FAKULTÄT FÜR ELEKTROTECHNIK
UND INFORMATIONSTECHNIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis

Semi-Supervised Suppression of Background Music in Monaural Speech Recordings

Jordi Feliu Hurtado

FAKULTÄT FÜR ELEKTROTECHNIK UND INFORMATIONSTECHNIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis

Semi-Supervised Suppression of Background Music in Monaural Speech Recordings

Author: Jordi Feliu Hurtado
Advisor: Dipl.-Inf. Felix Weninger

Date: May 31, 2011

Acknowledgments

First of all, I would like to thank my supervisor and advisor Dipl.-Inf. Felix Weninger for let me do the thesis about this interesting topic and for all his help and advices.

Abstract

After a presentation of Non-Negative Matrix Factorization (NMF) and its applications in audio processing, we introduce a semi-supervised algorithm NMF based to improve separation of speech from background music in monaural signals. In this approach, fixed speech basis vectors are obtained from training data whereas music bases are estimated iteratively to cope with spectral variability. A small number of NMF components is used for decreased computation effort and most important NMF parameters are optimized, as the DFT window size used for transformation to the frequency domain. Extensive experimental validation with 168 speakers from the TIMIT database test set and four different music genres mixed at various speech-to-music ratios reveals that the semi-supervised method outperforms conventional supervised NMF for low speech-to-music ratios and low music bases, and that sparsity constraints on the music bases to enforce harmonicity can further improve separation performance depending on the music style.

Contents

List of Figures

List of Tables

Part I.

Introduction and Theory

1. Introduction

Consider the situation where one wants to apply Automatic Speech Recognition (ASR) in the presence of background music, as inside the car while one is listening to music or watching TV at home. A human has no problem separating the speech of this person from the background music but the ASR system fails since music is disturbing the frequency spectrum in a selective way.
The aim of this thesis is to find an appropriate method to separate the speech from the music in order to improve the ASR with a low computational cost tool. The optimal results would be like the image 2.1, from a mixed signal done with a short sentence of a known speaker and a musical fragment, obtaining the independent sources perfectly separated.

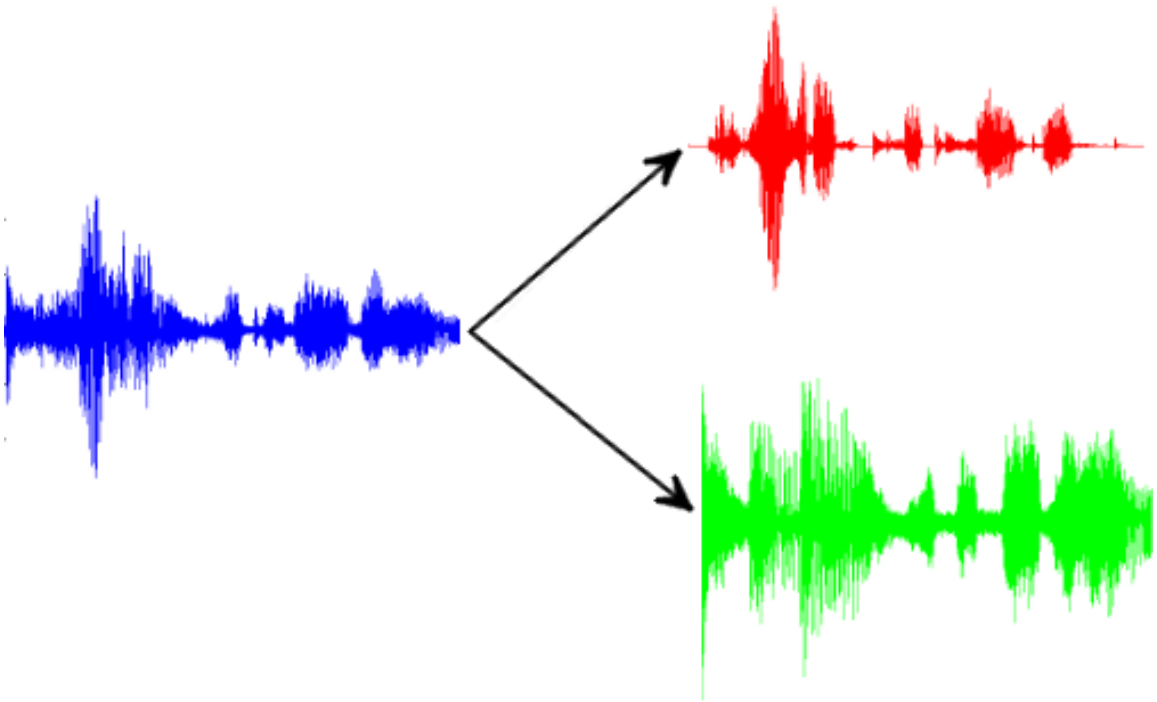


Figure 1.1.: Separation of speech from music. The blue graphic is the mixed signal, the red one is the speech separated signal and the green one is the music separated signal.

In the following section, the main methods used, so far, in source separation of speech from background music are cited and briefly described.

1.1. Source separation of speech from background music

Separating speech from non-stationary noises remains a difficult problem without a satisfactory solution to date. Although speech and music can be robustly distinguished by acoustic parameters [20], separation of both sources has not an adequate solution. The similar spectral characteristics of some parts of both sources make more difficult extract the speech without quality loss and a complete suppression of music. A robust method to suppress background music would have a large variety of applications, like speech enhancement for in-car human-machine interfaces before using a voice activity detector (e.g., [30]), speech enhancement for mobile telephony in highly noisy environments such as discotheques, speech recognition for multimedia information retrieval in TV series or on-line videos, or even lyrics transcription of rap/hip-hop music.

A great deal of approaches achieved significant improvements in speech from stationary or slowly-varying noises separation. In [2], an estimation of the average noise magnitude is calculated during non-speech activity to clean speech by subtraction the noise magnitude spectrum from the noisy speech spectrum, in [7], a noise reduction from the speech signal itself is done in a different way. The problem resides in non-stationary noises, as music. Exist a lot of methods for isolation of vocal parts in music (e. g., [5]), others, treat to extract speech in the presence of music sources in a multi-microphone scenario (e. g., [29]), but it is from recently that first relevant results for monaural background music suppression have been obtained in [24] using convolutive NMF for speech de-noising. In [18], the speech extraction is done using an exemplar-based approach based on supervised NMF taking a large set of speech and music spectral bases from training data and changing iteratively its corresponding activations in order to obtain the estimation of the speech signal.

Our approach is a modification of the last citation, we are using a semi-supervised variant of NMF, pre-defining only the speech spectra while the music bases are randomly initialized and changing iteratively during the separation, to cope with variability of the music over time. A low set of speech and music bases is used to decrease computational effort compared to exemplar-based approaches. Furthermore, we add sparsity constraints on the music activations in order to improve discrimination of speech and music bases, similar to [27] approach, and develop this algorithm to add sparsity constraints on the music spectral bases to enforce harmonicity in music spectra.

We also execute a supervised NMF separation to compare with the semi-supervised method. In this method, music bases are pre-defined from training data formed of parts of the ground truth music.

The speech data is from the TIMIT test set database and all the experiments are speaker-dependent using the different sentences that the TIMIT database offers

for each speaker. Four different music styles are evaluated: classical, jazz, latin and pop rock.

Following, the structure of the thesis is clearly presented:

- Chapter 2 introduces NMF as an algorithm for audio processing tasks and its mathematical and algorithmic background as well as its application for suppression of music in monaural speech recordings.
- Chapter 3 defines corpora and noisy datasets, experimental parameters, evaluation methodology and procedure of the experiment.
- In Chapter 4, the semi-supervised approach is introduced with its corresponding separation results. The most important experimental parameters are evaluated as well as the music style dependency.
- In Chapter 5, a supervised approach is done and used as an upper benchmark to compare its results with the semi-supervised results.
- Chapter 6 extends the semi-supervised approach by adding sparsity constraints to the NMF matrices. Different configurations are tested and compared with the non-sparse approach.
- In Chapter 7 the before mentioned approaches are compared and some specific separation examples are evaluated individually in order to know how the sung voice is affecting the separation performance.
- Finally, in Chapter 8, the conclusions are presented.

2. Theory

Basis decompositions have been an important tool in signal processing in recent years. A lot of different applications have been developed with a wide variety of approaches to obtain bases. Some well known examples are the Principal Component Analysis (PCA) [12], the Independent Component Analysis (ICA) [11] or the Singular Value Decomposition (SVD) [14] algorithms. Other recent works are exploiting the statistical characteristics of each source to separate it from the others, like in [17], where expectation-maximization (EM) algorithms are used, or in [8], where is combined with sparseness constraints as we will use in our approach.

Another basis decomposition approach is Non-negative Matrix Factorization (NMF) [16], which we are using for our experiments. It was first introduced by Lee and Seung, originally proposed for image decomposition [15] and it is an increasingly popular algorithm in signal processing, particularly in the fields of speech and music processing where these decompositions are used on the magnitude spectra of monaural recordings. In the literature, we can find a great deal of recent publications in the context of source separation where NMF based algorithms are used (e.g., [8], [4] or [13]). Many different variants of the basic NMF algorithm are theoretically explained in [16] and [3]. In the following sections we will describe in detail the methods that we will use for suppressing background music from speech recordings. Firstly, the basic NMF algorithm 2.1, followed by the supervised and semi-supervised description in 2.2, before finishing with the sparse approach in Section 2.3.

2.1. Basic NMF Algorithm

Non-Negative Matrix Factorization is a linear basis decomposition approach that assumes non-negativity of both the basis and the data to be approximated. Its formulation is as follows. Having a non-negative matrix $\mathbf{V} \in \mathbb{R}_+^{M \times N}$ and a constant $R \in \mathbb{N}$, the goal is to approximate the matrix as a product of two also non-negative matrices $\mathbf{W} \in \mathbb{R}_+^{M \times R}$ and $\mathbf{H} \in \mathbb{R}_+^{R \times N}$, such that

$$\mathbf{V} \approx \mathbf{WH} \tag{2.1}$$

where $\mathbf{W}$ represents the spectra of the events occurring in the signal and $\mathbf{H}$ their time-varying gains.

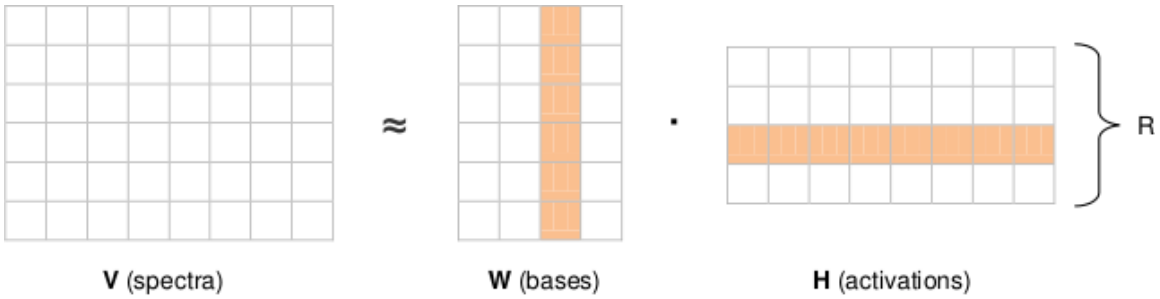


Figure 2.1.: Schema of NMF basis decomposition. R components are the result of multiply each column of **W** with its corresponding row in **H**.

The number of components R, will be one of the most important parameters to choose during experiments. Using one component, we obtain a rank-1 approximation of the input, if we use M components we can achieve a perfect reconstruction of the input, as R is reduced we start obtaining low-rank approximations. By examination of NMF decomposition results in other publications as [25], we can see how the columns of **W** contain the spectral bases and tend to reveal the vertical structure of the input, while their corresponding rows in **H** contain the temporal information and reveal the horizontal structure. In the above mentioned publication, an approach is presented for polyphonic music transcription. The author is using one component for each piano note, then, every resulting component is a variation of the original song where only one frequency note is played in its corresponding moment in time.

In [23], the same author is improving his method introducing Non-Negative Matrix Deconvolution (NMD) in order to impose a temporal structure to the frequency description of each auditory object. In NMF the spectra is constraint to be static and certain characteristics of their spectral evolution is lost, depending on the DFT window size that we are using each object can be represented in this window or not. With the NMD representation, using different **W** matrices, the spectrum of one object is described time after it has begun. The same author is evaluating its application to supervised speech separation in [24], it is shown that the most important parameters to combine are the number of components, the DFT window size and the number of spectra for NMD. In our very first experiments, we tested out that NMD was not always improving the music suppression and required more computational effort, for these reasons we decided to use NMF in our experiments instead of NMD.

Speech signal is more difficult to be modelled than piano music for example, we don't have different notes to know how many components gives the best separation performance. Although a higher number of components is not considered harmful as the superfluous components' contributions to the whole magnitude spectrum will be nearly zero, a higher number of components results in smaller absolute values and thus less maximum amplitudes of the separated components.

In the above mentioned publication, different number of components are checked for speech separation, giving good results using a low number of speech components depending on the other parameters. In the case of the music, the number of components is chosen according to our experience, different experiments were done in order to choose the better option, we got different results for supervised and semi-supervised NMF. In [22], the problem of choosing the number of music components is treated for blind enhancement of the rhythmic and harmonic parts of music recordings.

After this introduction about NMF and the meaning of its components we will show the equations necessary to use this method. To apply NMF we are using *openBliSSART* [28], a framework and toolbox for Blind Source Separation for Audio Recognition Tasks.

