CFM-init (which was initialized by PG-CFM) does not alter the source estimates in the feature domain, it provides decent low-rank spectral templates and temporal activations which enable source estimates in a higher fidelity than is possible in with the implicit sinusoidal model of PG-CFM.

6.8 Conclusion

This chapter presented NMF-CFM, a novel NMF extension designed for the separation of frequency-modulated musical sources from a monaural recording. NMF-CFM restructures the NMF model so that the observed spectrogram is approximated by a non-stationary sinusoidal in addition to the low-rank spectral templates and temporal activations typical of NMF. Intra-source CFM and smoothness in the algorithm parameters are encouraged in the NMF-CFM decomposition by the inclusion of several penalty terms in the formulated cost function. An example analysis of a vibrato singing voice mixture, and some algorithm behaviors and issues in the implementation were discussed. NMF-CFM was observed to resolve some of the fidelity issues in source estimation characteristic of PG-CFM, but interestingly was unable to surpass the performance of its initialization algorithm NMF-CFM-init on the example application.

Chapter 7

Conclusion

This chapter summarizes the main contributions of this thesis and suggests several possible directions for future research on musical source separation using a CFM source model.

7.1 Summary of contributions

Chapter 3 discussed the local frequency-slope-to-frequency ratios using SMA and DDM in tandem within the framework of a non-stationary additive sinusoidal signal model. These ratio features were then shown to be useful in the grouping of the sinusoidal partials under a source model containing a CFM term, and an algorithm to provide such a grouping, called PG-CFM, was described.

Chapter 4 discussed the PG-CFM algorithm parameters, and described a parameter sweep experiment designed to determine the ideal set of separation parameters via the analysis of simulated mixtures of synthetic vibrato sounds with note values in the range [C4, C5]. Two pairs of parameters that performed consistently well on the synthetic data were identified.

In chapter 5, the separation performance of PG-CFM was evaluated using one of the pairs of optimal separation parameters found in the previous experiment and compared with the results from an NMF-based source separation. In the analysis of 500 simulated mixtures of synthetic vibrato sounds with notes in the range [A0, C8], PG-NMF was found to provide a moderate gain over NMF. In the analysis of real vibrato singing voice mixtures, PG-CFM provided a small benefit over NMF but produced estimated sources of a low fidelity due to the underlying additive sinusoidal model, which was unable to reproduce

high-frequency signal components in the estimated sources. This application also revealed SMA and DDM to be fragile in the extraction of local frequency-slope-to-frequency ratio features from natural sounds.

In chapter 6, an extension to NMF was proposed that could hopefully leverage the success of PG-CFM in using the extracted ratio features while avoiding the apparent pitfalls of the additive sinusoidal model in the estimation of sources. The proposed algorithm, called NMF-CFM, uses a restructured NMF model that takes into account the sinusoidal signal components tracked by SMA in its approximation of the observed spectrogram. Intra-source CFM, along with smoothness in the model parameters, is encouraged in the estimated sources by way of penalty terms.

The application of NMF-CFM to the analysis of a single simulated mixture of two vibrato singing voice sounds was then discussed in detail. This example illustrated a benefit of approximating sources in the spectrogram domain in the correct identification of higher-frequency partials. Interestingly, the algorithm for initializing was found to outperform NMF-CFM in the separation task, which may suggest that the estimation of spectral templates and temporal activations is best kept separate from the task of grouping partials by CFM cues.

7.2 Future work

There are several steps that could be taken to directly advance the research presented in this thesis. For example, while this thesis focused on the analysis of musical sounds with frequency vibrato, the algorithms presented herein could be extended to musical sounds with glissando or pitch bending effects. It is likely that the convergence speed of NMF-CFM could be greatly improved by updating factors with Majorization-Minimization, as in (Févotte 2011), which provides an updated factor that globally optimizes a function majorizing the cost function, which likely reduces the cost much more than a single multiplicative update. Also, the hypothesis presented in chapter 6, that partial grouping and spectrogram factor estimation are best done in separate domains, should be more thoroughly investigated.

Generally speaking, there is a great potential for future research on musical source separation using a CFM source model, since there are a large variety of musical sounds with CFM that are not well-modeled by the basic NMF. Improving the robustness of CFM-

based source separation techniques would likely entail achieving an independence from the sinusoidal model-based tracking tools like SMA, whose parameters needed to be hand-tuned to specific audio input in practice. We first appealed to the additive sinusoidal model since it more directly encodes the frequency modulation when compared with the STFT or PV. However, it is conceivable that the intra-source frequency modulation coherence could be leveraged by a parameter estimation in the STFT domain, which would in turn require a STFT-like CFM signal model.

(Li, Woodruff, and Wang 2009) encoded amplitude modulations per frequency channel in auditory model to produce a nonnegative tensor to be factorized with an NMF-like framework, along with appropriate masking and rectification to ensure that only valid frequency slope estimates are considered, and that the data are nonnegative. It is conceivable that DDM could be employed in the STFT domain to produce a similar tensor signal representation. After all, STFT atoms are used to construct the linear system that provides the DDM parameter estimates.

Appendix A

Analysis/synthesis specifications

A.1 Short-time Fourier Transform

A.1.1 Transform definitions

The STFT is defined as

$$[\text{STFT}\{\mathbf{x}\}]_{f,n} = \mathbb{X}_{f,n} = \frac{1}{\mathcal{W}} \sum_{l=-\frac{\mathcal{N}}{2}}^{\frac{\mathcal{N}}{2}-1} \bar{\mathbf{x}}_l^{(n)} \bar{\boldsymbol{w}}_l \exp\left(\frac{-j2\pi lf}{\mathcal{N}}\right) \tag{A.1}$$

where $\mathcal{N}$ is the number of frequency bins in the Discrete Fourier Transform (DFT), $\bar{\mathbf{x}}^{(n)}$ is the n-th (possibly zero-padded) signal buffer, $\bar{\boldsymbol{w}}$ is the analysis window $\bar{\boldsymbol{w}}$ shifted by $n\mathcal{H}$ samples, $\bar{\boldsymbol{w}}$ is the (possibly zero-padded) normalized analysis window, and $f \in [0, F-1]$ is the set of non-redundant frequency bins, which is discussed further in section A.1.3. We synthesize output audio from the time-frequency domain using the inverse short-time Fourier transform (ISTFT), defined as

$$[\text{ISTFT}\{\mathbb{X}\}]_l = \mathbf{x}_l = \sum_{m=-\infty}^{+\infty} \left(\sum_{n=-\frac{\mathcal{N}}{2}}^{\frac{\mathcal{N}}{2}-1} \mathbb{X}_{f,n} \exp\left(\frac{j2\pi lf}{\mathcal{N}}\right) \right). \tag{A.2}$$

Appropriate analysis parameters must be chosen to ensure proper sampling of $\mathbb{X}$ in both time and frequency (Allen and Rabiner 1977). Synthesis is implemented using the overlap-add technique with an implicit rectangular synthesis window.

A.1.2 Signal buffer and analysis window

The possibly zero-padded n-th signal buffer is defined as

$$\bar{\mathbf{x}}_l^{(n)} = \begin{cases} \mathbf{x}_{m+n\mathcal{H}} & 1 \leq l \leq M \\ 0 & M < l \leq \mathcal{N}. \end{cases} \tag{A.3}$$

We use a length-M normalized Hann window for analysis with zero padding when $\mathcal{N} > M$. The window is expressed as

$$\bar{\boldsymbol{w}}_m = \begin{cases} \frac{1}{\mathcal{W}} \sin^2\left(\frac{\pi(m-1)}{M}\right) & 1 \leq m \leq M \\ 0 & M < m \leq \mathcal{N} \end{cases} \tag{A.4}$$

where $\mathcal{W}$ is the normalization term included so that the window has unity sum[1], i.e,

$$\mathcal{W} = \sum_{m=1}^{M} \sin^2\left(\frac{\pi(m-1)}{M}\right) = \frac{M-1}{2}. \tag{A.5}$$

The first derivative of the window is expressed as

$$\bar{\boldsymbol{w}}'_m = \begin{cases} \frac{\pi}{\mathcal{W}(M-1)} \sin\left(\frac{2\pi(m-1)}{M-1}\right) & 1 \leq m \leq M \\ 0 & \text{else.} \end{cases} \tag{A.6}$$

Source separation by Nonnegative Matrix Factorization necessitates a nonnegative short-time spectral representation, for which the STFT magnitudes, termed *spectrogram*, or STFT square magnitudes, termed *power spectrogram* are often used.

