Nonlinear Audio Recurrence Analysis with Application to Music Genre Classification.

Carlos A. de los Santos Guadarrama

MASTER THESIS UPF / 2010
Master in Sound and Music Computing

Master thesis supervisors:

Joan Serrà and Ralph G. Andrzejak.

Department of Information and Communication Technologies

Universitat Pompeu Fabra, Barcelona.

Nonlinear Audio Recurrence Analysis with Application to Music Genre Classification.

Master Thesis, Master in Sound and Music Computing.

Carlos A. de los Santos Guadarrama.

carlos.dlsg@gmail.com

Department of Information and Communication Technologies,

Music Technology Group.

Universitat Pompeu Fabra.

Barcelona, Spain.

Abstract

Audio classification is a Music Information Retrieval (MIR) area of interest, dedicated to extract key features from music by means of automatic implementations. On this research, nonlinear time series analysis techniques are used for the processing of audio waveforms. The use of nonlinear time series analysis in audio classification tasks is relatively new. These techniques are implemented with the assumption that the temporal evolution of audio signals can be analyzed over a multidmensional space, with the intention of finding additional information that usual audio analysis tools, such as the Fourier Transform, might not bring. In particular, iterative or recurrent patterns in audio signals over a multidimensional space is the desired additional information to find. Some first evidence show these tools can be sensitive to audio signal analysis.

In this thesis, two complementary sources for feature extraction based on nonlinear time series analysis are presented. The process consists in performing a recurrence analysis over framed audio signals and representing the output in two different formats: the first, a histogram of the found recurrences at different times in the audio frame. The second, a frequency histogram obtained by transforming and fitting the recurrence time histogram into frequency values with the same resolution as the correspondent frequency spectrum. A specific set of spectral features are then extracted from both representations and used for classifier training and testing.

The reliability of new data obtained through these sources is tested by comparing to a common automatic classification methodology, choosing music genre as the target of classification. Among other results described, the combination of features extracted from the Fourier frequency spectrum and features extracted from histograms resulted in a 5.5% increment in the highest common classification accuracy, raising it from 66.0% using common methodology to 71.5%. Moreover, the creation of new specific features for these histograms and the maximization of parameters used to perform the nonlinear analysis is suggested as future work on this research.

Acknowledgements

I would primarily like to thank my tutors, Joan Serrà and Ralph Andrzejak, for their support, time, and patience in the development of this research. Without their help and guidance, this thesis would not have been accomplished. I would also like to thank Xavier Serra for the counseling and for having the trust in me to become part of the Music Technology Group. A special acknowledgement to George Tzanetakis for providing the audio database for the analysis done on this research. My gratitude goes also to all my colleagues at the MTG and to all the very special people that I have met during this year, for their cheering, for being by my side and never letting go.

This thesis is specially dedicated to my Parents. My father, for being the captain, for steering the wheel, and for being the greatest support ever on every step I take; and my mother, for always being there, for caring, and for telling me that if goals were easy, anyone would accomplish them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Music is one of the most popular elements of the Internet. There are uncountable online services dedicated to downloading, live-streaming, sharing or creating this type of content. Given the increasing amount of information related to online music databases over the past years, a new challenge in searching, retrieving and organizing music content is arising. On the present day, there are two different approaches confronting these tasks: the first is manual labeling, which relies on cultural and musical knowledge about performers, instrumentation, tonality and genre, to mention a few. The second is automatic classification, consisting in extraction of audio features related to the music signal and its adaptation to predict a label. Since manually labeling millions of songs on a given database can be temporally unfeasible, automatic classification systems are receiving much attention in the musical community, at the point of developing a relatively new research field called Music Information Retrieval [9]. This field is dedicated to the development of signal processing techniques, music perception models and audio files cataloging, among others, in order to achieve tasks such as artist recognition, audio fingerprinting, genre classification, music recomendation, cover song detection and many more [17].

An emerging MIR practice is related to the application of nonlinear time series analysis methods to obtain supplementary information about the audio signal. There is evidence that this type of analysis is susceptible to audio signals in a

constructive way, meaning that reliable information can be obtained through these methods [5].

The motivation for this thesis is to contribute with two additional sources of information for automatic classification systems based on nonlinear analysis tools, refered to as *Recurrence Histogram* and *Frequency Histogram*. The reliability of new data will be tested by comparing to a common automatic classification methodology, choosing music genre as the target of classification.