As a non-negative matrix is needed, NMF is applied in the frequency domain, by factorizing magnitude spectrogram matrices obtained by short-time Fourier transformation (STFT). Thereby the signal is split into overlapping frames of constant size. Each frame is multiplied by a window function and transformed to the frequency domain using Discrete Fourier Transformation (DFT), with transformation size equal to the number of samples in each frame. Examples of window functions are the rectangular window, the Hann window as well as the square root of the Hann function, the latter was used in [8] and it is which we are using in our experiments:

$$h(k) = \sqrt{0.5 - 0.5\cos\left(\frac{2\pi k}{T-1}\right)} \tag{2.2}$$

After transformation, the magnitudes of the DFT coefficients are put in the columns of the V matrix. Denoting the number of columns by N and the DFT window size by T, considering the symmetry of the coefficients, the number of rows of V will be $M = \lfloor T/2 \rfloor + 1$.

An iterative minimization of a cost function is done in order to factorize the input according to equation 2.1.

$$(\mathbf{W}, \mathbf{H}) = \arg\min_{\mathbf{W},\mathbf{H}} c(\mathbf{W}, \mathbf{H}) \tag{2.3}$$

We obtain different variants of NMF only by changing the cost function. This function is measuring the reconstruction error between the product of the NMF factors and the original matrix. The cost function that we are using consists of a modified version of Kullback-Leibler (KL) divergence:

$$c(\mathbf{W}, \mathbf{H}) = \sum_{i=1}^{M}\sum_{t=1}^{N}\left(\mathbf{V}_{i,t}\log\frac{\mathbf{V}_{i,t}}{(\mathbf{WH})_{i,t}} - (\mathbf{V} - \mathbf{WH})_{i,t}\right) \tag{2.4}$$

For minimization of the cost function, $\mathbf{W}$ and $\mathbf{H}$ are iteratively modified using the following 'multiplicative update' rules:

$$\mathbf{H}_{j,t} \leftarrow \mathbf{H}_{j,t} \frac{(\mathbf{W}^T(\mathbf{V}./\mathbf{WH}))_{j,t}}{(\mathbf{W}^T\mathbf{1})_{j,t}} \qquad j = 1, \ldots, R; t = 1, \ldots, N \tag{2.5}$$

$$\mathbf{W}_{i,j} \leftarrow \mathbf{W}_{i,j} \frac{((\mathbf{V}./(\mathbf{WH}))\mathbf{H}^T)_{i,j}}{(\mathbf{1}\mathbf{H}^T)_{i,j}} \qquad i = 1, \ldots, M \tag{2.6}$$

where $\mathbf{1}$ is an all-unity matrix and $./$ indicates element-wise division. The above matrix formulation has been shown to yield better performance than the scalar product formulations in [16] when using fast implementations of matrix multiplication.

The update rules are applied for 100 iterations starting from a (Gaussian) random solution. The multiplication of each basis vector $\mathbf{W}_{:,j}$ with its activation $\mathbf{H}_{j,:}$ represents each component $\mathbf{V}^{(j)}$. For reconstruction, a *Wiener filter* approach has been used:

$$\mathbf{V}^{(j)} = \mathbf{V} \otimes \frac{\mathbf{W}_{:,j}\mathbf{H}_{j,:}}{\mathbf{WH}} \qquad j = 1, \ldots, R \tag{2.7}$$

In recent works (e.g., [26]), co-occurrence constraints are added to the multiplicative update rules in order to enforce dependence within predetermined groups of bases. This modification is used to represent objects using multiple spectral bases because some of them may require more than one to be approximated accurately. Since is not demonstrated that the co-occurrence constraints can improve speech separation we are not using it in our experiments.

2.2. Source separation by supervised and semi-supervised NMF

In our very first experiments, an unsupervised NMF based approach for blind source separation similar to [8] was tested. Using this method, both NMF matrices were randomly initialized and changing iteratively during the separation. At the end of the separation, features were extracted from the separated components before a support vector machine (SVM) classifier sorted out each component as speech or music. We tried different configurations changing the number of components but the results were not so good since the separated source signals were so much corrupted for the undesired source.

After assume that blind source separation was giving unsatisfying results we start thinking about supervised NMF approaches. The following signal model will help us to understand the supervised separation.

As we assume that speech is corrupted by addition of background music, we can write the input as follows:

$$\mathbf{V} = \mathbf{V}^{(s)} + \mathbf{V}^{(m)}, \tag{2.8}$$

where $\mathbf{V}^{(s)}$ is the spectrogram of the original speech signal, and $\mathbf{V}^{(m)}$ is the original music spectrogram.

As we explained before, the speech and music spectrograms can be modelled as linear combinations of base spectra $\mathbf{w}_j^{(s)} \in \mathbb{R}_+^M$, $j = 1, \ldots, R^{(s)}$, and $\mathbf{w}_j^{(m)}$, $j = 1, \ldots, R^{(m)}$ respectively. Defining

$$\mathbf{W}^{(s)} = [\mathbf{w}_1^{(s)} \ \cdots \ \mathbf{w}_{R^{(s)}}^{(s)})] \tag{2.9}$$

and

$$\mathbf{W}^{(m)} = [\mathbf{w}_1^{(m)} \ \cdots \ \mathbf{w}_{R^{(m)}}^{(m)})] \tag{2.10}$$

obtaining the following matrix notation of this signal model:

$$\mathbf{V} \approx \mathbf{\Lambda} = \mathbf{\Lambda}^{(s)} + \mathbf{\Lambda}^{(m)} = \mathbf{W}^{(s)}\mathbf{H}^{(s)} + \mathbf{W}^{(m)}\mathbf{H}^{(m)}, \tag{2.11}$$

or $\mathbf{\Lambda} = \mathbf{WH}$ for $\mathbf{W} := [\mathbf{W}^{(s)}\mathbf{W}^{(m)}]$, $\mathbf{H} := \begin{bmatrix} \mathbf{H}^{(s)} \\ \mathbf{H}^{(m)} \end{bmatrix}$.

In the supervised separation, we assume that speech and music bases, $\mathbf{W}^{(s)}$ and $\mathbf{W}^{(m)}$ respectively, are fixed after estimation from training data. The extraction of bases from prior information is explained in Section 3.4.

In the semi-supervised case, only the speech bases are initialized with training data, while the music bases are randomly initialized such as the $\mathbf{H}$ matrix.

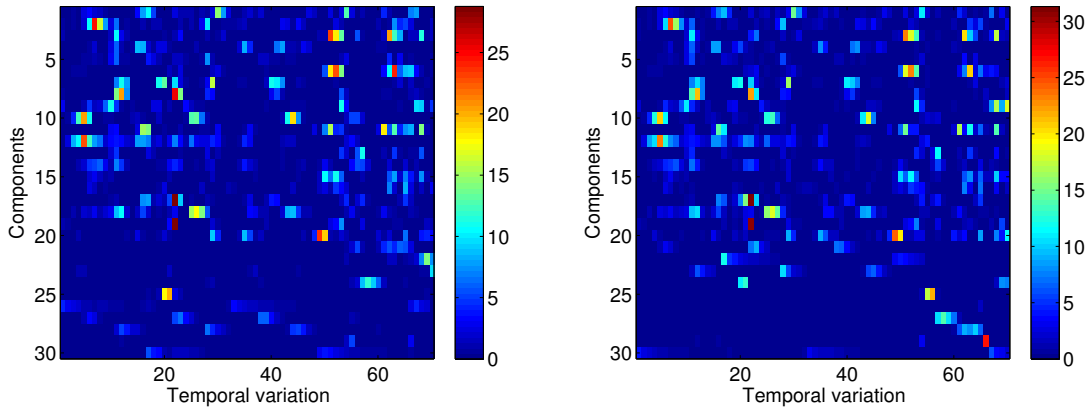
We are using a procedure similar to [24] for our supervised approach but in speech and music mixtures. In the before mentioned experiment, the author is separating the speaker sources using a priori information of all of them. The trained bases are the result to apply NMF to the training data for all the independent sources, the union of all these bases will form the $\mathbf{W}$ matrix.

Related to semi-supervised NMF, in [18], a supervised and a semi-supervised approach using overcomplete dictionaries consisting of random exemplars of training data are evaluated. In our first experiments, we tried an approach using overcomplete speech and music dictionaries, extracting trained bases from all the different speakers we tested and from different music styles too. The results showed that the system were not able to identify the right bases necessary to extract the speech of the known speaker and suppress robustly the music. Because of that we decided to test out other approaches such as the speaker dependent semi-supervised approach which we are explaining in Chapter 4.

2.3. Sparse semi-supervised NMF

Sparse NMF is based on a variation of the original NMF cost function, which measures the reconstruction error. The concept of sparse coding [1, 6, 10] refers to a representational scheme where only a few units—out of a large population—are

effectively used to represent typical data vectors. In effect, this implies most units taking values close to zero while only few of them take significantly non-zero values. The following images show the effect to add sparsity constraints in a part of a matrix:



((a)) ((b))

Figure 2.2.: Two H matrices at the end of a separation with the same conditions but, in the ((b)) matrix, sparsity constraints were added to the last 10 rows of the matrix.

To add sparsity the following cost function is minimized:

$$c(\mathbf{W}^{(m)}, \mathbf{H}) = c_r(\mathbf{W}^{(m)}, \mathbf{H}) + \lambda\, c_s^{\mathbf{H}}(\mathbf{H}^{(m)}) + \mu c_s^{\mathbf{W}}(\mathbf{W}^{(m)}) \tag{2.12}$$

where c_r corresponds to the reconstruction error of the extended Kullback-Leibler divergence 2.4, the added cost functions are

$$c_s(\mathbf{W}^{(m)}) = \sum_{j=1}^{R^{(m)}} \frac{1}{\sigma(\mathbf{W}^{(m)}_{:,j})} \sum_{k=1}^{M} \mathbf{W}^{(m)}_{k,j} \tag{2.13}$$

$$c_s(\mathbf{H}^{(m)}) = \sum_{j=1}^{R^{(m)}} \frac{1}{\sigma(\mathbf{H}^{(m)}_{j,:})} \sum_{t=1}^{N} \mathbf{H}^{(m)}_{j,t} \tag{2.14}$$

and λ and μ are positive factors to weight the previous functions ($0 \leq \lambda, \mu \ll 1$), and $\sigma(\mathbf{W}^{(m)}_{:,j})$ and $\sigma(\mathbf{H}^{(m)}_{j,:})$ are standard deviation estimates for the j-th column of $\mathbf{W}^{(m)}$ and the j-th row of $\mathbf{H}^{(m)}$, respectively that are introduced to avoid dependency on the scaling of the matrices, following [27]. In Eq. 2.12, the term $\mathbf{W}^{(s)}$ not appears because as a semi-supervised approach, speech bases will not change during the separation contrary to $\mathbf{W}^{(m)}$ and $\mathbf{H}$.

In our approach, sparsity constraints were applied only on the music part. The intention of enforcing sparsity on $\mathbf{H}^{(m)}$ is to palliate the fact that the algorithm can 'mis-use' the bases specified to discriminate the music for modelling the speech parts; furthermore, sparsity on $\mathbf{W}^{(m)}$ is imposed to increase the discrimination between speech and music, as the latter is arguably characterized by higher harmonicity compared to speech.

The cost function (2.12) is minimized by applying component-wise multiplicative updates to $\mathbf{W}^{(m)}$, $\mathbf{H}^{(s)}$ and $\mathbf{H}^{(m)}$ based on the algorithm proposed in [27]. We straightforwardly extend the algorithm to the semi-supervised case, including the sparsity constraint for the spectra $\mathbf{W}^{(m)}$ which was not considered in [27], yielding the following update rule for $\mathbf{W}^{(m)}$ and $\mathbf{H}$:

$$\mathbf{W}^{(m)} \leftarrow \mathbf{W}^{(m)} \otimes \frac{\nabla c^{-}(\mathbf{W}^{(m)}, \mathbf{H})}{\nabla c^{+}(\mathbf{W}^{(m)}, \mathbf{H})} \tag{2.15}$$

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{\nabla c^{-}(\mathbf{W}^{(m)}, \mathbf{H})}{\nabla c^{+}(\mathbf{W}^{(m)}, \mathbf{H})} \tag{2.16}$$

where $\otimes$ indicates Hadamard product; since the gradient of the cost function can be written as a subtraction

$$\nabla c(\mathbf{W}^{(m)}, \mathbf{H}) = \nabla c^{+}(\mathbf{W}^{(m)}, \mathbf{H}) - \nabla c^{-}(\mathbf{W}^{(m)}, \mathbf{H}) \tag{2.17}$$

of element-wise non-negative terms

$$\nabla c^{+}(\mathbf{W}^{(m)}, \mathbf{H}) = \nabla c_r^{+}(\mathbf{W}^{(m)}, \mathbf{H}) + \lambda \nabla c_s^{+\mathbf{H}}(\mathbf{H}^{(m)}) + \mu \nabla c_s^{+\mathbf{W}}(\mathbf{W}^{(m)}) \tag{2.18}$$

$$\nabla c^{-}(\mathbf{W}^{(m)}, \mathbf{H}) = \nabla c_r^{-}(\mathbf{W}^{(m)}, \mathbf{H}) + \lambda \nabla c_s^{-\mathbf{H}}(\mathbf{H}^{(m)}) + \mu \nabla c_s^{-\mathbf{W}}(\mathbf{W}^{(m)}). \tag{2.19}$$