A.1.3 Spectral symmetry

When $\mathbf{x}$ is strictly real, which is satisfied in the case of audio signals, all short-time spectra have a symmetry property, expressed as

$$\boldsymbol{S}_{f,\sigma} = -\boldsymbol{S}_{\mathcal{N}-f,\sigma} \; \forall \; f \in [1, \frac{\mathcal{N}}{2} - 1]. \tag{A.7}$$

[1]The normalized window function also has the property of unity spectral energy in the DC bin, i.e., $[\mathrm{DFT}\{\mathbf{x}\}]_0 = 1$.

In other words, the STFT for frequency bins $f > \frac{\mathcal{N}}{2}$ are redundant to the signal representation as they can be reproduced prior to the resynthesis without loss of information. Therefore, in an NMF-like analysis of the STFT magnitudes, we discard these redundant frequency bins and consider only the STFT magnitudes in the range $f = [0, F-1]$ where F number of non-redundant bins, given by

$$F = \frac{\mathcal{N}}{2} + 1. \tag{A.8}$$

A.2 Additive synthesis

An overlap-add additive synthesis scheme is used to generate a time-domain estimate of the analyzed sound according to the model representation, i.e., the per-frame frequency and amplitude estimates from the analysis. The n-th frame of partial k is synthesized as

$$\hat{x}_k^n(m) = \sum_{k=1}^{P} \hat{A}_k^n(m) \cos(\hat{\phi}_k^n(m)) \tag{A.9}$$

where $\hat{A}_k^n(m)$ and $\hat{\phi}_k^n(m)$ are the amplitude and phase estimates for partial k at frame n evaluated at sample m. They are derived by a linear interpolation of the SMA amplitude and frequency estimates $\{\breve{A}_k^n, \breve{f}_k^n\}$ of the current and previous frames as

$$\hat{A}_k^n(m) = \breve{A}_k^{n-1} + \frac{\breve{A}_k^n - \breve{A}_k^{n-1}}{\mathcal{H}} m, \tag{A.10}$$

and

$$\hat{\phi}_k^n(m) = \hat{\phi}_k^{n-1}(0) + \frac{2\pi m}{f_s}(\breve{f}_k^{n-1} + \frac{\breve{f}_k^n - \breve{f}_k^{n-1}}{2\mathcal{H}} m) \tag{A.11}$$

where f_s is the sampling rate. The signal is then synthesized by overlap-add of all frame estimates as

$$\hat{x}(m) = \sum_{n=1}^{N} \sum_{k=1}^{K} \hat{x}_k^n(m - n\mathcal{H}). \tag{A.12}$$

It is worth noting that a more involved interpolation of the per-sample phases $\hat{\phi}_k^n(m)$ may be better suited to resynthesis of signals with amplitude and frequency modulations, as in (McAulay and Quatieri 1986), where an order-3 polynomial frequency interpolation

scheme was used. We use linear frequency interpolation in the implementation, however, since we defer to `sms-tools` for analysis and synthesis by additive sinusoidal model, which interpolates the frequency linearly.

A.3 Synthetic signals

For the experiments described in Chapters 4 and 5, we generate mixtures of synthetic signals, where each source in the mixture is a simple harmonic waveform subject to a coherent sinusoidal frequency modulation. The mixture is defined as

$$x[m] = \sum_{r=1}^{R} x_r[m] \tag{A.13}$$

where $x_r[m]$ is the r-th source in the mixture. It is expressed as

$$x_r[m] = \sum_{p=1}^{P} A_p[m] \cos(\phi_p[m]) \tag{A.14}$$

where $A_p[m]$ and $\phi_p[m]$ are respective instantaneous amplitude and instantaneous phase (in radians) of partial p, and f_s is the sampling rate. The instantaneous phase is specified by initial phase $\phi_p[0]$ and subsequently calculated as an integral of the instantaneous frequency as

$$\phi_p[m+1] = \phi_p[m] + \frac{2\pi}{f_s} \bar{f}_p (1 + \beta_r[m]) \tag{A.15}$$

where $\bar{f}_p$ is the steady-state frequency (in Hz) of partial p and f_s is the sampling rate. $\beta_r[m]$ is the sinusoidal coherent modulation for source r, expressed as

$$\beta_r[m] = \frac{\Delta_r}{2\pi} \cos(\varphi_r[m]) \tag{A.16}$$

where Δ_r and $\varphi_r[m]$ are the vibrato depth (in Hz) and instantaneous vibrato phase (in radians) of source r. The instantaneous vibrato phase is specified by initial phase $\varphi_r[0]$ and subsequently calculated as the integration of the instantaneous vibrato frequency as

$$\varphi_r[m+1] = \varphi_r[m] + \frac{2\pi}{f_s} \rho_r \tag{A.17}$$

where ρ_r is the vibrato rate (in Hz/sec) of source r.

In the aforementioned experiments, parts $x_r[m]$ in the mixture are generated as simple harmonic waveforms (square, triangle, or sawtooth) with constant amplitude, i.e., $f_p[m] = \bar{f}_p$. The generating amplitudes and frequencies $\{\bar{A}_p, \bar{f}_p\}$ for square wave, triangle wave, and sawtooth wave are expressed as follows:

A.3.1 Square Wave

$$\bar{A}_p = \frac{2}{\pi(2p-1)} \tag{A.18}$$

$$\bar{f}_p = \frac{(2p-1)\bar{f}_1}{f_s} \tag{A.19}$$

where $\bar{f}_1$ is the fundamental frequency (in Hz).

A.3.2 Sawtooth Wave

$$\bar{A}_p = \frac{1}{\pi p} \tag{A.20}$$

$$\bar{f}_p = \frac{p\bar{f}_1}{f_s} \tag{A.21}$$

A.3.3 Triangle Wave

$$\bar{A}_p = \frac{4}{\pi^2(2p-1)^2} \tag{A.22}$$

$$\bar{f}_p = \frac{(2p-1)\bar{f}_1}{f_s} \tag{A.23}$$

Appendix B

PG-CFM Optimization details

This appendix provides details for the inner loop variable updates of PG-CFM and NMF-CFM, which were presented in algorithms 3.1 and 6.1, respectively. In both PG-CFM and NMF-CFM, we show an equivalence of the updates to $p^r(f,n)$ and $v^r(n)$ to optimization problems with a standard form, which permits the use of existing efficient solvers in the inner loop. In both NMF-CFM, we derive the multiplicative updates to $w^r(f)$ and $h^r(n)$, which amounts to additive gradient descent with a fixed step size that assures satisfaction of the nonnegativity constraints along with a non-increasing cost between iterations.

B.1 Quadratic program

A Quadratic Program (QP) is a constrained optimization problem of the form

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}x^T P x + q^T x + r \\ \text{subject to} \quad & Gx \preceq h \\ & Ax = b \end{aligned} \tag{B.1}$$

where $P \in \mathbb{R}^n$ is a symmetric matrix, $G \in \mathbb{R}^{m\times n}$, $A \in \mathbb{R}^{p\times n}$, $h \in \mathbb{R}^m$, $b \in \mathbb{R}^m$, and $x \in \mathbb{R}^n$ (Boyd and Vandenberghe 2009). When $P \in \mathbb{S}^n_+$, i.e., is both symmetric positive semidefinite, then the QP is convex and permits a globally optimal solution that can be found efficiently using existing solvers.

B.2 PG-CFM updates

The PG-CFM algorithm finds a locally optimal solution to the optimization problem expressed by equation 3.35 via iterative alternating minimization of the variables p and υ. We show that each of these optimization steps is equivalent to a convex QP for which an optimal solution with respect to the independent variable can be found efficiently by using the `quadprog` function from the Matlab optimization toolbox. In fact, in update of υ is equivalent to an unconstrained convex QP, which permits a closed form solution.