1.1 Goals

The goals of this research are the following:

1. Develop a genre classification system based on temporal and spectral features extraction, using common methods of analysis.

2. Develop a nonlinear analysis module for audio features extraction, based on four specific techniques:

   - State-space embedding.
   - Recurrence plot analysis.
   - Recurrence time histogram (H_t).
   - Recurrence frequency histogram (H_f).

3. Test classification accuracy relying on music genre as the target of classification, using different combinations of features obtained from the histograms and features extracted from the classic methodology.

4. Compare the new accuracy results with the accuracy obtained through common classification methodology.

5. Conclude about the influence that new information from the nonlinear analysis has on the classification accuracy.

1.2 Structure of the thesis

The remainder of this document is organized as follows: chapter 2 reviews the state of the art and basic principles in music classification tasks. Starting with a brief definition of audio signals, it goes through different types of features usually extracted from the frequency spectrum. In addition, an introduction to Music Information Retrieval is given, explaining how it is related to classification and generation of automatic classification tasks. Finally, a review of nonlinear time series analysis is given, showing how these techniques have been used in other works as well. Chapter 3 describes the common classification methodology used in this thesis. Extracted features, applied tools for audio analysis, and feature selection processes are described in this chapter as well. On chapter 4 the nonlinear audio recurrence analysis is explained, starting with a description of the state-space reconstruction, the recurrence analysis of the resulting trajectory, and how this information is translated into the final sources of information for feature extraction: the recurrence time histogram and the recurrence frequency histogram. Chapter 5 shows the accuracy results for several classifications, using different feature combinations extracted from the common methodology and from the nonlinear audio recurrence analysis. It also shows the changes in classification accuracies caused by modifying the parameters of the nonlinear analysis tools. Finally, chapter 6 states the conclusions about this research and suggests extensions of the nonlinear audio recurrence analysis to be done in the future.

Chapter 2

State of the Art

This chapter is a description of the basic principles used for the elaboration of this thesis. It covers basic audio signal analysis, feature extraction for music information retrieval, and an introduction to the nonlinear time series analysis used on audio signals.

2.1 Overview of Digital Signals

2.1.1 The Audio Signal

An audio signal is an electrical representation of the acoustical energy produced by sound. This type of energy is caused by continuous-time pressure variations on a physical medium, usually air. Therefore, an audio signal is a continuous-time (CT) signal, defined on a continuum of points over time [4].

2.1.2 Digital Representation of Signals

Nowadays, most audio signal processing and analysis is done using computers, microcontrollers, and other programmable devices based on digital circuitry. Since digital processing requires the information to be presented as a numerical time series, digital equivalents must be created from the information given by original CT

signals [21].

The digital signal representation of a CT signal is achieved by analog-to-digital conversion (ADC). ADC systems perform sampling and quantization of the CT signal. Sampling means capturing the values of a CT signal at discrete points in time. A common practice is to define a sampling frequency (f_s) to obtain values from the signal at a fixed time rate. This type of signals are refered to as discrete-time (DT) signals [21]. On the other hand, quantization means adjusting the amplitude values of the DT signal to fixed values called *levels*. These quantization levels will range from -2^{n-1} to $2^{n-1} - 1$ where n is the number of quantization bits. Usually, this range is normalized between -1 and 1 [36]. Common quantization values are 16 and 24 bits. An example of a CT signal and its equivalent digital signal can be seen on figure 2.1.

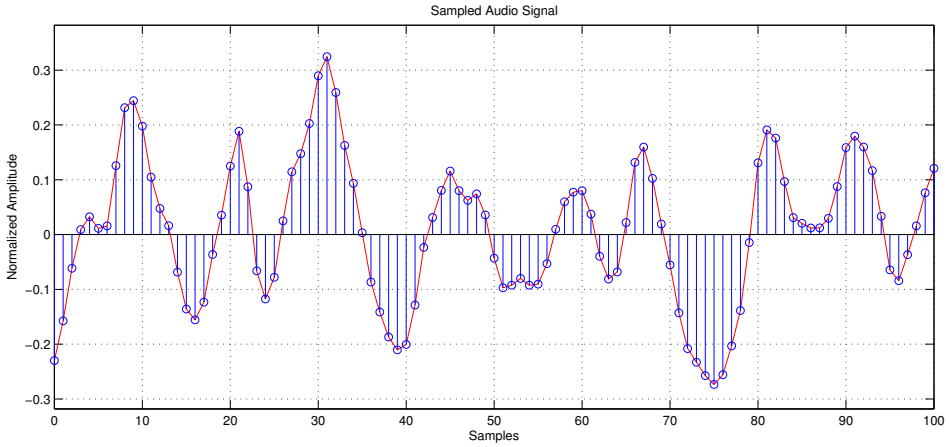


Figure 2.1: CT signal (red) and its equivalent digital signal (blue).