Defining the gradients of the reconstruction error for the $\mathbf{H}$ matrix,

$$\nabla c_r^{+}(\mathbf{H}) = \mathbf{W}^T \mathbf{1} \tag{2.20}$$

$$\nabla c_r^{-}(\mathbf{H}) = \mathbf{W}^T(\mathbf{V}./(\mathbf{\Lambda})) \tag{2.21}$$

where $./$ indicates element-wise division and $\mathbf{\Lambda} = \mathbf{WH}$; and the gradients of the $\mathbf{H}$ sparseness terms,

$$[\nabla c_s^{+}(\mathbf{H}^{(m)})]_{j,t} = \frac{\sqrt{N}}{\sqrt{\sum_{k=1}^{N} \mathbf{H}_{j,k}^2}} \qquad j = 1, \ldots, R^{(m)}; t = 1, \ldots, N \tag{2.22}$$

$$[\nabla c_s^{-}(\mathbf{H}^{(m)})]_{j,t} = \mathbf{H}_{j,t} \frac{\sqrt{N} \sum_{k=1}^{N} \mathbf{H}_{j,k}}{(\sum_{k=1}^{N} \mathbf{H}_{j,k}^2)^{3/2}} \tag{2.23}$$

as laid out in [27], and the reconstruction error for the $\mathbf{W}^{(m)}$ matrix,

$$\nabla c_r^+(\mathbf{W}^{(m)}) = (\mathbf{1}\mathbf{H}^T) \quad (2.24)$$

$$\nabla c_r^-(\mathbf{W}^{(m)}) = ((\mathbf{V}./\mathbf{\Lambda})\mathbf{H}^T) \quad (2.25)$$

and the gradients of the $\mathbf{W}^{(m)}$ sparseness terms,

$$[\nabla c_s^{\mathbf{W}^+}(\mathbf{W}^{(m)})]_{i,j} = \frac{\sqrt{M}}{\sqrt{\sum_{k=1}^{M} \mathbf{W}_{k,j}^{(m)2}}} \qquad i = 1, \ldots, M; j = 1, \ldots, R^{(m)} \quad (2.26)$$

$$[\nabla c_s^{\mathbf{W}^-}(\mathbf{W}^{(m)})]_{i,j} = \mathbf{W}_{i,j}^{(m)} \frac{\sqrt{M} \sum_{k=1}^{M} \mathbf{W}_{k,j}^{(m)}}{(\sum_{k=1}^{M} \mathbf{W}_{k,j}^{(m)2})^{3/2}}. \quad (2.27)$$

In our experiments, different configurations were used enforcing sparsity constraints in $\mathbf{W}^{(m)}$, $\mathbf{H}^{(m)}$ or both matrices.

Part II.

Experiments and results

3. Corpora and noisy dataset, experimental parameters, evaluation methodology and procedure of the experiment

After the theoretical study of some NMF-based algorithms, it is time to see how this technique has been used and which performance provides. Firstly, we will describe the datasets used in the experiments, the experimental parameters in order to understand the relevance of each one, the evaluation methodology that we will apply to the separation results, and the procedure of the experiment.

3.1. Corpora and noisy dataset

We performed speech from music separation on digital mixtures. Each mixture is artificially mixed using SoX[1].

The speech material consists of 1680 mixtures of 168 different english speakers (56 females and 112 males) from the TIMIT database test set. There are 10 sentences for each speaker. Each sentences is around 3 seconds long. As a supervised experiment, all these audio files are used for testing or training depending on the iteration, 1 sentence of the known speaker is used for testing and the other 9 for training.

The test utterances are corrupted by digital addition of music. For the music, we used 4 different styles: classical, jazz, latin and pop rock.

The classical music database is formed by 136 Viennese Waltz from the Ballroom Dance (BRD) database [21], each one is 30 seconds long. A random cut of a Waltz with the same duration of the speech file is used for testing and, for Supervised NMF, the concatenation of the two resulting parts is used for training.

The jazz music database is formed by 132 songs from different albums of jazz standards. The duration of each song is variable, there are more than 6 hours of music.

[1]SoX is a command line utility that can play, mix, concatenate and apply various effects to audio files.

The latin music database is formed by 136 songs from different albums. There are more than 6 hours of latin music.

The pop rock music database is formed by 136 songs of the MTV Top 10 of 80's and 90's. There are also more than 6 hours of pop rock music.

The songs of jazz, latin and pop rock databases are used for testing and also for training in supervised separations. The song is randomly chosen as well as the cut of the song that we will use for the mixture, then, 25 seconds of training material are extracted from the remaining parts of the same song.

All the material was downsampled from 44.1 to 16kHz sampling frequency and downmixed to mono.

3.2. Experimental parameters

In this section, the most important parameters affecting the separation will be described.

As we explained in the theoretical part, NMF is applied to the audio files transformed to the frequency domain using STFT [2]. The hop size is set to 50% of the DFT size, zero padding is not used, before the DFT the data is scaled according to a square root of Hann function, and we estimated the bases and their weights for one hundred iterations. The generator function for initialization of the matrices is Gaussian noise. The DFT window size will be an important parameter to optimize during the next experiments, the following values will be tested in Sections 5.2 and 4.2, DFT windows size = [8ms, 16ms, 32ms, 64ms, 128ms, 256ms, 512ms].

Another important parameter is the number of components. In Sections 5.1 and 4.1 the number of music components will be chosen empirically. For speech, we will use 20 components because it is not a very high number that let me do a fast separation and, in related publications, other researchers have achieved good results.

The mixtures are done with these SMR [3] = [-5dB, 0dB, 5dB, 10dB, 15dB] or these, SMR = [-7.5dB, -5dB, -2.5dB, 0dB, 2.5dB, 5dB], depending on the experiment.

3.3. Evaluating methodology

This Section describes the methods used to evaluate the results. The following notation had been used:

$m(t)$: original speech and music mixed signal
$x_s(t)$: original speech signal
$x_m(t)$: original music signal

[2] Short-Time Fourier Transformation
[3] Speech to Music Ratio

$y_s(t)$: separated speech signal
$y_m(t)$: separated music signal
$r(x(t), y(t))$: correlation between $x(t)$ and $y(t)$.

Using the previous notation, two measures are derived to measure performance, the similarity of the output with the target and the source ratio in dB. The similarity index measures how much the output resembles the desired output. It is measured by taking the correlation of the extracted source with the desired output (the original signal):

$$SI_i = 10 \log_{10} r(x_i(t), y_i(t)) \tag{3.1}$$

where $i = [speech, music]$. The similarity results are compared to the threshold obtained applying the previous operation but to the mixture:

$$th_i = 10 \log_{10} r(x_i(t), m(t)) \tag{3.2}$$

Lower values indicate that the result is not too similar to the desired sentence. Note that this measure is also influenced by the quality of the separation since traits of the undesired source would get lower its value. There will be values lower than zero, being zero the most desired case. If in any case $SI_i \leq th_i$ it would say that this source is more distinguishes in the mixture than in the separation and the system is not working out.

The source ratio is computed by comparing the correlations of the original sources to the extracted sounds:

$$SR_i = 10 \log_{10} \frac{r(x_i(t), y_i(t))}{r(x_j(t), y_i(t))} = SI_i - 10 \log_{10} r(x_j(t), y_i(t)) \tag{3.3}$$

This measure will tell how much the signals of the undesired source have been suppressed. Higher values will reveal better extraction of the desired source. Lower values than zero reveal that the undesired signal is more similar than the desired signal.

In the following experiments, all the results are the average of the previous evaluation measurements for all the sentences of the database.

3.4. Procedure of the experiment

In this section we explain the common parts of all the experiments that we will run during the following three chapters. Depending on the method used to separate the mixtures, the experiment will change but the procedure described below is always the same.

3.4.1. Extracting bases from a training file

Other sentences of the same speaker who is talking in the test file are needed to train the system. The speech training file is made by concatenation of all the others sentences, approximately it is 20 seconds long.

In the supervised case, when a cut is extracted from a song for testing, another 25 seconds long cut is needed for training. For classical music, we are using the concatenation of the resulting two parts, because as we explained before in 3.1, the classical songs are 30 seconds long and we need all the file for training or testing. For the other three music styles, we are using also 25 seconds long music training files extracted from another part of the same song used for testing.

NMF is applied on the training files and the spectral bases are stored to use them in the separation.

3.4.2. Making the mixtures

With the speech file not used previously and a random cut from the known song is done the mixture. The speech and the music file have the same duration and are mixed using SoX. Before the mixing it is time to choose the SMR.

3.4.3. Separating the test files

Once the mixture is done, the next step is to separate it in components with one of NMF-based methods, using the bases obtained during the training. Logically, the number of the components for this separation has to be the addition of the speech and music training components. The separation system is exporting the resulting speech and music components.

Finally, the components of the speech have to be mixed as well as the music components to get the resulting audio files.

4. Semi-supervised separation

In this Chapter, a semi-supervised separation is evaluated. As explained in Section 2.2, we initialized the speech bases while the music bases and the activations are changing iteratively during the separation to approximate the input.

First of all, the music components were chosen in Section 4.1, after that, the DFT window size in 4.2 and finally the experiment was done for all the music styles in Section 4.3.

4.1. Optimizing the number of components

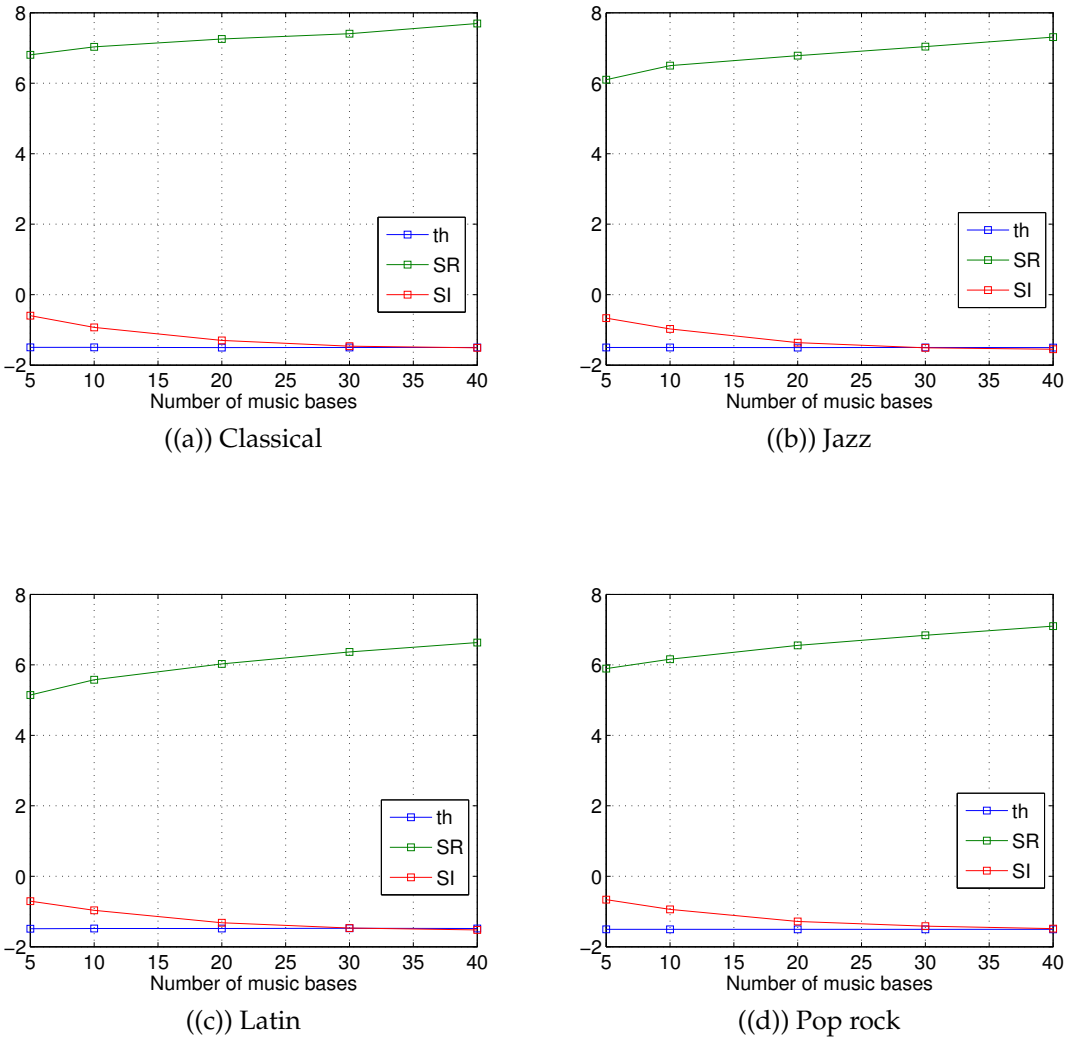
The number of components is obviously an important parameter. We had to choose double because speech and music are completely two different sources. In other publications, one can see that 20 components are the best for a speaker separation [24], moreover, this low number of components allows a quite fast separation, it takes around one second long for each second of audio file with an ordinary desktop computer. In the other hand, we will find the number of music bases necessary to provide a good extraction of music.

In order to find the music components we ran different experiments with the parameters below:

- *Separation method:* Semi-Supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* 0dB.
- *Number of speech components:* 20.
- *Number of music components:* 5, 10, 20, 30 and 40.
- *Music styles:* classical, jazz, latin and pop rock.

The methods to evaluate the experiment are explained in section 3.3.

In the graphics below one can see the separation performance for the different music styles while changing the number of music components:



((a)) Classical ((b)) Jazz

((c)) Latin ((d)) Pop rock

Figure 4.1.: Semi-supervised separation results while changing the number of music components for different music styles.

For all the music styles, the source ratio increase with the number of components but not the similarity. We also can see that the similarity is almost the same for the different music genres while the source ratio is changing, at first glance we can see that suppress classical music is easier than latin music. Higher source ratio means better music suppression and lower similarity means a loss in speech quality during the separation and there is a threshold that we didn't have to go beyond. Knowing these facts, we had to choose depending on the application.