B.2.1 Optimization problem for the update of $p^r(k,n)$

We here discuss the inner loop update to $p^r(k,n)$, which is accomplished by minimizing the cost function formalized in equation 3.34 with $\upsilon^r(n)$ held fixed to its previously assigned value. With fixed $\upsilon^r(n)$, the optimization problem function expressed by equation 3.35 simplifies and can be expressed as

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{r=1}^{R} p^r(k,n)|\Upsilon(k,n) - \upsilon^r(n)|^2 \\ & + \lambda_{\Delta p}\sum_{n=2}^{N}\sum_{r=1}^{R}(p^r(k,n) - p^r(k,n-1))^2 \\ \text{subject to} \quad & p^r(k,n) \geq 0 \,\forall\, r,k,n, \\ & \sum_{r=1}^{R} p^r(k,n) = 1 \,\forall\, \{k,n\}. \end{aligned} \tag{B.2}$$

We now show that this optimization problem is equivalent to a QP of the form expressed by equation B.1.

B.2.2 Column vector definitions

Let $\tilde{\mathbf{p}} \in \mathbb{R}^{RKN}$ be the "unfolding" of the tensor $p^r(k,n) \in \mathbb{R}^{R\times K\times N}$ into a column vector, i.e.,

$$\tilde{\mathbf{p}} \triangleq \begin{pmatrix} p^1(1,1) \\ p^2(1,1) \\ \vdots \\ p^R(1,1) \\ p^1(2,1) \\ p^2(2,1) \\ \vdots \\ p^R(2,1) \\ \vdots \\ p^R(K,1) \\ p^1(1,2) \\ \vdots \\ p^R(1,2) \\ \vdots \\ p^R(K,2) \\ \vdots \\ p^R(K,N) \end{pmatrix}. \tag{B.3}$$

The i-th entry of $\tilde{\mathbf{p}}$ can alternately be determined by

$$\tilde{\mathbf{p}}_i = p^{\tilde{r}(i)}(\tilde{k}(i), \tilde{n}(i)) \tag{B.4}$$

where $\tilde{r}(i)$, $\tilde{k}(i)$, and $\tilde{n}(i)$ are defined as

$$\tilde{r}(i) \triangleq ((i-1) \mod R) + 1, \tag{B.5}$$

$$\tilde{k}(i) \triangleq (\lfloor \frac{i-1}{R} \rfloor \mod K) + 1, \tag{B.6}$$

and

$$\tilde{n}(i) \triangleq \lfloor \frac{i-1}{RK} \rfloor + 1, \tag{B.7}$$

respectively. $\mod$ denotes the modulus operator and $\lfloor \cdot \rfloor$ denotes the floor operator.

$\tilde{\mathbf{d}} \in \mathbb{R}^{RKN}$ is similarly defined as the column vector expansion of the tensor $|\Upsilon(k,n) - \upsilon^r(n)|^2 \in \mathbb{R}^{R\times K\times N}$, i.e.,

$$\tilde{\mathbf{d}} \triangleq \begin{pmatrix} (\Upsilon(1,1)-\upsilon^1(1))^2 \\ (\Upsilon(1,1)-\upsilon^2(1))^2 \\ \vdots \\ (\Upsilon(1,1)-\upsilon^R(1))^2 \\ (\Upsilon(2,1)-\upsilon^1(1))^2 \\ (\Upsilon(2,1)-\upsilon^2(1))^2 \\ \vdots \\ (\Upsilon(2,1)-\upsilon^R(1))^2 \\ \vdots \\ (\Upsilon(K,1)-\upsilon^R(1))^2 \\ (\Upsilon(1,2)-\upsilon^1(2))^2 \\ \vdots \\ (\Upsilon(1,2)-\upsilon^R(2))^2 \\ \vdots \\ (\Upsilon(K,2)-\upsilon^R(2))^2 \\ \vdots \\ (\Upsilon(K,N)-\upsilon^R(N))^2 \end{pmatrix}. \tag{B.8}$$

As with previous column vector definition, $\tilde{\mathbf{d}}$ can be computed element-wise as

$$\tilde{\mathbf{d}}_i = \left(\Upsilon(\tilde{k}(i),\tilde{n}(i)) - \upsilon^{\tilde{r}(i)}(\tilde{n}(i))\right)^2. \tag{B.9}$$

The column vector notation permits the expression of element-wise tensor sums as vector dot products, e.g.,

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{r=1}^{R} p^r(k,n)|\Upsilon(k,n) - \upsilon^r(n)|^2 = \tilde{\mathbf{d}}^T\tilde{\mathbf{p}}. \tag{B.10}$$

B.2.3 Smoothing matrix

We define a smoothing matrix which computes the first order difference of $\tilde{\mathbf{p}}$, as in the second term of B.2. The smoothing matrix $\mathbf{\Lambda}_p \in \mathbb{R}^{RKN \times RKN}$ is defined as

$$\mathbf{\Lambda}_p = \begin{pmatrix} \mathbf{0}_{RK,1} & \mathbf{0}_{RK,RK-1} & \mathbf{0}_{RK,1} & \mathbf{0}_{RK,1} & \cdots & \mathbf{0}_{RK,1} & \mathbf{0}_{RK,1} \\ -1 & \mathbf{0}_{1,RK-1} & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & \mathbf{0}_{1,RK-1} & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & \mathbf{0}_{1,RK-1} & 1 & 0 \\ 0 & \cdots & 0 & 0 & -1 & \mathbf{0}_{1,RK-1} & 1 \end{pmatrix} \tag{B.11}$$

where $\mathbf{0}_{a,b} \in \mathbb{R}^{a \times b}$ denotes a rectangular zeros matrix. Starting at row $RK+1$, the matrix corresponds to an first-order FIR smoothing filter matrix acting the n dimension of $p^r(k,n)$. In other words, $p^r(k,n) - p^r(k,n-1)$ can be expressed by the reshaping of $\mathbf{\Lambda}_p \tilde{\mathbf{p}}$ like in equation B.3. Setting the first RK rows to zero effectively sets the first filter output to zero, i.e., sets the starting index the sum over n in the second term of equation B.2 to $n = 1$ rather than $n = 0$ (for which $p^r(k, n-1)$ is undefined).

An example of the smoothing matrix $\mathbf{\Lambda}_p$ for $R = 2$, $K = 3$, $N = 4$ is given in figure B.1.

$$\mathbf{\Lambda}_p = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&1
\end{pmatrix}$$

Fig. B.1 Example smoothing matrix $\mathbf{\Lambda}_p$ for $R = 2$, $K = 3$, $N = 4$

The second sum in the cost function of equation B.2 is thus expressed as

$$\lambda_{\Delta p} \sum_{n=2}^{N} \sum_{r=1}^{R} (p^r(k,n) - p^r(k,n-1))^2 = \lambda_{\Delta p} \tilde{\mathbf{p}}^T \mathbf{\Lambda}_p^T \mathbf{\Lambda}_p \tilde{\mathbf{p}}. \tag{B.12}$$

B.2.4 Equality constraints

The sum-to-one constraints on $p^r(k,n)$ can be equivalently expressed in the vector notation via the introduction of a matrix which effectively sums across the r dimension. The matrix $\mathbf{\Sigma}_r \in \mathbb{R}^{KN \times RKN}$ is a block diagonal matrix of row vectors of ones, and is expressed as

$$\mathbf{\Sigma}_r = \begin{pmatrix} \mathbf{1}_{1,R} & \mathbf{0}_{1,R} & \cdots & \mathbf{0}_{1,R} & \mathbf{0}_{1,R} \\ \mathbf{0}_{1,R} & \mathbf{1}_{1,R} & \mathbf{0}_{1,R} & \cdots & \mathbf{0}_{1,R} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0}_{1,R} & & & \ddots & \mathbf{0}_{1,R} \\ \mathbf{0}_{1,R} & & \cdots & \mathbf{0}_{1,R} & \mathbf{1}_{1,R} \end{pmatrix} \tag{B.13}$$

where $\mathbf{1}_{a,b} \in \mathbb{R}^{a \times b}$ is a rectangular ones matrix (a row vector, in this case). An example of the summing matrix $\mathbf{\Sigma}_r$ for $R = 2$, $K = 3$, $N = 4$ is given in figure B.2.

$$\mathbf{\Sigma}_r = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Fig. B.2 Example summing matrix $\mathbf{\Sigma}_r$ for $R = 2$, $K = 3$, $N = 4$

The second constraint in equation B.2, which requires $p^r(k,n)$ to sum to one across all parts, can be equivalently expressed as

$$\mathbf{\Sigma}_r \tilde{\mathbf{p}} = \mathbf{1}_{KN,1} \tag{B.14}$$

where $\mathbf{1}_{KN,1}$ is a column vector containing KN ones.