2.1.3 The Sampling Theorem

The Nyquist frequency f_n is the highest frequency present on a defined CT signal. The sampling theorem states that, if a CT signal is sampled with f_s twice the value of the Nyquist frequencyf_n or more, the original CT can be reconstructed from its samples. By having frequency content above f_n, a phenomenon known as *aliasing* takes place, where frequencies higher than $\frac{f_s}{2}$ are reconstructed with lower frequency

values [35].

Considering the human audible spectrum from 20 to 20,000 Hz, the minimum f_s for audio signals is 40,000 Hz. Nevertheless, a f_s value of 20,000 Hz is also valid for musical audio signals. Traditional music instruments produce defined sounds called notes. Each note is characterized by having a fundamental frequency that is perceived by the human ear as pitch. These fundamental frequencies, for traditional instruments, are below 10,000 Hz[1]. Professional audio studios sample at 96,000 Hz but downsample to 22,050 Hz or 44,100 Hz when transfering to CD or MP3 formats.

2.2 Time to Frequency Transformation

2.2.1 The Frequency Spectrum

The spectrum of a signal is a representation of its energy distribution across the frequency range. The spectrum of a digital signal can be computed by the Discrete Fourier Transform (DFT) [21]. For N consecutive samples taken from a digital signal $x(n)$, the DFT $X(k)$ is calculated by:

$$X(k) = F\{x(n)\} = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{nk}{N}} \tag{2.1}$$

where k is the number of frequency bins and goes from $0, ..., N-1$. The frequency value for each bin is obtained by:

$$f(k) = \frac{k}{N} f_s \tag{2.2}$$

For real-valued signals, the sampling operation leads to repetitions of the spectrum of the CT signal, as can be seen on figure 2.2. The original spectrum from the CT signal goes from bin 0 to bin $\frac{N}{2} - 1$[36]. The remaining part, which is a replicated

[1] Independent Recording Network. Interactive frequency chart, 2006. http://www.independentrecording.net/irn/resources/freqchart/main_display.htm

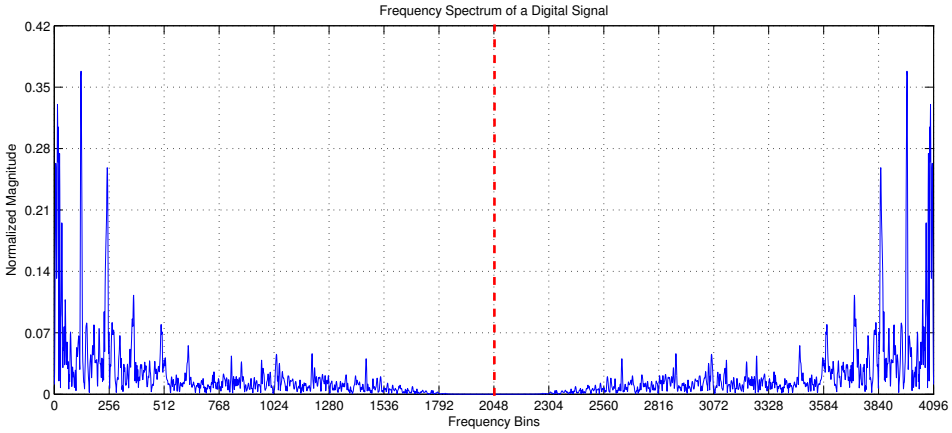


Figure 2.2: Frequency spectrum of a digital signal using $N = 4096$. The original spectrum is below the red line, representing the bin where the Nyquist Frequency is located.

reflection of the original spectrum, can be left out of the analysis for the purposes of this thesis.

The Fast Fourier Transform (FFT) is the computational algorithm that calculates the DFT on power-of-two values of N [36]. It is widely used on digital signal processing applications such as filtering, voice processing, and audio synthesis among others [21].