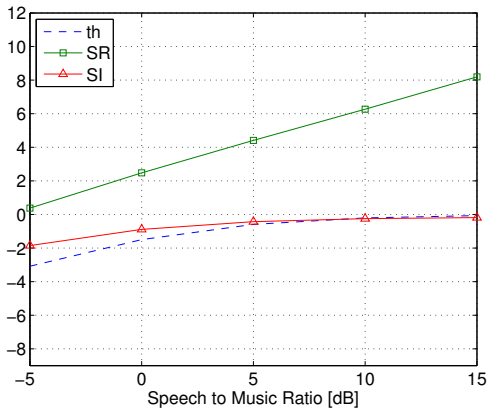
For all the experiments we chose to take 10 music components, because its similarity is distant to the threshold and the speech ratio is better than for 10 components. This decision let us suppress music and don't loose so much speech information.

4.2. Optimizing the DFT window size

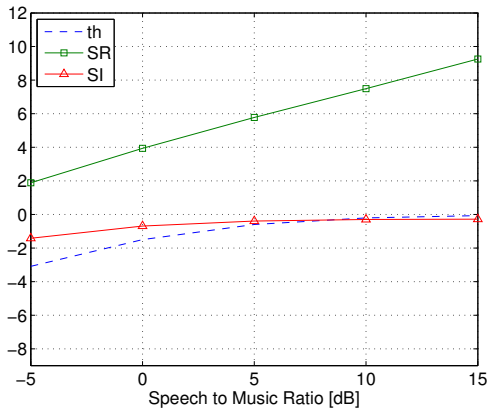
As we explained in 2.1, the DFT window size is, together with the number of components, the most relevant parameters for the separation. We ran the experiments with the parameters below:

- *Separation method:* Semi-Supervised NMF.
- *DFT window size:* 8ms, 16ms, 32ms, 64ms, 128ms and 256ms.
- *Speech to Music Ratio:* -5dB, 0dB, 5dB, 10dB and 15dB.
- *Number of speech components:* 20.
- *Number of music components:* 10.
- *Music styles:* classical.

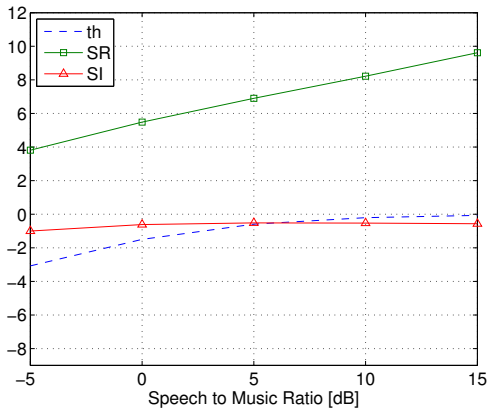
In the following figures one can see the separation performance while we increase the SMR for different DFT window sizes.



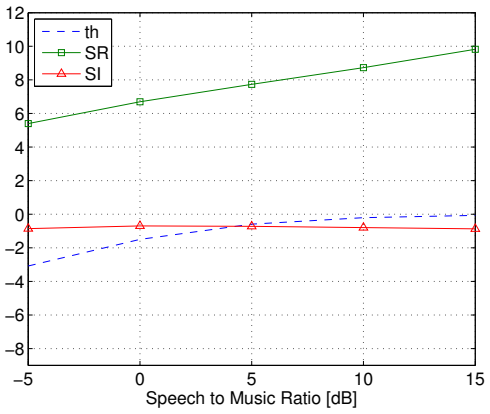
((a)) 8 ms



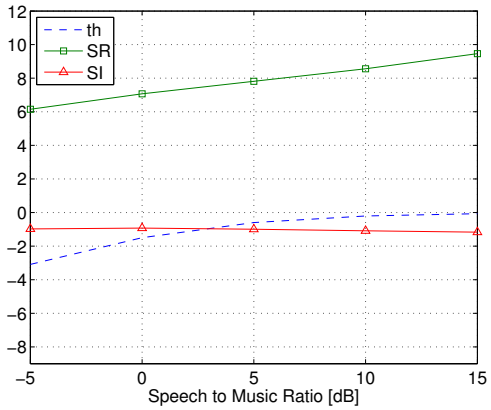
((b)) 16 ms



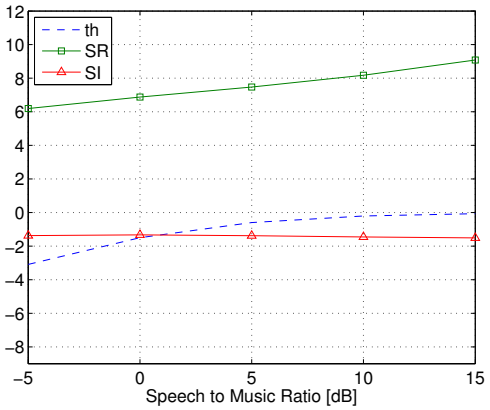
((c)) 32 ms



((d)) 64 ms



((e)) 128 ms



((f)) 256 ms

Figure 4.2.: Evolution of the separation while changing the Speech to Music Ratio for different DFT window size.

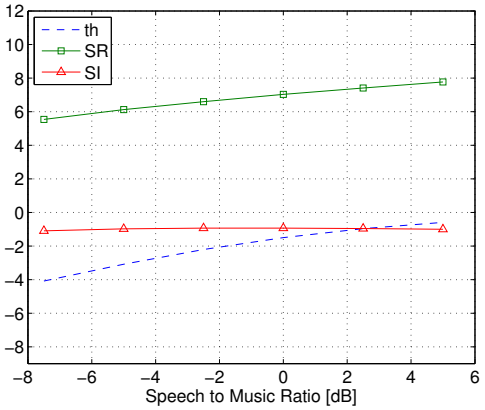
The system improves the separation only when the red line (SI) is upper than the discontinuous blue line (th). The green line (SR) shows us how the music has been suppressed from the speech. The best performance is achieved in the ((e)) graphic, using a DFT window size equal to 128ms but only for $SMR < 5dB$. These results agree with other similar experiments in [24]. We can see that while the SMR ratio is increasing in $5dB$, the SR increases around $1dB$. This fact makes more relevant the separation improvement in very noisy mixtures. With bigger SMR the speech starts losing information and this fact will affect the recognition performance.

4.3. Evaluating the influence of the music style

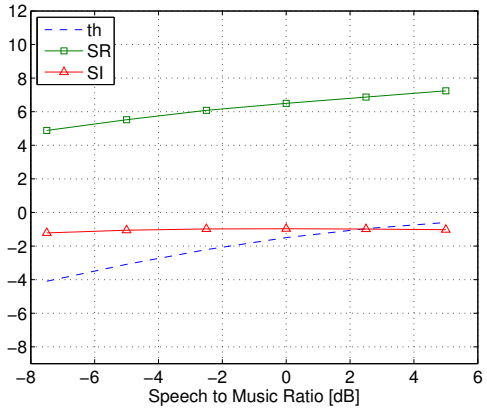
Once we had chosen the number of components and the DFT window size we ran the semi-supervised separation for different music styles to see how affects to the separation performance. Following, the experimental parameters are presented:

- *Separation method:* Semi-Supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* -7.5dB, -5dB, -2.5dB, 0dB, 2.5dB and 5dB.
- *Number of speech components:* 20.
- *Number of music components:* 10.
- *Music styles:* classical, jazz, latin and pop rock.

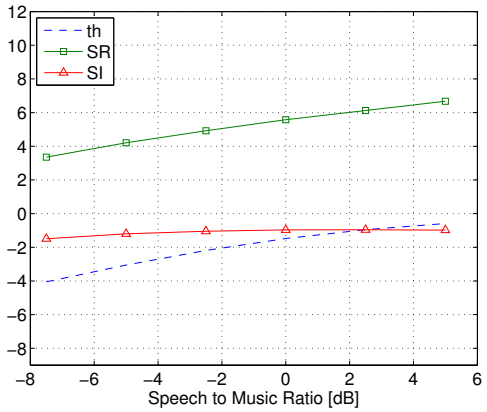
From the following graphics one can see how the quality of the separation varies depending on the music style:



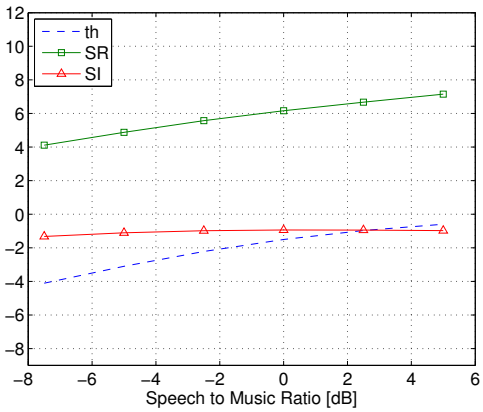
((a)) Classical



((b)) Jazz



((c)) Latin



((d)) Rock

Figure 4.3.: Separation performance on speech corrupted by different music styles. Semi-supervised separation using a DFT window size equal to 128ms.

Up to 2dB can change approximately the SR comparing classical and latin music when the SMR is very low. The most interesting differences among styles that can vary the separation results are:

- *The amoung of sung voice in music.* As we will see in 7.2.
- *The music variability.* As faster varies the music more difficult is to suppress it during the separation because its behaviour is more similar to speech, and so speech and music bases are similar. This fact drives the system to confuse when computing the H matrix because it can use the bases from the wrong source.

Keeping this in mind, it is reasonable that latin music is which results in a worst separation performance, due to the higher amount of sung voice and its fast variability in time of its melodies.

5. Supervised separation

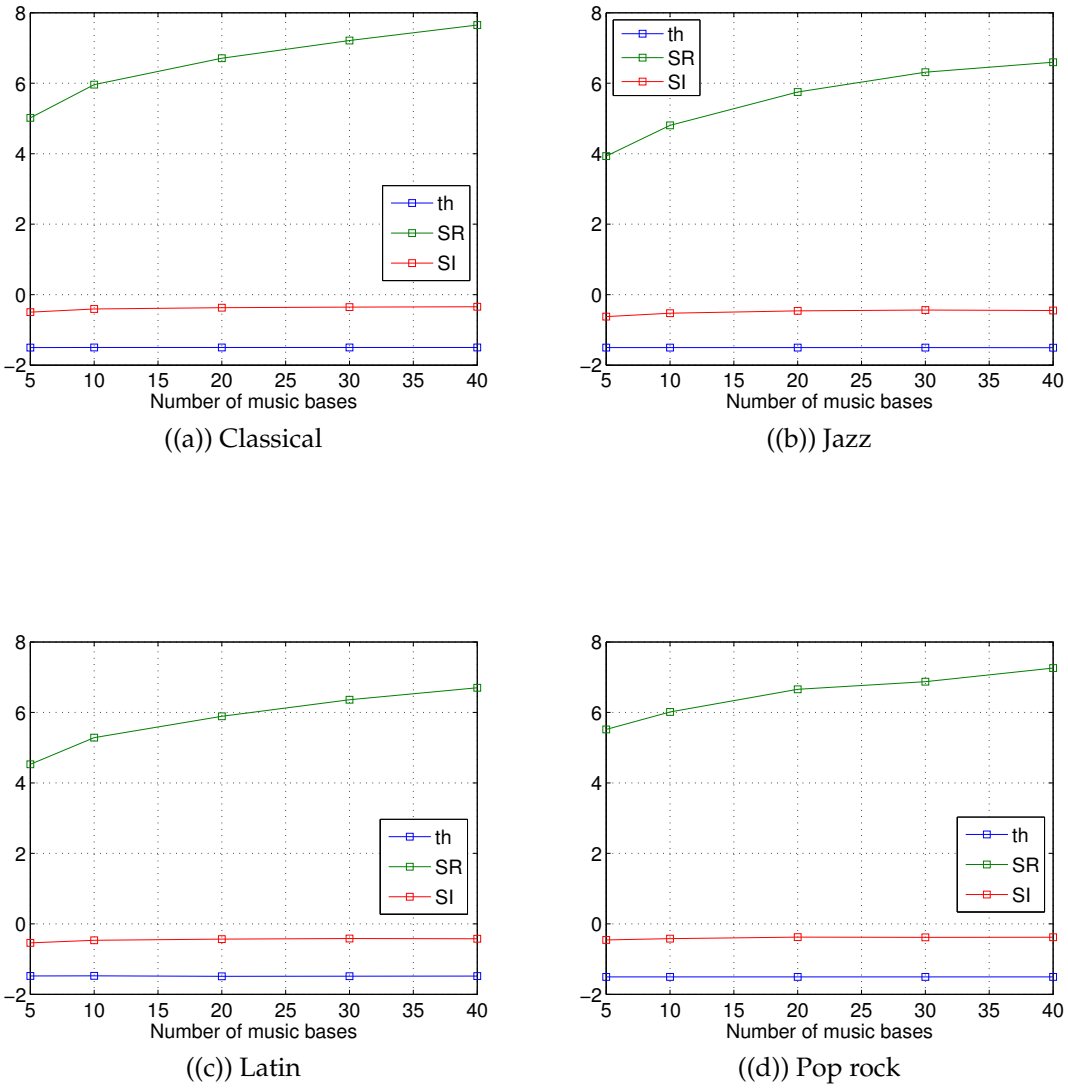
In this Chapter, a supervised NMF approach is used in order to compare it to the semi-supervised separation done in Chapter 4. This experiment is not so much realistic because the music training file used to get the music bases is a cut from the same song used for testing. The aim of this experiment is to know the best performance that one can achieved using supervised approximations for comparing the semi-supervised results.