B.2.5 Inequality constraints

The nonnegativity constraints on $p^r(k,n)$ can be expressed in the vector notation simply by a multiplication by the identity matrix and element-wise comparison to a column vector of zeros. This is expressed as

$$-\mathbf{I}_{RKN}\tilde{\mathbf{p}} \preceq \mathbf{0}_{RKN,1} \tag{B.15}$$

where $\mathbf{I}_{RKN} \in \mathbb{R}^{RKN \times RKN}$ is the identity matrix and $\mathbf{0}_{RKN,1} \in \mathbb{R}^{RKN}$ is a column vector of zeros. Note that we favor the element-wise inequality comparison using the $\preceq$ operator, which reflects QP formulation given in equation B.1.

B.2.6 Quadratic program equivalence

The optimization problem expressed by equation B.2 is equivalent to the QP expressed by

$$\begin{aligned} \underset{\tilde{\mathbf{p}}}{\text{minimize}} \quad & \lambda_{\Delta p}\tilde{\mathbf{p}}^T\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p\tilde{\mathbf{p}} + \tilde{\mathbf{d}}^T\tilde{\mathbf{p}} \\ \text{subject to} \quad & -\mathbf{I}_{RKN}\tilde{\mathbf{p}} \preceq \mathbf{0}_{RKN,1} \\ & \mathbf{\Sigma}_r\tilde{\mathbf{p}} = \mathbf{1}_{KN,1}. \end{aligned} \tag{B.16}$$

B.2.7 Quadratic program convexity

The above QP is convex if and only if $\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p$ is a positive semidefinite and symmetric matrix. Matrix symmetry requires $A^T = A$, which is satisfied trivially for $\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p$. Matrix positive semidefiniteness requires $x^TAx \geq 0 \,\forall\, x$, which is satisfied by construction in the case of $\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p$, since $\tilde{\mathbf{p}}^T\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p\tilde{\mathbf{p}}$ is equivalent to $\sum_{n=2}^{N}\sum_{r=1}^{R}(p^r(k,n) - p^r(k,n-1))^2$, which is nonnegative. The QP expressed by equation B.16 is thus a convex optimization problem, for which a globally optimal $\tilde{\mathbf{p}}^*$ can be found efficiently.

B.2.8 Implementation

The `quadprog` function from Matlab's optimization toolbox is used to determine the solution $\tilde{\mathbf{p}}^*$ to the convex QP expressed by equation B.16. The `interior-point-convex` algorithm is chosen as the solver, which implements an iterative interior point Newton's step method detailed in (Mehrotra 1992). Sparse matrix structures are used where applicable in order to reduce the memory requirements of the algorithm.

B.2.9 Optimization problem for the update of $\upsilon^r(n)$

$\upsilon^r(n)$ is updated in the inner loop by minimizing the cost function formalized in equation 3.34 with $p^r(k,n)$ held fixed to its previously assigned value. When $p^r(k,n)$ is fixed, the optimization problem function expressed by equation 3.35 simplifies and can be expressed as

$$\begin{aligned} \underset{\upsilon}{\text{minimize}} \quad & \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{r=1}^{R} p^r(k,n)|\Upsilon(k,n) - \upsilon^r(n)|^2 \\ & + \lambda_{\Delta\upsilon}\sum_{n=2}^{N}\sum_{r=1}^{R}(\upsilon^r(n) - \upsilon^r(n-1))^2. \end{aligned} \tag{B.17}$$

Note that this optimization problem is unconstrained, since $p^r(f,n)$ is fixed to its previous values and thus the constraints given in equation 3.35 will remained satisfied following the update to $\upsilon^r(n)$. We show that this optimization problem is equivalent to an unconstrained convex QP that permits a closed-form solution.

B.2.10 Column vector definitions

Let $\tilde{\boldsymbol{\upsilon}} \in \mathbb{R}^{RN}$ be the "unfolding" of the tensor $\upsilon^r(n) \in \mathbb{R}^{R\times N}$ into a column vector, i.e.,

$$\tilde{\boldsymbol{\upsilon}} \triangleq \begin{pmatrix} \upsilon^1(1) \\ \upsilon^2(1) \\ \vdots \\ \upsilon^R(1) \\ \upsilon^1(2) \\ \upsilon^2(2) \\ \vdots \\ \upsilon^R(2) \\ \vdots \\ \upsilon^R(N) \end{pmatrix}. \tag{B.18}$$

The i-th entry of $\tilde{\boldsymbol{\upsilon}}$ can alternately be determined by

$$\tilde{\boldsymbol{\upsilon}}_i = \upsilon^{\tilde{r}(i)}(\tilde{n}(i)) \tag{B.19}$$

where $\tilde{r}(i)$ and $\tilde{n}(i)$ are the same definitions as B.5 and B.7, except with $K = 1$ to eliminate the k dimension.

Let $\tilde{\boldsymbol{\Upsilon}} \in \mathbb{R}^{KN}$ be the "unfolding" of the tensor $\Upsilon(k, n) \in \mathbb{R}^{K\times N}$ into a column vector, i.e.,

$$\tilde{\boldsymbol{\Upsilon}} \triangleq \begin{pmatrix} \Upsilon(1,1) \\ \Upsilon(2,1) \\ \vdots \\ \Upsilon(K,1) \\ \Upsilon(1,2) \\ \Upsilon(2,2) \\ \vdots \\ \Upsilon(K,2) \\ \vdots \\ \Upsilon(K,N) \end{pmatrix}. \tag{B.20}$$

The i-th entry of $\tilde{\boldsymbol{\Upsilon}}$ can alternately be determined by

$$\tilde{\boldsymbol{\Upsilon}}_i = \Upsilon(\tilde{k}(i), \tilde{n}(i)) \tag{B.21}$$

where $\tilde{k}(i)$ and $\tilde{n}(i)$ are the same definitions as B.6 and B.7, except with $R = 1$ to eliminate the r dimension.

B.2.11 Repeating matrix

$\mathbf{U}_r \in \mathbb{R}^{RKN \times KN}$ repeats $\tilde{\boldsymbol{\Upsilon}}$ along the r dimension. It is a block diagonal matrix of column vectors of ones, and is expressed as

$$\mathbf{U}_r = \begin{pmatrix} \mathbf{1}_{R,1} & \mathbf{0}_{R,1} & \cdots & \mathbf{0}_{R,1} & \mathbf{0}_{R,1} \\ \mathbf{0}_{R,1} & \mathbf{1}_{R,1} & \mathbf{0}_{R,1} & \cdots & \mathbf{0}_{R,1} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0}_{R,1} & & & \ddots & \mathbf{0}_{R,1} \\ \mathbf{0}_{R,1} & & \cdots & \mathbf{0}_{R,1} & \mathbf{1}_{R,1} \end{pmatrix} \tag{B.22}$$

The application of the repeating matrix is then expressed as

$$\mathbf{U}_r \tilde{\boldsymbol{\Upsilon}} = \begin{pmatrix} \bar{\Upsilon}(1,1) \\ \bar{\Upsilon}(2,1) \\ \vdots \\ \bar{\Upsilon}(K,1) \\ \bar{\Upsilon}(1,2) \\ \bar{\Upsilon}(2,2) \\ \vdots \\ \bar{\Upsilon}(K,2) \\ \vdots \\ \bar{\Upsilon}(K,N) \end{pmatrix} \tag{B.23}$$

where $\bar{\Upsilon}(k,n) \in \mathbb{R}^R$ is a column vector containing R repetitions of $\Upsilon(k,n)$, i.e.,

$$\bar{\Upsilon}(k,n) = \begin{pmatrix} \Upsilon(k,n) \\ \Upsilon(k,n) \\ \vdots \\ \Upsilon(k,n) \end{pmatrix}. \tag{B.24}$$