2.2.2 The Short-Time Fourier Transform

In practice, long digital signals such as recorded songs or audio tracks are processed in small sections or frames, not only because it is more significant to the analysis of its temporal evolution, but because it is computationally faster. A common way to obtain the DFT locally on consecutive frames of a digital signal is by the Short-Time Fourier Transform (STFT). The STFT is defined as:

$$X_l(k) = \sum_{n=0}^{N-1} w(n)x(n + lH)e^{-j2\pi\frac{nk}{N}} \tag{2.3}$$

Where $X_l(k)$ is the DFT of frame l, $w(n)$ is a window function of length N, and H is the hop-size or number of samples the frame advances on $x(n)$ [26].

The window function smoothens the spectrum by itself, but it can also modify the frequency resolution by increasing its length to the next power of two. By doing so, the missing values can be filled with zeros without affecting the outcome and increasing the values of k, which translates intro a frequency resolution increment. This technique is known as *zero padding*, and it is used to increase frequency resolution without changing the frame length of the digital signal being analyzed [36]. Examples of windows are Rectangular, Hamming, Hanning and Blackman-Harris windows. More information on the STFT and windowing processes can be found in [36] and [26].

As explained before, the hop-size H is the number of samples each frame advances on the digital signal for the DFT analysis. A different approach to H is known as *overlapping percentage*, since it represents a portion of N that overlaps between one frame analysis and the next one.

2.3 Music Information Retrieval

2.3.1 Definition

Music Information Retrieval (MIR) is an interdisciplinary science dedicated to obtain representative features from music by automatic implementations. These features may be related to meaningful dimensions of music such as timbre, melody, harmony and rhythm [17]. Since musical pieces are presented in digital formats nowadays, features are obtained from temporal evolution and frequency spectra of digital music signals using the STFT. Given that they are obtained from the raw information of the audio signal, they are known as *low-level features*. The analysis of combined low-level features can define the dimensions of music mentioned earlier in this paragraph [22].

2.3.2 Temporal Features

Among the most common low-level temporal features for MIR are the following:

- Zero-Crossing Rate (ZCR): Number of temporal sign changes on the audio signal. Commonly used to determine the noisiness of a signal. It is calculated by:

$$Z_t = \frac{1}{2}\sum_{n=1}^{N-1} |sign(x(n)) - sign(x(n-1))| \tag{2.4}$$

  Where $sign(x(n))$ is 1when $x(n)$ is positive, and 0 otherwise [31].

- Energy Envelope: Root Mean Squared (RMS) value of the audio signal, usually performed over different frequency ranges, or bands, of the spectrum. Used as intermediate process for onset detection or beat tracking [20].

- Periodicity Functions: Algorithms that find recurrent behaviors between frames of the audio signal and periods of time when these recurrences occur. An example is the autocorrelation function. Used to determine an estimate of the tempo (speed) of a song [22].

2.3.3 Spectral Features

On the other hand, common low-level spectral features are the following:

- Brightness: Measurement of the spectral energy above a threshold frequency, calculated by:

$$b_r = \frac{\sum_{k=0}^{N-1} X(k) - \sum_{k=0}^{k_b} X(k)}{\sum_{k=0}^{N-1} X(k)} \tag{2.5}$$

  Where k_b is the frequency bin correspondant to the threshold frequency. It is used to provide aditional information about the pitch of a song and the overall timbre of a music audio signal [19].

- Roll-off: Calculation of the frequency value up to which a certain percentage of the total spectral energy is located [31]. Given by:

$$\sum_{k=0}^{k_r} X^2(k) = p_r \sum_{k=0}^{N} X^2(k) \tag{2.6}$$

Where p_r is the fraction of the total energy and k_r is the frequency bin correspondant to the roll-off frequency. It is used to describe the shape of the spectrum [22] and to identify timbre, which is the characteristic sound of a music instrument, in combination with other features [9].