5.1. Optimizing the number of components

As in the semi-supervised separation, the number of components is obviously an important parameter for this experiments. In the following experiments we will use 20 components for speech and 10 for music because it is what we decided in the semi-supervised separation, see Section 4.1. Anyway, the influence of the number of music components is evaluated in this section for supervised separations using the following parameters:

- *Separation method:* Supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* 0dB.
- *Number of speech components:* 20.
- *Number of music components:* 5, 10, 20, 30 and 40.
- *Music styles:* classical, jazz, latin and pop rock.

The methods to evaluate the experiment are explained in section 3.3.



((a)) Classical

((b)) Jazz

((c)) Latin

((d)) Pop rock

Figure 5.1.: Supervised separation results while changing the number of music components for different music styles.

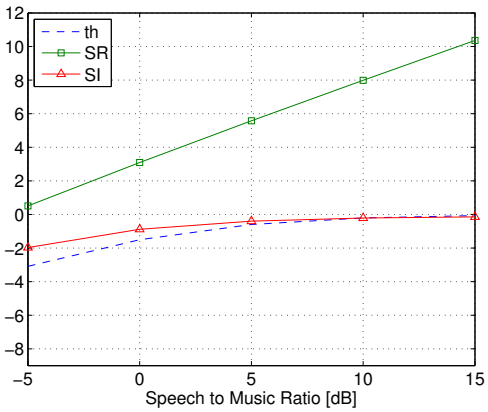
The results show how the similarity is almost constant for the different number of music bases but the source ratio is increasing with this number. Contrary to the semi-supervised results, the use of a great deal of music components improves the separation results, the reason is the characteristics of the music bases, for this experiment, the bases are real information of the song; in the semi-supervised case, the music bases were randomly initialized by the system and changing iteratively to approximate the input. Although 40 music components give better separation performance, the following experiment will be run using 10 music components in order to compare the results to the semi-supervised ones.

5.2. Optimizing the DFT window size

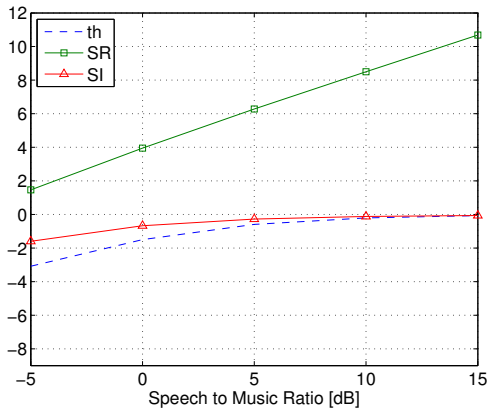
As in the semi-supervised experiment, we have to choose the DFT window size for the supervised approach. Different experiments were run with the parameters below:

- *Separation method:* Supervised NMF.
- *DFT window size:* 8ms, 16ms, 32ms, 64ms, 128ms and 256ms.
- *Speech to Music Ratio:* -5dB, 0dB, 5dB, 10dB and 15dB.
- *Number of speech components:* 20.
- *Number of music components:* 10.
- *Music styles:* classical.

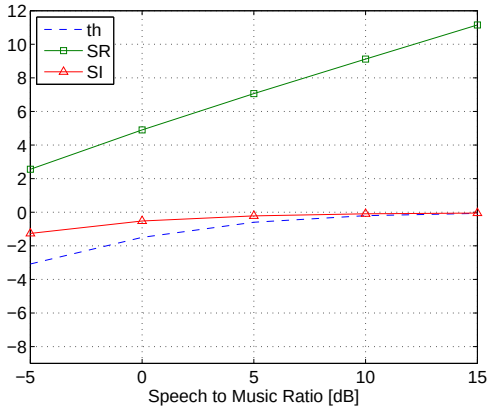
In the following figures one can see the separation performance:



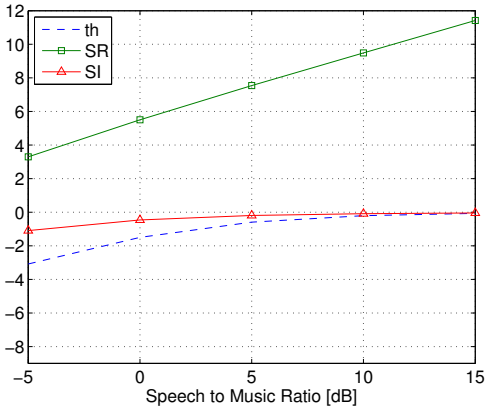
((a)) 8 ms



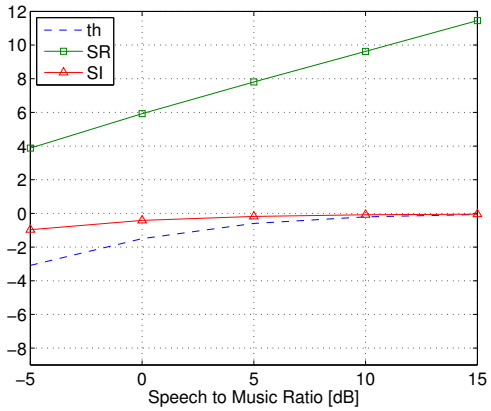
((b)) 16 ms



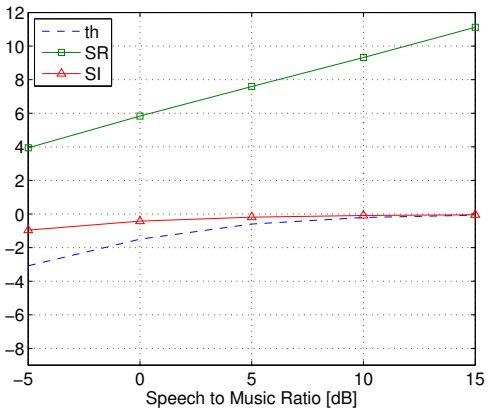
((c)) 32 ms



((d)) 64 ms



((e)) 128 ms



((f)) 256 ms

Figure 5.2.: Evolution of the supervised separation while changing the SMR and the DFT window size in each graphic.

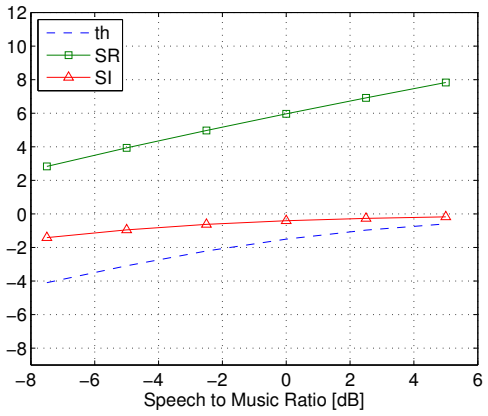
Contrary to the semi.supervised approach, the system is always improving the separation because the red line (SI) is upper than the discontinuous blue line (th) for any SMR. The green line (SR) shows us how the music has been suppressed from the speech. The best performance is achieved in the ((e)) graphic with a DFT window size equal to $128ms$ as in the semi-supervised experiment.

5.3. Evaluating the influence of the music style

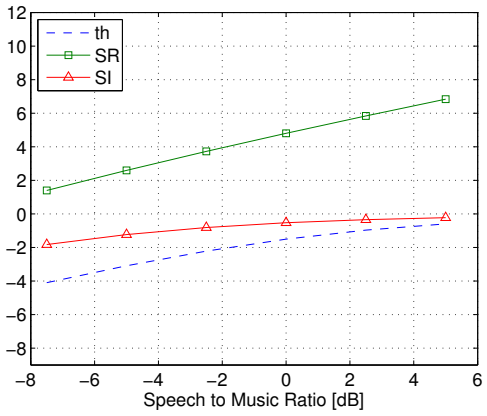
Once we chose the number of components and the DFT window size we ran the supervised separation for different music styles to see how affects to the separation performance. Following, the experimental parameters are presented:

- *Separation method:* Supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* -7.5dB, -5dB, -2.5dB, 0dB, 2.5dB and 5dB.
- *Number of speech components:* 20.
- *Number of music components:* 10.
- *Music styles:* classical, jazz, latin and pop rock.

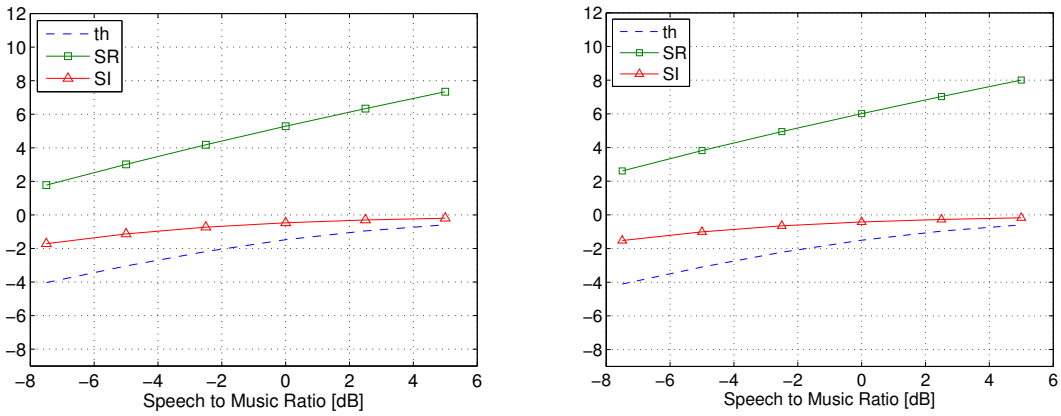
From the following graphics one can see how the quality of the separation varies depending on the music style:



((a)) Classical



((b)) Jazz



((c)) Latin ((d)) Pop rock

Figure 5.3.: Separation performance on speech corrupted by different music styles. Supervised separation using a DFT window size equal to 128ms and 10 music components.

More than 1.5dB can change the SR comparing classical and jazz music when the SMR is very low. The worst results are for jazz music, the fast variability in time of this music style makes more difficult its initialization with useful music bases. The amount of sung voice is not affecting so much the supervised separation because the music training data already contain information of the singer, and the initialized music bases are useful to approximate it and extract better the speech. In this approach, the most problematic fact is the fast variability of the music genre.

6. Semi-supervised separation using sparsity constraints

In this Chapter, sparsity constraints are added to the part of the music on $\mathbf{H}$ and/or $\mathbf{W}$. The method used is sparse semi-supervised NMF. The aim of applying sparsity constraints on the activations—$\mathbf{H}$ matrix—is to improve the discrimination of speech and music bases, for the spectral bases—$\mathbf{W}$ matrix—, is to enforces the harmonicity of music spectra.

In the following three Sections, the same experiments are done changing only the sparse matrix, $\mathbf{H}$ in Section 6.1, $\mathbf{W}$ in Section 6.2 and $\mathbf{WH}$ in Section 6.3. In the last Section 6.4, sparse semi-supervised NMF is applied for all the music styles. The experimental parameters are described below:

- *Separation method:* Sparse semi-supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* -7.5dB, -5dB, -2.5dB, 0dB, 2.5dB and 5dB.
- *Number of speech components:* 20.
- *Number of music components:* 10.
- *Sparse matrices:* $\mathbf{H}$, $\mathbf{W}$ and $\mathbf{WH}$.
- *Sparsity weights (λ and μ):* 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6}.
- *Music styles:* classical.

6.1. Sparse semi-supervised separation on the H matrix

OpenBliSSART already had the option to use sparsity on the $\mathbf{H}$ matrix, we only modified the code to apply it in the part of the music, i.e. the last 10 rows of the matrix.

To observe the effect of adding sparsity constraints, two $\mathbf{H}$ matrices were plotted previously at the end of separations with the same conditions, using sparsity constraints in 2.2(b) or without sparsity constraints in 2.2(a). One can see that for

the sparse matrix there are more dark points in the part of the music because the sparsity term penalize the non-zero values.

In Figure 6.1, one can see graphically the comparison between the non-sparse and sparse results. For evaluation, the same parameters as before are used, see 3.3. In the graphic, dotted lines represent the non-sparse results for $SMR = -2.5dB, 0dB, 2.5dB$. Solid lines represent sparse results, the thickness is changing the SMR. Depending on the weight the sparse method improves or not the results, using $\lambda = 10^{-4}$ we got the best results although the improvement is only $0.1dB$.

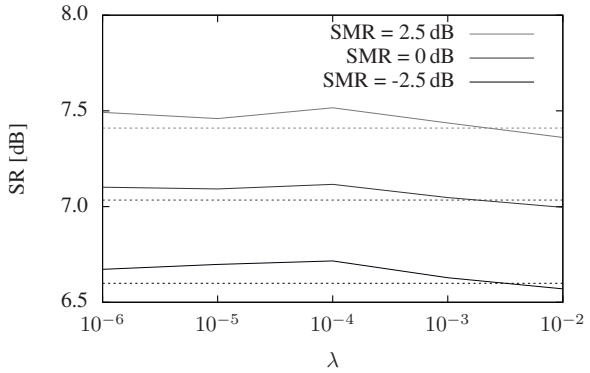


Figure 6.1.: Comparison between non-sparse and **H** sparse results for different Speech to Music Ratios while changing the sparsity weight λ.

In tables 6.1 and 6.2, the separation performance depending on the sparsity weight is shown for different Speech to Music Ratios and sparsity weights. In the first column ($\lambda = 0$) there are results without adding the sparsity term.

Table 6.1.: SR_{speech} results applying sparsity on the unsupervised part of the **H** matrix.