An example of the expanding matrix $\mathbf{U}_r$ for $R = 2$, $K = 3$, $N = 4$ is given in figure B.3.

$$\mathbf{U}_r = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Fig. B.3 Example repeating matrix $\mathbf{U}_r$ for $R = 2$, $K = 3$, $N = 4$

$\mathbf{U}_k \in \mathbb{R}^{RKN \times RN}$ is constructed similarly to expand $\tilde{\boldsymbol{v}}$ along the k dimension. It is a block diagonal matrix of stacked identity matrices, and resembles $\mathbf{U}_r$, but requires the

permutation of rows since repeats occur along an inner dimension. This repeating matrix is expressed as

$$\mathbf{U}_k = \begin{pmatrix} \mho(R,K) & \mathbf{0}_{RK,R} & \cdots & \mathbf{0}_{RK,R} & \mathbf{0}_{RK,R} \\ \mathbf{0}_{RK,R} & \mho(R,K) & \mathbf{0}_{RK,R} & \cdots & \mathbf{0}_{RK,R} \\ \vdots & & \ddots & & \vdots \\ \mathbf{0}_{RK,R} & & & \ddots & \mathbf{0}_{RK,R} \\ \mathbf{0}_{RK,R} & \mathbf{0}_{RK,R} & \cdots & \mathbf{0}_{RK,R} & \mho(R,K) \end{pmatrix} \tag{B.25}$$

where $\mho(R,K) \in \mathbb{R}^{RK,R}$ is the vertical concatenation of K identity matrixes of size R, i.e.,

$$\mho(R,K) = \begin{pmatrix} \mathbf{I}_R \\ \vdots \\ \mathbf{I}_R \end{pmatrix}. \tag{B.26}$$

An example of the expanding matrix $\mathbf{U}_k$ for $R = 2$, $K = 3$, $N = 4$ is given in figure B.4.

B.2.12 Smoothing matrix

We formulate $\mathbf{\Lambda}_\upsilon \in \mathbb{R}^{RN \times RN}$, a smoothing matrix for $\tilde{\boldsymbol{\upsilon}}$, which takes the same form as B.11 with $K = 1$, and is expressed as

$$\mathbf{\Lambda}_\upsilon = \begin{pmatrix} \mathbf{0}_{R,1} & \mathbf{0}_{R,R-1} & \mathbf{0}_{R,1} & \mathbf{0}_{R,1} & \cdots & \mathbf{0}_{R,1} & \mathbf{0}_{R,1} \\ -1 & \mathbf{0}_{1,R-1} & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & \mathbf{0}_{1,R-1} & 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & \mathbf{0}_{1,R-1} & 1 & 0 \\ 0 & \cdots & 0 & 0 & -1 & \mathbf{0}_{1,R-1} & 1 \end{pmatrix} \tag{B.27}$$

B.2.13 Weighting matrix

We formulate a weighting matrix for $\mathbf{D}_p \in \mathbb{R}^{RKN \times RKN}$, a diagonal matrix whose diagonal elements are $p^r(k,n)$, i.e.,

$$[\mathbf{D}_p]_{i,j} = \tilde{\mathbf{p}}_i \delta(i-j) \tag{B.28}$$

$$\mathbf{U}_k = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

Fig. B.4 Example repeating matrix $\mathbf{U}_k$ for $R = 2$, $K = 3$, $N = 4$

This permits an expression of the argument to the first summation in equation B.17 as

$$p^{\tilde{r}(i)}(\tilde{k}(i),\tilde{n}(i))|\Upsilon(\tilde{k}(i),\tilde{n}(i)) - \upsilon^{\tilde{r}(i)}(\tilde{n}(i))|^2 = [(\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}} - \mathbf{U}_k\tilde{\boldsymbol{\upsilon}})^T\mathbf{D}_p(\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}} - \mathbf{U}_k\tilde{\boldsymbol{\upsilon}})]_i \quad \text{(B.29)}$$

where $\tilde{r}(i)$, $\tilde{k}(i)$, and $\tilde{n}(i)$ are the indexing functions defined by equations B.5, B.6, and B.7, respectively.

B.2.14 Quadratic program equivalence

The optimization problem expressed by equation B.17 is equivalent to the unconstrained minimization problem defined as

$$\underset{\tilde{\boldsymbol{\upsilon}}}{\text{minimize}} \quad (\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}} - \mathbf{U}_k\tilde{\boldsymbol{\upsilon}})^T\mathbf{D}_p(\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}} - \mathbf{U}_k\tilde{\boldsymbol{\upsilon}}) + \lambda_{\Delta\upsilon}\tilde{\boldsymbol{\upsilon}}^T\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon\tilde{\boldsymbol{\upsilon}} \quad \text{(B.30)}$$

This is quadratic in $\tilde{\boldsymbol{\upsilon}}$ and can thus be expanded and regrouped as the unconstrained QP

$$\underset{\tilde{\boldsymbol{\upsilon}}}{\text{minimize}} \quad \tilde{\boldsymbol{\upsilon}}^T(\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_k + \lambda_{\Delta\upsilon}\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon)\tilde{\boldsymbol{\upsilon}} - 2\tilde{\boldsymbol{\Upsilon}}^T\mathbf{U}_r^T\mathbf{D}_p\mathbf{U}_k\tilde{\boldsymbol{\upsilon}} + \tilde{\boldsymbol{\Upsilon}}^T\mathbf{U}_r^T\mathbf{D}_p\mathbf{U}_k\tilde{\boldsymbol{\Upsilon}}. \quad \text{(B.31)}$$

Note that the third term is a constant and thus has no effect on the solution.

B.2.15 Closed-form solution

The optimization problem expressed by equation B.31 is an unconstrained QP, which is convex since the quadratic term $(\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_k + \lambda_{\Delta\upsilon}\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon)$ is positive semidefinite and symmetric by the same arguments given in section B.2.7. In fact we can make the stronger statement that the quadratic term is positive definite since $\mathbf{U}_k$ is full rank. The lack of constraints and positive definiteness of the quadratic term permit a closed-form solution to the optimization problem, which is expressed as

$$\tilde{\boldsymbol{\upsilon}}^* = (\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_k + \lambda_{\Delta\upsilon}\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon)^{-1}(\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}}) \quad \text{(B.32)}$$

Appendix C

NMF-CFM Optimization details

C.1 NMF-CFM updates

The NMF-CFM algorithm finds a locally optimal solution to the optimization problem expressed by equation 6.11 via iterative alternating minimization of the variables w, h, p and v. The multiplicative updates for w and h are derived here. As with the PG-CFM, the updates to p and v come from the solution to equivalent convex QPs, which are formulated here.

C.1.1 Multiplicative updates of $w^r(f)$ and $h^r(n)$

We here derive the inner loop updates to the nonnegative factors $w^r(f)$ and $h^r(n)$ according to the standard form for NMF multiplicative updates, which was described in section 2.7.3. We express the partial derivative of the cost function with respect to each of the two variables $\theta \in \{w, h\}$, i.e., $\frac{\partial J}{\partial \theta}$. We then reformulate the partial derivative as the difference of two nonnegative terms, and finally give the multiplier used in the update as the ratio of these terms.