- Spectral Centroid: Considering the spectrum as a distribution, the centroid is the geometrical center of the spectrum. It gives information about where the highest concentration of energy is [31]. Calculated by:

$$s_c = \frac{\sum_{k=0}^{N-1} kX(k)}{X(k)} \tag{2.7}$$

- Spectral Spread: Based on the previous feature, is a measure of the dispersion, or spread, of the distribution around the spectral centroid [12]. It is calculated by:

$$s_s = \frac{1}{N-1} \sum_{k=0}^{N-1} (X(k) - s_c)^2 \tag{2.8}$$

- Spectral Flatness: Measurement of noise of a frequency spectrum. Values range from 0 to 1, indicating less noisiness as the value increases. It is computed for several frequency bands [19]. Calculated by:

$$s_f = \frac{\sqrt[N]{\prod_{k=0}^{N-1} X(k)}}{\frac{1}{N}\sum_{k=0}^{N-1} X(k)} \tag{2.9}$$

It is used to detect tonality on a music audio signal. Values close to 1 indicate

a noisy signal and close to 0 indicate a signal made of pure tones or sinusoids.

- Mel-Frequency Cepstrum Coefficients (MFCC): The mel-cepstrum is the discrete cosine Transform (DCT) of the logarithmic spectrum after a nonlinear frequency warping onto a perceptual scale called the Mel scale [2]. A number of l coefficients c_l can be calculated by:

$$c_l = \sum_{q=1}^{Q} \chi(q) cos(l \frac{\pi}{Q}(q - \frac{1}{2})) \tag{2.10}$$

Where:

$$\chi(q) = ln(\sum_{k=0}^{N-1} |X(k)| \cdot H(k,q)) \tag{2.11}$$

Where $q = 1, ..., Q$, $H(k,q)$ is the Mel Filter Bank, and Q is the filter bank number.

Low order MFCC's give information about smooth changes on the spectrum, while high order MFCC's give information about sudden variations. They are widely used on speech recognition systems, musical instrument detection and timbre modeling [27].

- Chromagram: The chromatic scale is a western musical scale with 12 equally spaced pitches or notes. On a piano keyboard, repetitions of these 12 notes are placed. Each repetition is called an octave. Among different octaves, the names of the notes are kept the same, but the pitch of each note increases by doubling the frequency of that same note on the previous octave.

The chromagram is a 12 bin histogram, each corresponding to a note on the chromatic scale, not considering the octave it belongs to. A graphic reprensentation of a chromagram is shown in figure 2.3. It can bring important information about the melody, tonality and musical scale [3]. Chromagram features

are used for extracting musical key [18], for extracting general information about tonality [6], and to detect cover songs [24].

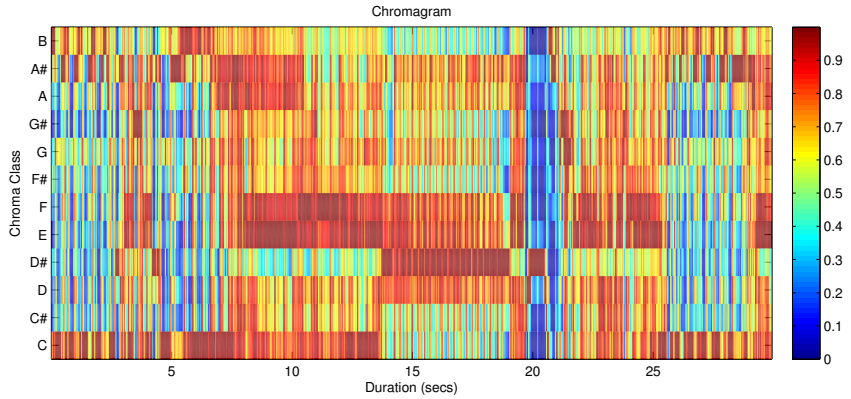


Figure 2.3: Graphic representation of a chromagram for a 30 seconds musical audio signal.

2.4 Genre Classification

2.4.1 Background

Music genres are labels created by humans, used to identify songs based on the instrumentation, rhythmic description and harmonic content of the music. To categorize music, a list of common characteristics from songs that belong to a specific genre must be elaborated to distinguish one genre from another. This group of characteristic elements is called *taxonomy* [22]. In addition, recent changes in music industry have forced the development of genre identification methods and techniques to manage song databases, which have been growing during the last years thanks to the appearance of digital formats.