SMR [dB]	$\lambda = 0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
-7.5	5.543	5.626	5.501	5.592	**5.633**	5.595	5.608
-5	6.128	6.203	6.089	6.136	**6.213**	6.186	6.195
-2.5	6.599	6.683	6.570	6.628	**6.716**	6.698	6.672
0	7.034	**7.123**	6.996	7.047	7.116	7.092	7.101
2.5	7.410	7.496	7.361	7.437	**7.516**	7.460	7.492
5	7.772	7.841	7.737	7.788	**7.863**	7.835	7.831

Numbers in boldface are the highest result in each row. In Table 6.1, with $\lambda = 10^{-4}$ we achieve these results except for $SMR = 0dB$. In terms of similarity, Table 6.2 shows that $\lambda = 10^{-4}$ also gives the best results for all the cases.

In conclusion, adding sparsity constraints to the musical part of the **H** matrix is improving the separation performance for classical music using $\lambda = 10^{-4}$.

Table 6.2.: SI_{speech} results applying sparsity on the unsupervised part of the **H** matrix.

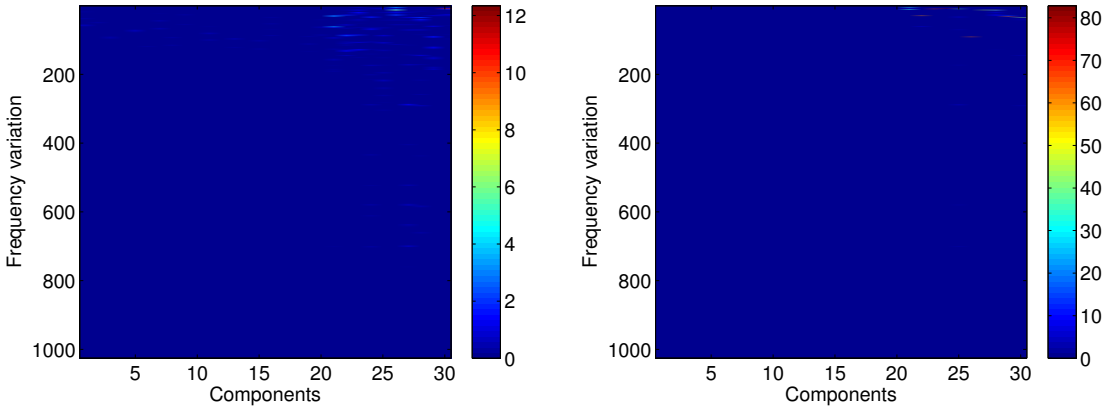
SMR [dB]	$\lambda = 0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
-7.5	-1.095	-1.087	-1.109	-1.091	**-1.080**	-1.090	-1.086
-5	-0.977	-0.972	-0.979	-0.979	**-0.964**	-0.967	-0.967
-2.5	-0.932	-0.926	-0.933	-0.927	**-0.912**	-0.915	-0.920
0	-0.931	-0.922	-0.930	-0.928	**-0.916**	-0.920	-0.922
2.5	-0.957	-0.952	-0.961	-0.955	**-0.942**	-0.954	-0.942
5	-1.001	-0.999	-0.999	-0.996	**-0.991**	-0.992	-0.992

6.2. Sparse semi-supervised separation on the W matrix

We added the option of applying sparsity constraints on **W** with *OpenBliSSART*. The reason to apply sparsity constraints to the music bases is to increase the discrimination between the input sources, as music is arguably characterized by higher harmonicity compared to speech. Thus, we are using sparsity only in the part of the music, i.e. the last 10 columns of the matrix.

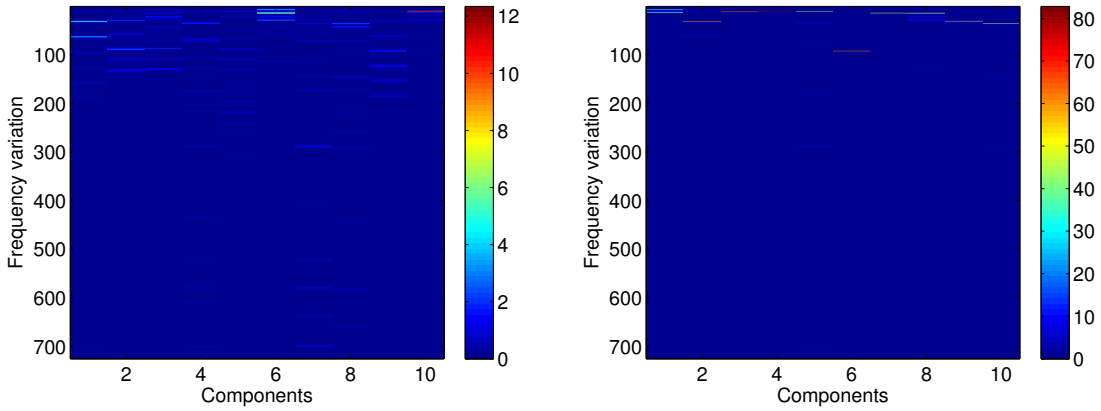
In order to observe the effect of adding sparsity constraints to $\mathbf{W}^{(m)}$, in Figures 6.2(a) and 6.2(b), two **W** matrices are plotted at the end of separations with the same conditions, but in one of them, sparsity constraints were added in the last 10 columns of the matrix. There is a big difference between speech and music bases: on the one hand, speech bases are the result to apply NMF to training data and they are not changing during the separation; on the other hand, music bases are changing iteratively during the matrix decomposition to approximate the input. *OpenBliSSART* has an option for normalization the matrices, using it for the **W** matrix has no effect to the separation results.

Figures 6.2(c) and 6.2(d) are a zoom corresponding to the low frequencies of last 10 columns of each matrix.



((a)) Without sparsity

((b)) With **W** sparsity



((c)) Without sparsity

((d)) With **W** sparsity

Figure 6.2.: Comparison between the **W** matrix while applying or not sparsity in the music part. $SMR = -2,5dB$, $\mu = 10^{-5}$ for ((b)) and ((d)) cases. The images below represents the lower frequencies of the last 10 columns of **W** corresponding to the music bases.

In Figure 6.3, one can see graphically the comparison between the non-sparse and sparse results. In the graphic, dotted lines represent the non-sparse results for $SMR = -2.5dB, 0dB, 2.5dB$. Solid lines represent sparse results, the thickness is changing the SMR. For $\mu = 10^{-5}$ the best results are achieved, even better than the results of adding sparsity constraints to the **H** matrix.

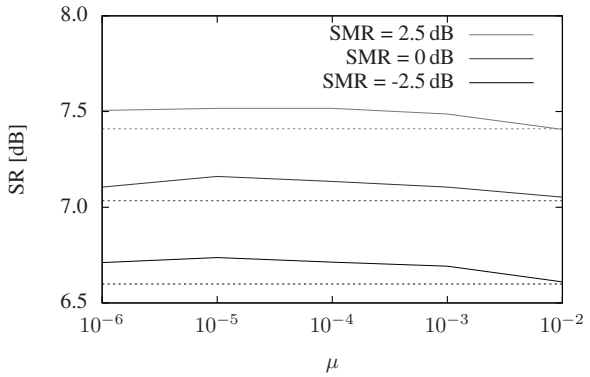


Figure 6.3.: Comparison between non-sparse and **W** sparse results for different Speech to Music Ratios while changing the sparsity weight μ.

In Tables 6.3 and 6.4 the separation performance depending on the sparsity weight is presented. As seen in the previous figure, if a good suppression of the music is required, $\mu = 10^{-5}$ (biggest SR) is the best choice, but whether any loss in speech information is permitted, $\mu = 10^{-1}$ (biggest SI) is the best option. Applying sparsity on the **W** matrix improves the SI for all the cases, but only some μ values improve the SR.

Table 6.3.: SR_{speech} results applying sparsity on the unsupervised part of the **W** matrix.

SMR [dB]	$\mu = 0$	$\mu = 10^{-1}$	$\mu = 10^{-2}$	$\mu = 10^{-3}$	$\mu = 10^{-4}$	$\mu = 10^{-5}$	$\mu = 10^{-6}$
-7.5	5.543	5.509	5.572	5.604	5.650	**5.660**	5.629
-5	6.128	6.068	6.149	6.190	6.220	**6.259**	6.194
-2.5	6.599	6.564	6.610	6.692	6.713	**6.737**	6.711
0	7.034	6.982	7.053	7.105	7.135	**7.161**	7.105
2.5	7.410	7.325	7.407	7.487	7.517	**7.517**	7.506
5	7.772	7.679	7.760	7.832	7.868	**7.905**	7.835

6.3. Sparse semi-supervised separation on the H and W matrices

To execute this experiment we had to modify the multiplicative update rules in the part of the music of both matrices: **W** and **H**, i.e. the last 10 columns of **W** and the last 10 rows of **H**.

In Tables 6.5, 6.6 the separation performance depending on the sparsity weight is presented. If a good suppression of the music is required, $\lambda = \mu = 10^{-4}$ or $\lambda = \mu = 10^{-6}$ (biggest SR) is the best choice, but whether any loss in speech

Table 6.4.: SI_{speech} results applying sparsity on the unsupervised part of the **W** matrix.

SMR [dB]	$\mu = 0$	$\mu = 10^{-1}$	$\mu = 10^{-2}$	$\mu = 10^{-3}$	$\mu = 10^{-4}$	$\mu = 10^{-5}$	$\mu = 10^{-6}$
-7.5	-1.095	**-1.073**	-1.086	-1.080	-1.075	-1.089	-1.086
-5	-0.977	**-0.951**	-0.965	-0.961	-0.965	-0.966	-0.968
-2.5	-0.932	**-0.892**	-0.927	-0.908	-0.916	-0.923	-0.914
0	-0.931	**-0.884**	-0.918	-0.910	-0.914	-0.919	-0.923
2.5	-0.957	**-0.917**	-0.954	-0.938	-0.941	-0.952	-0.946
5	-1.001	**-0.951**	-0.998	-0.986	-0.986	-0.989	-0.995

information is permitted, $\lambda = \mu = 10^{-1}$ (biggest SI) is the best option. Depending on the sparsity weight, this method is not improving the separation and its results are not so good as the results for **W** sparse.

Table 6.5.: SR_{speech} results applying sparsity on the unsupervised part of **H** and **W**. Note that in this experiment $\lambda = \mu$.

SMR [dB]	$\lambda = 0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
-7.5	5.543	5.415	5.575	5.479	5.596	5.574	**5.606**
-5	6.128	5.977	6.163	6.098	6.187	6.155	**6.206**
-2.5	6.599	6.476	6.661	6.573	**6.683**	6.668	6.677
0	7.034	6.896	7.078	7.008	7.078	7.059	**7.095**
2.5	7.410	7.240	7.463	7.399	7.483	7.433	**7.489**
5	7.772	7.573	7.791	7.738	**7.832**	7.786	7.810

6.4. Evaluating the influence of the music style

In this Section, the configuration which gave the best results using sparsity constraints, is used to evaluate the influence of the music style. These results were achieved with $\mu = 10^{-5}$, i.e. **W** sparse. The experimental parameters are mentioned below:

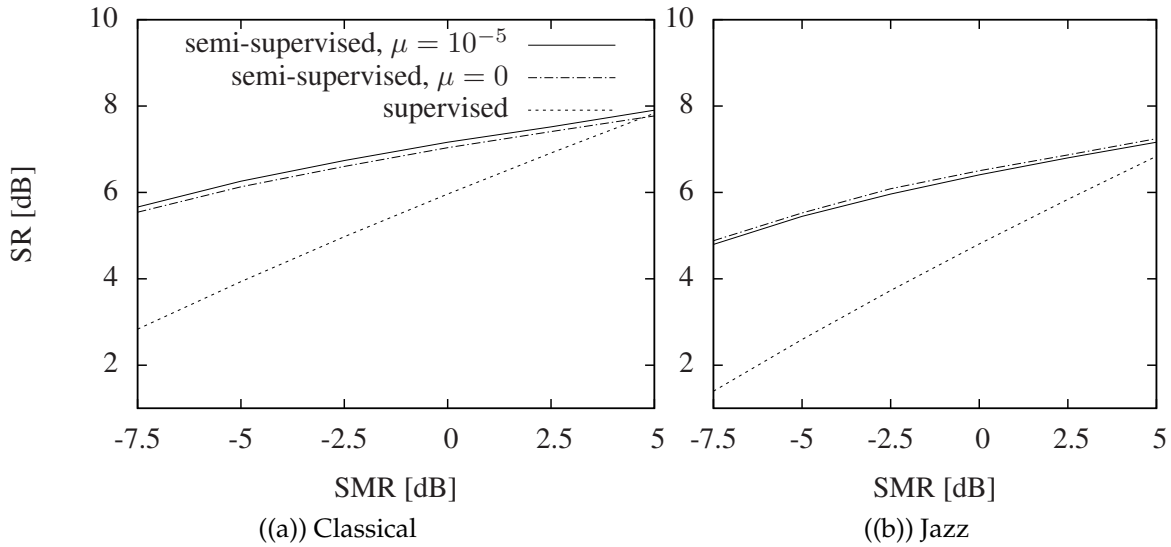
- *Separation method:* Sparse semi-supervised NMF.
- *DFT window size:* 128ms.
- *Speech to Music Ratio:* -7.5dB, -5dB, -2.5dB, 0dB, 2.5dB and 5dB.
- *Number of speech components:* 20.

Table 6.6.: SI_{speech} results applying sparsity on the unsupervised part of $\mathbf{H}$ and $\mathbf{W}$. Note that in this experiment $\lambda = \mu$.