C.1.2 Update of $w^r(f)$

With h, p, and v fixed, the NMF-CFM optimization problem, which is given by equation 6.11, simplifies, and can be expressed as

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & D(\mathbf{X}|\hat{\mathbf{X}}) + \lambda_{\Delta w} C_{\Delta w}(\mathbf{W}) \\ \text{subject to} \quad & w^r(f) \geq 0 \,\forall\, \{r, f\}. \end{aligned} \tag{C.1}$$

We can analyze the partial derivative with respect to each term individually by linearity of the partial derivative, i.e.,

$$\frac{\partial J}{\partial w^r(f)} = \frac{\partial D(\mathbf{X}|\hat{\mathbf{X}})}{\partial w^r(f)} + \lambda_{\Delta w} \frac{\partial C_{\Delta w}}{\partial w^r(f)} \tag{C.2}$$

The first term in equation C.2 is expressed as

$$\begin{aligned} \frac{\partial D(\mathbf{X}|\hat{\mathbf{X}})}{\partial w^r(f)} \quad &= 2 \sum_{n=0}^{N-1} \hat{X}(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))) \\ &\quad - 2 \sum_{n=0}^{N-1} X(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))) \end{aligned} \tag{C.3}$$

where $\hat{X}(f,n)$ is the model approximation, defined as

$$\hat{X}(f,n) = \sum_{r=0}^{R-1} w^r(f) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) \tag{C.4}$$

The second term in equation C.2 is expressed as

$$\lambda_{\Delta w} \frac{\partial C_{\Delta w}}{\partial w^r(f)} = \lambda_{\Delta w} (2\varrho(w^r(f)) - 2\varkappa(w^r(f))) \tag{C.5}$$

where

$$\varrho(w^r(f)) = \begin{cases} w^r(f) & f = 0 \\ 2w^r(f) & 0 < f < F-1 \\ w^r(f) & f = F-1 \end{cases} \tag{C.6}$$

and

$$\varkappa(w^r(f)) = \begin{cases} w^r(f+1) & f = 0 \\ w^r(f-1) + 2w^r(f+1) & 0 < f < F-1 \\ w^r(f-1) & f = F-1. \end{cases} \tag{C.7}$$

We can thus reformulate the expression for the partial derivative as the difference of two nonnegative terms, as

$$\frac{\partial J}{\partial w^r(f)} = G_{w^r(f)} - F_{w^r(f)}, \tag{C.8}$$

which permits multiplicative updates of the form

$$w^r(f) \leftarrow w^r(f) \frac{F_{w^r(f)}}{G_{w^r(f)}}. \tag{C.9}$$

The denominator of the multiplicative update is expressed as

$$G_{w^r(f)} = 2 \sum_{n=0}^{N-1} \hat{X}(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) + 2\lambda_{\Delta w} \varrho(w^r(f)), \tag{C.10}$$

while the numerator is expressed as

$$F_{w^r(f)} = 2 \sum_{n=0}^{N-1} X(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) + 2\lambda_{\Delta w} \varkappa(w^r(f)). \tag{C.11}$$

Both $F_{w^r(f)}$ and $G_{w^r(f)}$ are nonnegative functions since they sum over the nonnegative variables w, h, and p, and the weights $H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))$ represent Fourier transform magnitudes and are thus nonnegative. The full multiplicative update is thus expressed as

$$w^r(f) \leftarrow w^r(f) \frac{\sum_{n=0}^{N-1} X(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) + \lambda_{\Delta w} \varkappa(w^r(f))}{\sum_{n=0}^{N-1} \hat{X}(f,n) h^r(n) \sum_{k=0}^{K-1} p^r(k,n) H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) + \lambda_{\Delta w} \varrho(w^r(f))}. \tag{C.12}$$

C.1.3 Multiplicative update of $h^r(n)$

With w, p, and υ fixed, the NMF-CFM optimization problem, which is given by equation 6.11, simplifies, and can be expressed as

$$\underset{w,h,p,\upsilon}{\text{minimize}} \quad D(\mathbf{X}|\hat{\mathbf{X}}) \quad \text{subject to} \quad h^r(n) \geq 0 \, \forall \, \{r,n\}. \tag{C.13}$$

The partial derivative of the function-to-be-minimized can be expressed as

$$\begin{aligned}\frac{\partial D(\mathbf{X}|\hat{\mathbf{X}})}{\partial h^r(n)} = \; & 2\sum_{f=0}^{F-1} X(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)) \\ & - 2\sum_{f=0}^{F-1} \hat{X}(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)).\end{aligned} \tag{C.14}$$

We can thus reformulate the expression for the partial derivative as the difference of two nonnegative terms, as

$$\frac{\partial J}{\partial h^r(n)} = G_{h^r(n)} - F_{h^r(n)}, \tag{C.15}$$

which permits multiplicative updates of the form

$$h^r(n) \leftarrow h^r(n)\frac{F_{h^r(n)}}{G_{h^r(n)}}. \tag{C.16}$$

The denominator of the multiplicative update is expressed as

$$G_{h^r(n)} = \sum_{f=0}^{F-1} \hat{X}(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)), \tag{C.17}$$

while the numerator of the multiplicative update is expressed as

$$F_{h^r(n)} = \sum_{f=0}^{F-1} X(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n)). \tag{C.18}$$

Nonnegativity of $F_{h^r(n)}$ and $G_{h^r(n)}$ is assured as they are sums of nonnegative terms with nonnegative weights. The full multiplicative update of the temporal activations is thus expressed as

$$h^r(n) \leftarrow h^r(n)\frac{\sum_{f=0}^{F-1} X(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))}{\sum_{f=0}^{F-1} \hat{X}(f,n)w^r(f)\sum_{k=0}^{K-1} p^r(k,n)H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))}. \tag{C.19}$$

C.1.4 Update of $p^r(k,n)$

The inner loop update of p is discussed here. With w, h, and υ fixed, the NMF-CFM optimization problem given by 6.11, simplifies, and can be expressed as

$$\begin{aligned} \underset{p}{\text{minimize}} \quad & D(\mathbf{X}|\hat{\mathbf{X}}) + \lambda_{CFM} C_{CFM}(\boldsymbol{p}, \boldsymbol{\upsilon}) + \lambda_{\Delta p} C_{\Delta p}(\boldsymbol{p}) \\ \text{subject to} \quad & p^r(k,n) \geq 0 \, \forall \, r,k,n, \\ & \sum_{r}^{R} p^r(k,n) = 1 \, \forall \, \{k,n\}. \end{aligned} \tag{C.20}$$

As with PG-CFM, we show an equivalence of this optimization problem to a convex QP by using a similar vector notation, including the introduction of matrices that sum, repeat, smooth, and weight the vectors as in the previous sections. Since these matrices were explicitly defined before, and in the sake of brevity, we merely list the vectors and matrices used, along with their domain and functions, as follows:

- $\tilde{\mathbf{p}} \in \mathbb{R}^{RKN}$ is the column vector unfolding of $p^r(k,n)$
- $\tilde{\mathbf{d}} \in \mathbb{R}^{RKN}$ is the column vector unfolding of $|\Upsilon(k,n) - \upsilon^r(n)|^2$
- $\tilde{\boldsymbol{\Xi}} \in \mathbb{R}^{RFN}$ is the column vector unfolding of $\sum_{r=0}^{R-1} w^r(f) h^r(n)$
- $\mathbf{D}_{\Xi} \in \mathbb{R}^{RFN \times RFN}$ is the diagonal matrix whose diagonal entries are $\tilde{\boldsymbol{\Xi}}$
- $\tilde{\mathbf{H}} \in \mathbb{R}^{RFNK}$ is the column vector unfolding and repeating of $H(\frac{f}{\mathcal{N}} - \breve{f}(k,n))$
- $\mathbf{D}_H \in \mathbb{R}^{RFNK \times RFNK}$ is the diagonal matrix whose diagonal entries are $\tilde{\mathbf{H}}$
- $\tilde{\mathbf{X}} \in \mathbb{R}^{FN}$ is the column vector unfolding of $\mathbf{X}$
- $\mathbf{U}_f \in \mathbb{R}^{RFNK \times RFN}$ is the repeating matrix along the f dimension
- $\boldsymbol{\Sigma}_k \in \mathbb{R}^{RFN \times RFNK}$ is the summing matrix along the k dimension
- $\boldsymbol{\Sigma}_r \in \mathbb{R}^{FN \times RFNK}$ is the summing matrix along the r dimension
- $\boldsymbol{\Lambda}_p \in \mathbb{R}^{RKN \times RKN}$ is the smoothing matrix acting on p in the n dimension

Each of the three terms contributing to the function-to-be-minimized in equation C.20 can thus be expressed in the vector notation. In particular, the NMF-CFM model fit is expressed as

$$D(\mathbf{X}|\hat{\mathbf{X}}) = \sum_{f}\sum_{N} d(\mathbf{X}_{f,n}|\hat{\mathbf{X}}_{f,n}) = (\tilde{\mathbf{X}} - \mathbf{\Sigma}_r\mathbf{D}_\Xi\mathbf{\Sigma}_k\mathbf{D}_H\mathbf{U}_f\tilde{\mathbf{p}})^T(\tilde{\mathbf{X}} - \mathbf{\Sigma}_r\mathbf{D}_\Xi\mathbf{\Sigma}_k\mathbf{D}_H\mathbf{U}_f\tilde{\mathbf{p}}). \tag{C.21}$$

The CFM penalty is expressed as

$$C_{CFM}(\boldsymbol{p}) = \tilde{\mathbf{d}}^T\tilde{\mathbf{p}}. \tag{C.22}$$

The p smoothness penalty is expressed as

$$C_{\Delta p}(\boldsymbol{p}) = \tilde{\mathbf{p}}^T\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p\tilde{\mathbf{p}}. \tag{C.23}$$

As in sections B.2.4 and B.2.5, the equality constraints in equation C.20 can be expressed as

$$\mathbf{\Sigma}_r\tilde{\mathbf{p}} = \mathbf{1}_{KN,1}, \tag{C.24}$$

while the inequality constraints can be expressed as

$$-\mathbf{I}_{RKN}\tilde{\mathbf{p}} \preceq \mathbf{0}_{RKN,1} \tag{C.25}$$

Equality and inequality constraints on $\tilde{\mathbf{p}}$ are expressed as in sections B.2.4 and B.2.5 on page 108.

The optimization problem expressed by equation C.20 is thus expressed in the vector notation as

$$\begin{aligned} \underset{\tilde{\mathbf{p}}}{\text{minimize}} \quad & (\tilde{\mathbf{X}} - \mathbf{\Sigma}_r\mathbf{D}_\Xi\mathbf{\Sigma}_k\mathbf{D}_H\mathbf{U}_f\tilde{\mathbf{p}})^T(\tilde{\mathbf{X}} - \mathbf{\Sigma}_r\mathbf{D}_\Xi\mathbf{\Sigma}_k\mathbf{D}_H\mathbf{U}_f\tilde{\mathbf{p}}) + \lambda_{CFM}\tilde{\mathbf{d}}^T\tilde{\mathbf{p}} + \lambda_{\Delta p}\tilde{\mathbf{p}}^T\mathbf{\Lambda}_p^T\mathbf{\Lambda}_p\tilde{\mathbf{p}} \\ \text{subject to} \quad & -\mathbf{I}_{RKN}\tilde{\mathbf{p}} \preceq \mathbf{0}_{RKN} \\ & \mathbf{\Sigma}_r\tilde{\mathbf{p}} = \mathbf{1}_{KN,1}. \end{aligned} \tag{C.26}$$

The cost function to be minimized is quadratic in the independent variable, and can thus be rewritten as a QP in the form given by equation B.1, expressed as

$$\begin{aligned} \underset{\tilde{\mathbf{p}}}{\text{minimize}} \quad & \tilde{\mathbf{p}}^T(\mathbf{U}_f^T\mathbf{D}_H\boldsymbol{\Sigma}_k^T\mathbf{D}_\Xi\boldsymbol{\Sigma}_r^T\boldsymbol{\Sigma}_r\mathbf{D}_\Xi\boldsymbol{\Sigma}_k\mathbf{D}_H\mathbf{U}_f + \lambda_{\Delta p}\boldsymbol{\Lambda}_p^T\boldsymbol{\Lambda}_p)\tilde{\mathbf{p}} \\ & + (\lambda_{CFM}\tilde{\mathbf{d}}^T - 2\tilde{\mathbf{X}}^T\boldsymbol{\Sigma}_r\mathbf{D}_\Xi\boldsymbol{\Sigma}_k\mathbf{D}_H\mathbf{U}_f)\tilde{\mathbf{p}} + \tilde{\mathbf{X}}^T\tilde{\mathbf{X}} \\ \text{subject to} \quad & -\mathbf{I}_{RKN}\tilde{\mathbf{p}} \preceq \mathbf{0}_{RKN} \\ & \boldsymbol{\Sigma}_r\tilde{\mathbf{p}} = \mathbf{1}_{KN,1}. \end{aligned} \tag{C.27}$$

C.1.5 Update of $\upsilon^r(n)$

The only contributions to the NMF-CFM cost function, given by equation 6.10, that depend on $\upsilon^r(n)$ are the penalty terms C_{CFM} and $C_{\Delta\upsilon}$. Therefore, when p, w, and h are fixed, the optimization problem expressed by 6.11 simplifies and can be expressed as

$$\underset{p,\upsilon}{\text{minimize}} \quad \lambda_{CFM}\sum_{r=0}^{R-1}\sum_{k=0}^{K-1}\sum_{n=0}^{N-1}p^r(k,n)(\Upsilon(k,n)-\upsilon^r(n))^2 + \lambda_{\Delta\upsilon}\sum_{r=0}^{R-1}\sum_{n=1}^{N-1}(\upsilon^r(n)-\upsilon^r(n-1))^2. \tag{C.28}$$

This is equivalent to the unconstrained convex QP of the form

$$\underset{\tilde{\boldsymbol{\upsilon}}}{\text{minimize}} \quad \tilde{\boldsymbol{\upsilon}}^T(\lambda_{CFM}\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_k + \lambda_{\Delta\upsilon}\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon)\tilde{\boldsymbol{\upsilon}} - 2\lambda_{CFM}\tilde{\boldsymbol{\Upsilon}}^T\mathbf{U}_r^T\mathbf{D}_p\mathbf{U}_k\tilde{\boldsymbol{\upsilon}} + \lambda_{CFM}\tilde{\boldsymbol{\Upsilon}}^T\mathbf{U}_r^T\mathbf{D}_p\mathbf{U}_k\tilde{\boldsymbol{\Upsilon}}, \tag{C.29}$$

which permits a closed-form solution, expressed as

$$\tilde{\boldsymbol{\upsilon}}^* = (\lambda_{CFM}\mathbf{U}_k^T\mathbf{D}_p\mathbf{U}_k + \lambda_{\Delta\upsilon}\boldsymbol{\Lambda}_\upsilon^T\boldsymbol{\Lambda}_\upsilon)^{-1}(\lambda_{CFM}\mathbf{U}_k\mathbf{D}_p\mathbf{U}_r\tilde{\boldsymbol{\Upsilon}}). \tag{C.30}$$

Bibliography

Achan, K., S. T. Roweis, and B. J. Frey. 2003. Probabilistic inference of speech signals from phaseless spectrograms. In *Neural Information Processing Systems 16*, 1393–1400. MIT Press.

Allen, J., and L. Rabiner. 1977. A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE* 65: 1558–64.

Badeau, R., and M. Plumbley. 2014. Multichannel high-resolution NMF for modeling convolutive mixtures of non-stationary signals in the time-frequency domain. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22: 1670–1680.

Barker, T., and T. Virtanen. 2013. Non-negative tensor factorization of modulation spectrograms for monaural sound separation. In *INTERSPEECH-2013*, Lyon, France, 827–31.

Berry, M. W., M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. 2006. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis* 52 (1): 155–73.

Bertin, N., R. Badeau, and E. Vincent. 2010. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech, and Language Processing* 18: 538–49.

Betser, M. 2009. Sinusoidal polyphonic parameter estimation using the distribution derivative. *IEEE Transactions on Signal Processing* 57: 4633–45.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Boutsidis, C., and E. Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition* 41: 1350–62.

Boyd, S., and L. Vandenberghe. 2009. *Convex Optimization*. Cambridge, UK: Cambridge University Press.

Bregman, A. 1990. *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: The MIT Press.

Bronson, J., and P. Depalle. 2014. Phase constrained complex NMF: separating overlapping partials in mixtures of harmonic musical sounds. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Florence, Italy, 7475–9.

Brown, J. 1991. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America* 89: 425–34.

Bryan, N., G. Mysore, and G. Wang. 2013. Source separation of polyphonic music with interactive user-feedback on a piano-roll display. In *Proceedings of the International Conference on Music Information Retrieval*, Curitiba, Brazil, 119–24.

Choi, S. 2008. Algorithms for orthogonal nonnegative matrix factorization. In *Proceedings of the IEEE International Joint Conference Neural Networks*, Hong Kong, 1828–32.

Chowning, J. M. 1980. Computer synthesis of the singing voice. In *Sound Generation in Winds, Strings, Computers*, 4–13. Stokholm, Sweden: Kungl. Musikaliska Akademien.