SMR [dB]	$\lambda = 0$	$\lambda = 10^{-1}$	$\lambda = 10^{-2}$	$\lambda = 10^{-3}$	$\lambda = 10^{-4}$	$\lambda = 10^{-5}$	$\lambda = 10^{-6}$
-7.5	-1.095	**-1.077**	-1.085	-1.116	-1.087	-1.095	-1.092
-5	-0.977	**-0.954**	-0.963	-0.984	-0.962	-0.972	-0.968
-2.5	-0.932	**-0.895**	-0.911	-0.933	-0.918	-0.920	-0.925
0	-0.931	**-0.889**	-0.909	-0.925	-0.925	-0.926	-0.919
2.5	-0.957	**-0.920**	-0.939	-0.953	-0.947	-0.953	-0.952
5	-1.001	**-0.962**	-0.985	-0.997	-0.991	-1.000	-0.992

- *Number of music components:* 10.
- *Sparse matrices:* $\mathbf{W}$.
- *Sparsity weights (μ):* 10^{-5}.
- *Music styles:* classical, jazz, latin, pop rock.

The following graphics show the separation results:



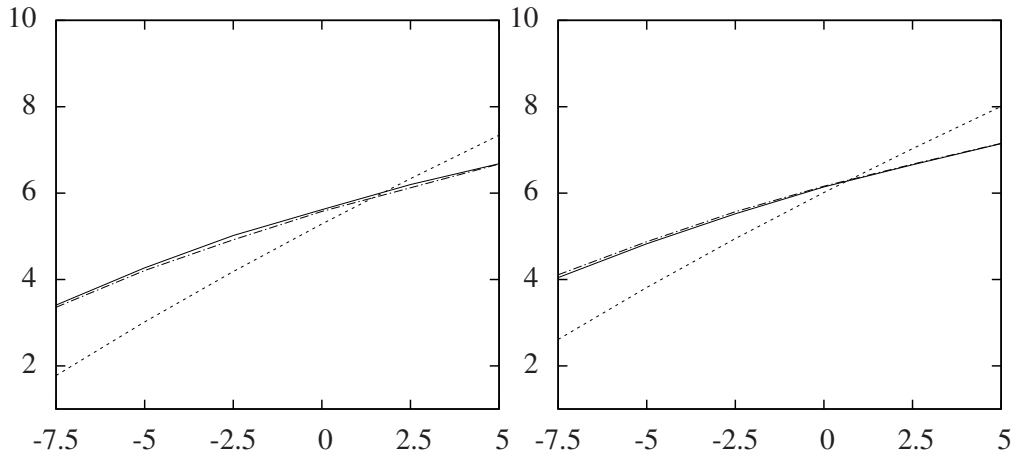


Figure 6.4.: Comparison between the separation performance using different methods and changing the music style. The methods used are: supervised NMF, semi-supervised NMF and sparse semi-supervised NMF (with $\mu = 10^{-5}$).

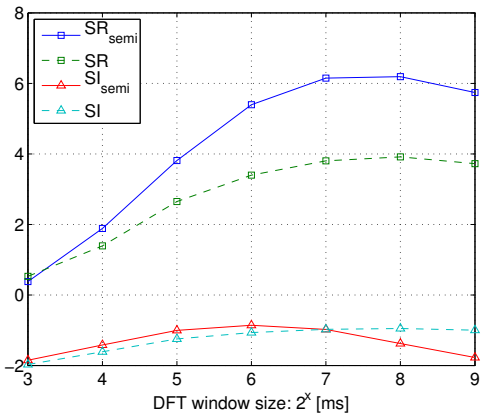
Figure 6.4 shows how sparsity constraints improve a little the separation for classical and latin music but not for jazz and pop rock. The reason is that classical and latin music have stronger harmonic characteristics and, is in these cases, when the addition of sparsity constraints to the $\mathbf{W}$ matrix can improve the discrimination between speech and music bases.

7. Comparison between supervised and semi-supervised method and specific examples

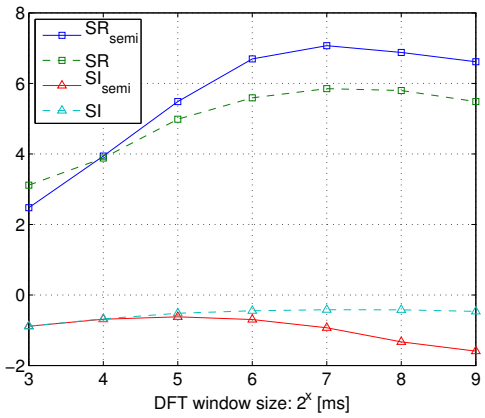
In this Chapter, the results of Chapters 4 and 5 are compared. In Section 7.2, specific examples of separations are evaluated in order to see the influence of the sung voice in the separation results.

7.1. Comparison of supervised and semi-supervised method

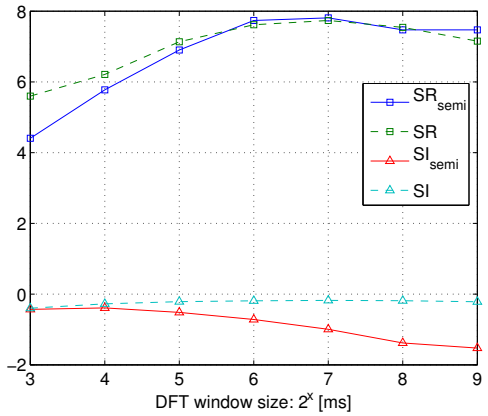
In this Section, both methods are compared using the following graphics. Each plot contains the source ratio and the similarity for both approaches, supervised and semi-supervised:



((a)) $SMR = -5dB$



((b)) $SMR = 0dB$



((c)) $SMR = 5dB$

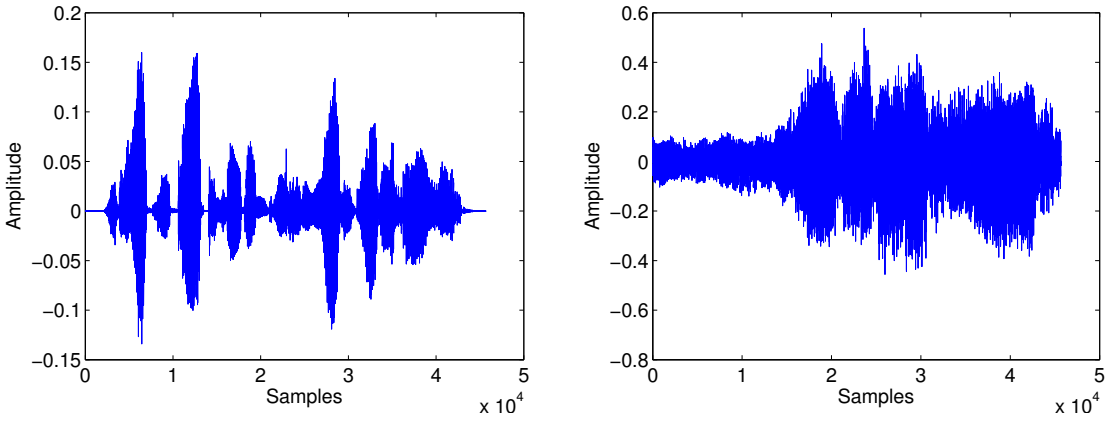
Figure 7.1.: Comparison of supervised and semi-supervised methods while changing the DFT window size and keeping constant the SMR (*semi*: semi-supervised method, the other one is the supervised method).

From the previous graphs, one can see that—in terms of Source Ratio—the DFT window size that gives better separation performance is $128ms$, these results are consistent with others from similar experiments [24]. Interestingly, for $SMR < 5dB$, using 10 music components, the separation done using the semi-supervised method is better than the supervised one, the most significant difference is in the figure ((a)). Comparing the SR_{semi} (semi-supervised) to the SR (supervised), when the DFT windows size is equal to 128ms, the difference is bigger than $2dB$. If we increase the $SMRs$, the SR_{semi} starts to resembles the SR; and the SI_{semi} loses quality over SI.

7.2. Specific examples: the effect of voice

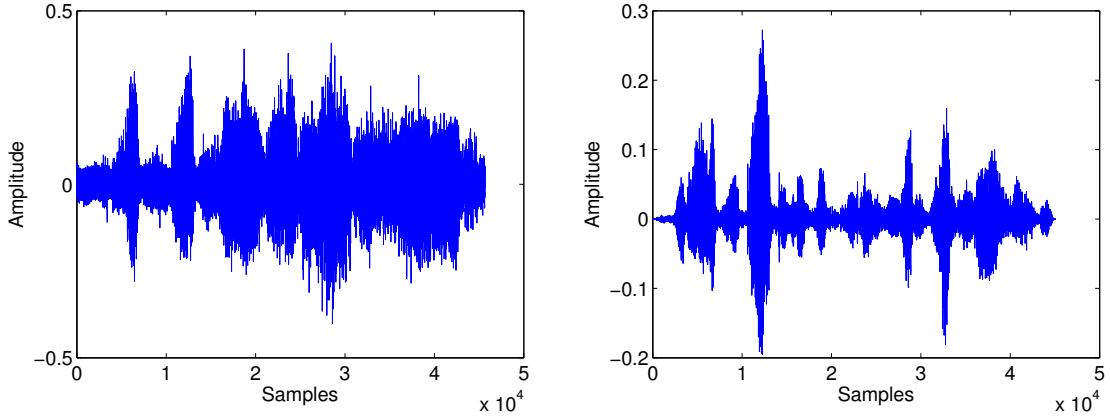
All the previous plots during the thesis are the result to apply a mean for all the 1680 separated sentences. If instead of doing this, we look at specific examples, we realize that when the music part of the mixed file contains sung voice the separation is not so good. It was predictable since the sung voice has similar characteristics to the speech. Following, two examples of separations using the semi-supervised method are presented to observe the sung voice problem.

Firstly, four graphics are plotted representing time signals of the original sources, the mixture and separated speech. Note that in this case the music cut not contains sung voice.



((a)) Original speech

((b)) Original music

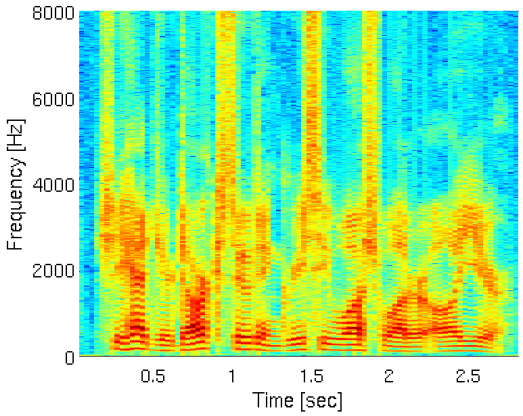


((c)) Mixed signal

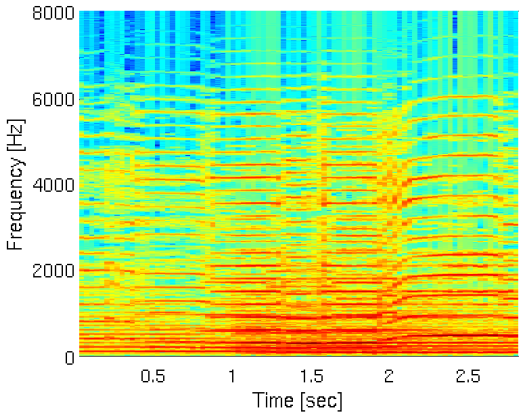
((d)) Separated speech

Figure 7.2.: Representation of different signals in time. The separation is done using the semi-supervised method, the music file has no voice, $SMR = -5dB$ and $DFT = 128ms$.

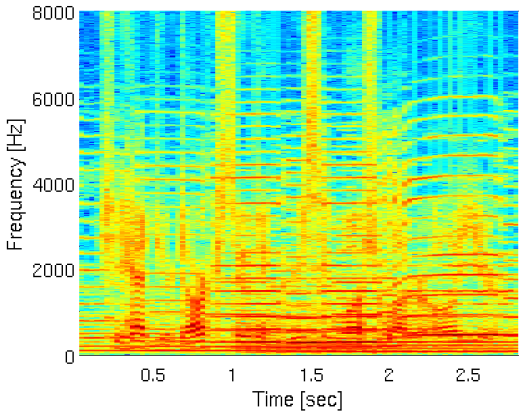
Comparing original and separated speech signals, we can see that some speech information is lost during the separation but almost all the music is suppressed. To evaluate better the extraction we will plot the previous signals in the frequency domain:



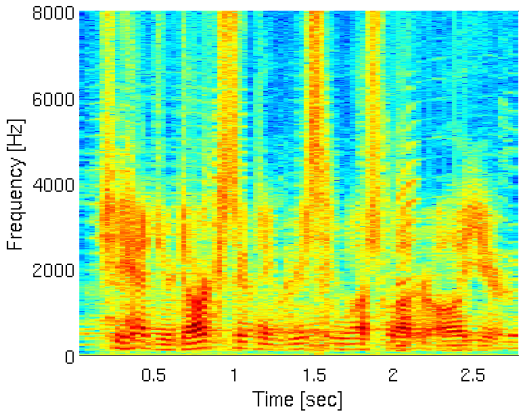
((a)) Original speech



((b)) Original music



((c)) Mixed signal

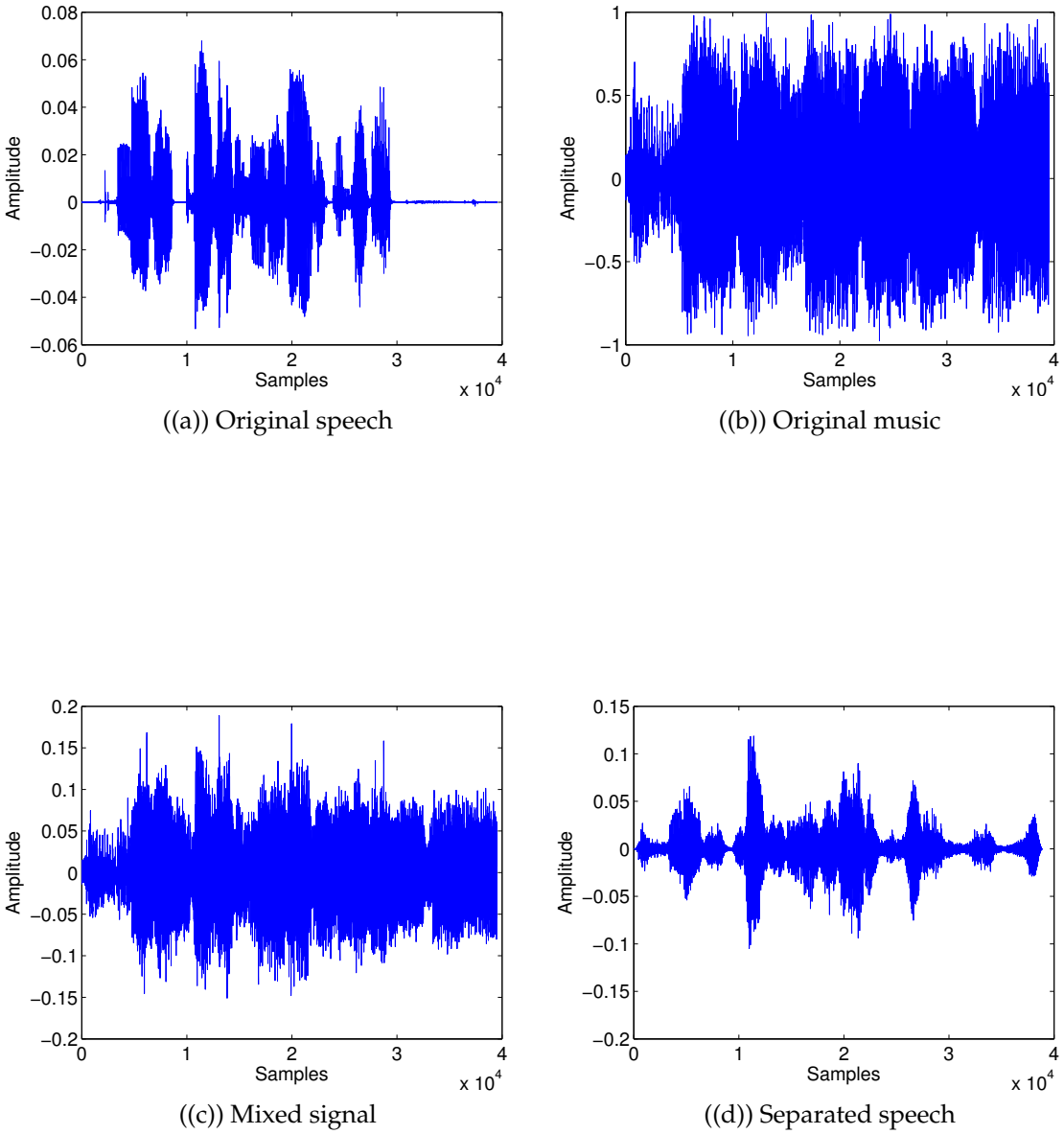


((d)) Separated speech

Figure 7.3.: Representation of different spectras. The separation is done using the semi-supervised method, the music file has no voice, $SMR = -5dB$ and $DFT = 128ms$.

In the frequency domain, original and separated speech signals are very similar, the only part of the music signal not suppressed in the separated signal is around $Time = 2s$, where an strong music object is located at the same instant than a speech object and the system can not discriminate the different sources.

The following figures represent a separation where the part of the music contains sung voice. As before, first of all in the time domain and later in frequency domain.



((a)) Original speech

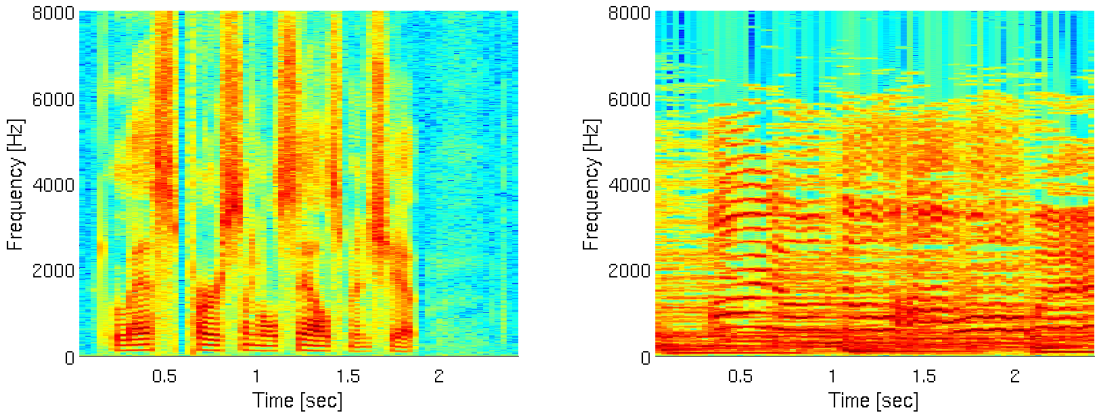
((b)) Original music

((c)) Mixed signal

((d)) Separated speech

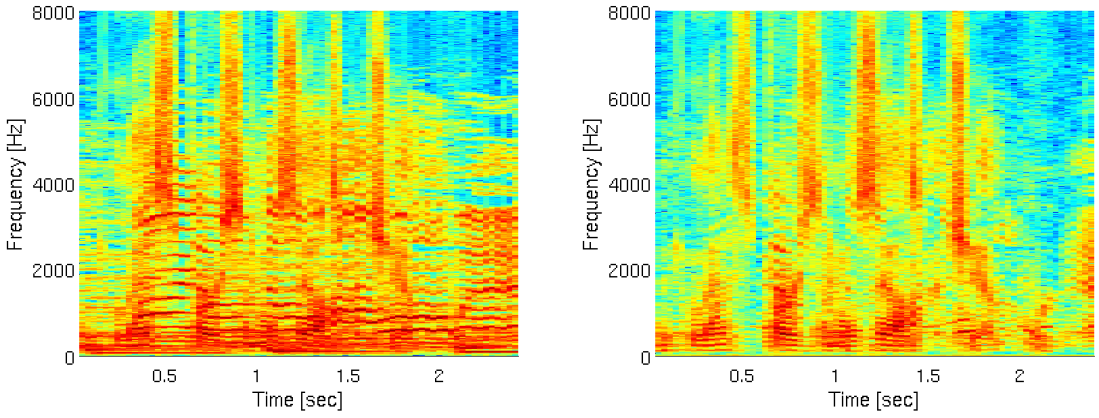
Figure 7.4.: Representation of different signals in time. The separation is done using the semi-supervised method, the music file contains voice, $SMR = -5dB$ and $DFT = 128ms$.

In this case, the music suppression is more difficult. Comparing original and separated speech signals we can see that, at the end of the separated signal, part of the undesired source appears corresponding to the voice of the singer. To evaluate better the extraction one can see the previous signals in the frequency domain:



((a)) Original speech

((b)) Original music



((c)) Mixed signal

((d)) Separated speech

Figure 7.5.: Representation of different spectras. The separation is done using the semi-supervised method, the music file contains voice, $SMR = -5dB$ and $DFT = 128ms$.

At the end of the separated speech spectrogram we can observe the last part of the music spectrogram. Although the NMF approach removes the major part of the music, it is not capable to eliminate the sung voice completely.

One of the reasons of the addition of sparsity constraints to the semi-supervised NMF approach was to improve the separation in these cases, when there is sung voice in music —almost all the music songs—. As sung voice has stronger harmonic character than speech, the addition of sparsity constraints to the $\mathbf{W}^{(m)}$ matrix can help the separation as was proved in Chapter 6.

8. Conclusions

We introduced a novel semi-supervised NMF algorithm for compensation of background music in speech and we demonstrated that it results in large source ratio improvements particularly in highly noisy environments. Comparing the semi-supervised method to an upper benchmark for supervised NMF assuming the characteristics of the music are known, it is highly notable that in many cases the semi-supervised method suppresses the music to a larger extent than the supervised benchmark, in terms of source ratio, while the similarity is almost the same for both methods or slightly worse for the semi-supervised approach. Finally, we could demonstrate that the performance of semi-supervised NMF could be improved both by enforcing sparsity constraints on the spectra and on the activations.

The most important parameters of NMF are evaluated for both, supervised and semi-supervised methods, concluding that a DFT window size equal to 128ms gives the best separation performance. 20 speech components are used for all experiments and the number of music components chosen is 10, which gives the best combination of source ratio and similarity semi-supervised results.

All the separation methods are evaluated with four different music styles: classical, jazz, latin and pop rock. The best results are achieved for classical music and the worst for latin. Finally, the addition of sparsity constraints to the part of the music on the spectra and on the activations improves the separation performance for classical and latin music, contrary to jazz and pop rock for which the non-sparse semi-supervised NMF is the best method.

If the separations are individually evaluated, one can see that the emergence of sung voice in the music file is making more difficult the music suppression during the separation. The fast variability in time and frequency, and the great deal of sung voice in latin music makes this music genre the worst for speech separation.

We believe that semi-supervised NMF approaches offer promising results in compensation of background music in speech, so do ASR with the separation results would be the next step to continue with our approach, since apply ASR is the best way to evaluate the separation performance. Furthermore, application of our approaches to realistic cases such as comprising speech enhancement for in-car human-machine interfaces or mobile telephony in highly noisy environments such as discotheques, speech recognition for multimedia information retrieval in TV series or on-line videos, or even lyrics transcription of rap/hip-hop music will be very interesting.

Appendix

Bibliography

[1] *Non-negative sparse coding*, 2002.

[2] S. Boll. Suppression of acoustic noise in speech using spectral subtraction. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 27(2):113 – 120, apr 1979.

[3] A. Cichocki, R. Zdunek, and S. Amari. New algorithms for non-negative matrix factorization in applications to blind source separation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, page V, may 2006.

[4] O. Dikmen and A.T. Cemgil. Unsupervised single-channel source separation using bayesian nmf. In *Applications of Signal Processing to Audio and Acoustics, 2009. WASPAA '09. IEEE Workshop on*, pages 93 –96, oct. 2009.

[5] J.-L. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):564–575, 2010.

[6] J. Eggert and E. Korner. Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529 – 2533 vol.4, july 2004.

[7] Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 33(2):443 – 445, apr 1985.

[8] M. Helen and T. Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. of EUSIPCO*, Antalya, Turkey, 2005.

[9] R. Hennequin, R. Badeau, and B. David. Nmf with time-frequency activations to model non stationary audio events. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 445 –448, march 2010.

[10] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints, August 2004.

[11] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94 – 128, 1999.

[12] J.E. Jackson. A user's guide to principal components. *Wiley-Interscience paperback series*.

[13] Y. Kitano, H. Kameoka, Y. Izumi, N. Ono, and S. Sagayama. A sparse component model of source signals and its application to blind source separation. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 4122 –4125, march 2010.

[14] B. De Moor L. De Lathauwer and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, apr 2000.

[15] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.

[16] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of NIPS*, pages 556–562, Vancouver, Canada, 2001.

[17] T. Nakatani and S. Araki. Single channel source separation based on sparse source observation model with harmonic constraint. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 13 –16, march 2010.

[18] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh. Non-negative matrix factorization based compensation of music for automatic speech recognition. In *Proc. of Interspeech*, Makuhari, Japan, 2010.

[19] M. N. Schmidt and R. K. Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Proc. of Interspeech*, Pittsburgh, PA, USA, 2006.

[20] B. Schuller, B. J. Brüning-Schmitt, D. Arsić, S. Reiter, M. Lang, and G. Rigoll. Feature selection and stacking for robust discrimination of speech, monophonic singing, and polyphonic music. In *Proc. ICME*, pages 840–843, Amsterdam, The Netherlands, 2005. IEEE.

[21] B. Schuller, F. Eyben, and G. Rigoll. Tango or Waltz?—Putting Ballroom Dance Style into Tempo Detection. *EURASIP Journal on Audio, Speech, and Music Processing (JASMP), Special Issue on "Intelligent Audio, Speech, and Music Processing Applications"*, 2008. Article ID 846135, 12 pages.

[22] B. W. Schuller, A. Lehmann, F. J. Weninger, F. Eyben, and G. Rigoll. Blind enhancement of the rhythmic and harmonic sections by NMF: Does it help? In

Proc. of the International Conference on Acoustics (NAG/DAGA 2009), Rotterdam, The Netherlands, 2009.

[23] P. Smaragdis. Discovering auditory objects through non-negativity constraints. In *Proc. of SAPA*, Jeju, Korea, 2004.

[24] P. Smaragdis. Convolutive speech bases and their application to supervised speech separation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(1):1–14, 2007.

[25] P. Smaragdis and J.C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177 – 180, oct. 2003.

[26] S.K. Tjoa and K.J.R. Liu. Multiplicative update rules for nonnegative matrix factorization with co-occurrence constraints. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 449 –452, march 2010.

[27] T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1066–1074, March 2007.

[28] Felix Weninger, Alexander Lehmann, and Björn Schuller. openblissart: Design and evaluation of a research toolkit for blind source separation in audio recognition tasks. In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, 2011.

[29] T. Yu and J. H. L Hansen. Automatic beamforming for blind extraction of speech from music environment using variance of spectral flux-inspired criterion. *IEEE Journal of Selected Topics in Signal Processing, Issue on "Speech Processing for Natural Interaction With Intelligent Environments"*, 4(5):785–797, 2010.

[30] Tao Yu and J.H.L. Hansen. An efficient microphone array based voice activity detector for driver's speech in noise and music rich in-vehicle environments. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2834 –2837, march 2010.