Cichocki, A., R. Zdunek, and S. Amari. 2008. Nonnegative matrix and tensor factorization. *IEEE Signal Processing Magazine* 25 (1): 142–5.

Cichocki, A., R. Zdunek, A. H. Phan, and S. Amari. 2009. *Nonnegative Matrix and Tensor Factorizations : Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Chichester, UK: Wiley.

Comon, P. 1994. Independent component analysis, A new concept? *Signal Processing* 36: 287–314.

Cont, A. 2006. Realtime multiple pitch observation using sparse non-negative constraints. In *Proceedings of the International Conference on Music Information Retrieval*, Victoria, Canada, 206–11.

Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data with the EM algorithm. *Journal of the Royal Statistical Society B* 39 (1): 1–38.

Dhillon, I., and S. Sra. 2006. Generalized nonnegative matrix approximations with Bregman divergences. In Y. Weiss, B. Schölkopf, and J. Platt (Eds.), *Advances in Neural Information Processing Systems 18*, 283–290. MIT Press.

Dolson, M. 1986. The phase vocoder: A tutorial. *Computer Music Journal* 10: 14–27.

Donoho, D., and V. Stodden. 2003. When does non-negative matrix factorization give a correct decomposition into parts? In *Proceedings of the Neural Information Processing Systems*, Vancouver, Canada.

Font, F., G. Roma, and X. Serra. 2013. Freesound technical demo. In *Proceedings of the 21st ACM international conference on multimedia*, Barcelona, Spain, 411–2.

Févotte, C. 2011. Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 1980–3.

Févotte, C., N. Bertin, and J. Durrieu. 2009. Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. *Neural Computation* 21 (3): 793–830.

Févotte, C., and J. Idier. 2011. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation* 23 (9): 2421–56.

Gray, R. 1984. Vector quantization. *IEEE Acoustics, Speech, and Signal Processing Magazine* 1: 4–29.

Griffin, D., and J. Lim. 1984. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 32: 236–43.

Harris, F. 1978. On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE* 66: 51–83.

Hennequin, R., R. Badeau, and B. David. 2010. NMF with time-frequency activations to model non stationary audio events. *IEEE Transactions on Audio, Speech, and Language Processing* 19 (4): 744–53.

Hennequin, R., B. David, and R. Badeau. 2011. Beta-divergence as a subclass of Bregman divergence. *IEEE Signal Processing Letters* 18 (2): 83–6.

Hotelling, H. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology* 24: 417–441, 498–520.

Hoyer, P. 2002. Non-negative sparse coding. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, Martigny, Switzerland, 557–65.

Hoyer, P. 2004. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research* 5: 1457–69.

Hsu, C.-L., and J.-S. R. Jang. 2009. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech and Language Processing* 18 (2): 310–9.

Hunter, D., and K. Lange. 2004. A tutorial on MM algoritms. *The American Statistician* 58: 30–7.

Itakura, F., and S. Saito. 1968. Analysis synthesis telephony based on the maximum likelihood method. In *Proceedings of the International Conference on Acoustics*, Tokyo, Japan, C17–20.

Jutten, C., and P. Comon (Eds.) 2010. *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Amsterdam, NL: Elsevier.

Kamoeka, H., N. Ono, K. Kashino, and S. Sagayama. 2009. Complex NMF: A sparse new representation for acoustic signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 3437–40.

Klingenberg, B., J. Curry, and A. Dougherty. 2009. Non-negative matrix factorization: Ill-posedness and a geometric algorithm. *Pattern Recognition* 42: 918–28.

Langville, A. N., C. D. Meyer, and R. Albright. 2006. Initialization for the nonnegative matrix factorization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Lee, D., M. Hill, and H. Seung. 2001. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems* 13: 556–62.

Lee, D., and H. Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401: 788–91.

Li, Y., J. Woodruff, and D. Wang. 2009. Monaural musical sound separation based on pitch and common amplitude modulation. *IEEE Transactions on Audio, Speech, and Language Processing* 17 (7): 1361–71.

Lyon, R. 1982. A computational model of filtering, detection, and compression in the cochlea. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Paris, France, 1282–5.

Lyon, R. 2010. Machine hearing: An emerging field. *IEEE Signal Processing Magazine* 27 (5): 131–9.

Maher, R., and J. Beauchamp. 1990. An investigation of vocal vibrato for synthesis. *Applied Acoustics* 30: 219–45.

Marin, C., and S. McAdams. 1991. Segregation of concurrent sounds II: Effects of spectral envelope tracing, frequency modulation coherence, and frequency modulation width. *Journal of the Acoustical Society of America* 89: 341–51.

McAdams, S. 1989. Segregation of concurrent sounds I: Effects of frequency modulation coherence. *Journal of the Acoustical Society of America* 86: 2146–59.

McAulay, R., and T. Quatieri. 1986. Speech analysis/synthesis based on sinusoidal representation. *IEEE Transactions on Audio, Speech, and Language Processing* 34: 744–54.

Mehrotra, S. 1992. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization* 2: 575–601.

Moorer, J. 1977. The use of the phase vocoder in computer music applications. *Journal of the Audio Engineering Society* 26: 42–5.

Paatero, P., and U. Tapper. 1994. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. In *Proceedings of the International Conference on Statistical Methods for the Environmental Sciences*, Espoo, Finland, 111–26.

Plumbley, M., S. Abdallah, J. Bello, M. Davies, G. Monti, and M. Sandler. 2002. Automatic music transcription and audio source separation. *Cybernetics and Systems* 33: 603–27.

Portnoff, M. 1976. Implementation of the digital phase vocoder using the fast Fourier transform. *IEEE Transactions on Audio, Speech, and Language Processing* 24: 243–8.

Rigaud, F., A. Falaize, B. David, and L. Daudet. 2013. Does inharmonicity improve an NMF-based piano transcription model? In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, 11–5.

Serra, X., and J. Smith. 1990. Spectral modeling synthesis: A sound analysis/synthesis system based on deterministic plus stochastic decomposition. *Computer Music Journal* 14 (4): 12–24.

Smaragdis, P. 2004. 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*, Grenada, Spain, 494–9.

Smaragdis, P., and J. Brown. 2003. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 177–80.

Smaragdis, P., B. Raj, and M. Shashanka. 2008. Sparse and shift-invariant feature extraction from non-negative data. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, Las Vegas, NV, 2069–72.

Smith, J., and X. Serra. 1987. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the International Conference on Computer Music*, Urbana, IL, 290–7.

Verfaille, V., C. Guastavino, and P. Depalle. 2005. Perceptual evaluation of vibrato models. In *Proceedings of the Conference on Interdisciplinary Musicology*, Montreal, Canada.

Vincent, E., and Y. Deville. 2010. Audio applications. In C. Jutten and P. Comon (Eds.), *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 779–819. Amsterdam, NL: Elsevier.

Vincent, E., R. Gribonval, and C. Févotte. 2006. Performance measurements in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing* 14 (4): 1462–9.

Vincent, E., S. Watanabe, J. Barker, J. L. Roux, F. Nesta, and M. Matassoni. 2013. The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, CZE, 162–7.

Virtanen, T. 2007. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech, and Language Processing* 15 (3): 1066–74.

Wang, A. 1995. Instantaneous and frequency-warped techniques for source separation and signal parameterization. In *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 47–50.

Wang, B., and M. Plumbley. 2005. Musical audio stream separation by non-negative matrix factorization. In *Proceedings of the DMRN Summer Conference*, Glasgow, Scotland, 23–4.

Wang, D., and G. Brown. 2006. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Hoboken, NJ: Wiley Interscience.

Wang, Y., and Y. Zhang. 2012. Non-negative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering* 25 (6): 1335–53.

Wild, S., J. Curry, and A. Dougherty. 2004. Improving non-negative matrix factorizations through structured information. *Pattern Recognition* 37 (11): 2217–32.

Yeh, C., A. Roebel, and X. Rodet. 2010. Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals. *IEEE Transactions on Audio, Speech, and Language Processing* 18: 1116–26.

Yen, F., Y.-J. Luo, and T.-S. Chi. 2014. Singing voice separation using spectro-temporal modulation filters. In *Proceedings of the International Conference on Music Information Retrieval*, Taipei, Taiwan, 617–22.