Music software such as iTunes and browsers like Last.fm rely on typed information known as metadata to gather similar artists, classify their content and analyze similarities between users' libraries to make future recommendations. Despite the effectiveness this method has shown, it is based on cultural metadata, which shows a dependency on musical experience and other non-music related knowledge such

as capitalization and spelling. Web 2.0 applications have made metadata content approval more democratic and generalized, but external elements such as cultural background, geographic regions and the number of users make metadata-based classification a relative and complex task [22]. Even if music experts such as musicologists were to create metadata, it is physically unfeasible. It is reported on [1] that the manual labeling of 100,000 songs on Microsofts MSN music search engine would take 30 musicologists a year to do it.

2.4.2 Automatic Genre Classification

An alternative proposed by MIR is the automatic genre classification based on the processing of the recorded audio waveform [9]. It basically consists of extracting temporal and spectral low-level features from a large database of songs from different genres by means of the STFT, described on section 2.2.2, and using machine learning algorithms to train categorization systems known as classifiers. These systems find structural patterns on data and organize them as a set of rules that allow making predictions about new incoming data. The stage where the classifier learns about patterns on available data is called training, while testing is the stage where new data is given to the classifier to verify its accuracy.

Having a large number of features for genre classification does not necessarily mean a better one. The use of high amounts of features might bring a very specific system that would not work when the input dataset of songs is changed. Creating this narrow margin on a classification system is called *overfitting* [32]. For this reason, a limited number of features must be pre-selected for training and testing the classifier. A common practice is to select a number of features below 5% of the number of instances. An often used pre-processing technique is the principal components analysis (PCA) [32]. It is a method that reduces data dimensionality used to reveal tendencies on data [28]. The results of PCA are weighted sums of grouped features, resulting in a reduced amount of total features used for training and testing the classifier.

Usually, 30 second segments of musical audio signals are used for genre classification tasks, as well as a limited number of genres. The first is due to similarities on instrumentation, rhythm and tonal characteristics throughout a complete song, which can be detected over a short segment. The second is due to the lack of a taxonomy that defines more specific genres than 'Rock' or 'Pop' and because overfitting is being avoided by not using large databases. In [7] song segments of 30 seconds and 8 different genres were used. In [16] 7 genres were used, while on [31] the dataset consists of 20 musical genres.

A subset from the whole collection of songs must be used for training and a different subset for testing. These subsets are chosen using *stratification*, which selects random songs keeping the proportionality of the genres from the whole set on the chosen subsets [32]. If this process is repeated several times, the effect of particular subsets on the classification system will be reduced, mitigating the overfitting explained on the previous paragraph. This whole process is called *M-fold validation*, where M stands for the number of iterations and subsets created for the training and testing processes. On every iteration, a number of $M - 1$ subsets is used to train the classifier, while the remaning subset is used to test it.

2.4.3 Classifiers

Among the most common classifiers used for automatic genre classification the following are found:

- Support Vector Machines (SVM): learning algorithm that selects a few critical instances from a specific genre called support vectors. The support vectors are located on a hyperplane, which can be seen as a multidimensional plot where each axis corresponds to an extracted low-level feature. From the position of the support vectors on the hyperplane, boundaries can be calculated by quadratic, cubic, or high order functions known as kernels. These boundaries are known as maximum margins, which separate groups of songs belonging to

a specific genre from others belonging to a different genre [22].

- Nearest Neighbors (KNN): instance-based learning algorithm based on vicinity. Each new song is compared to the training subset of songs by a distance metric. The classification is done by labeling the new song with the same genre as the majority of training songs that have the closest distance to it [1].

- Gaussian Mixture Models (GMM): algorithm that calculates the probability density of a genre on a space created by the values of extracted low-level features [22]. The probability density is a mixture of multidimensional Gaussian distributions, where each dimension corresponds to weighted probability functions of extracted low-level features [31].

2.4.4 Common Methodology on Genre Classification

Figure 2.4 represents a common methodology on genre classification tasks. First, the audio signal is framed, windowed and transformed into its frequency representation using the FFT. These 3 processes are done by the STFT. Temporal features are extracted from each frame, while spectral features are extracted from the frequency spectrum of each frame. To have meaningful values of the whole audio file, the mean and variance of each time series of features are calculated. Then, the set of means and variances is used to train and test the classifier. The accuracy results are obtained after testing.

2.5 Nonlinear Time Series Analysis

A very recent approach in MIR is the use of Nonlinear Time Series Analysis (NTSA) techniques to extract new features from the audio signal itself. These techniques are used with the assumption that the temporal description of an event is a variable which affects the development of a more complex time-evolving system.

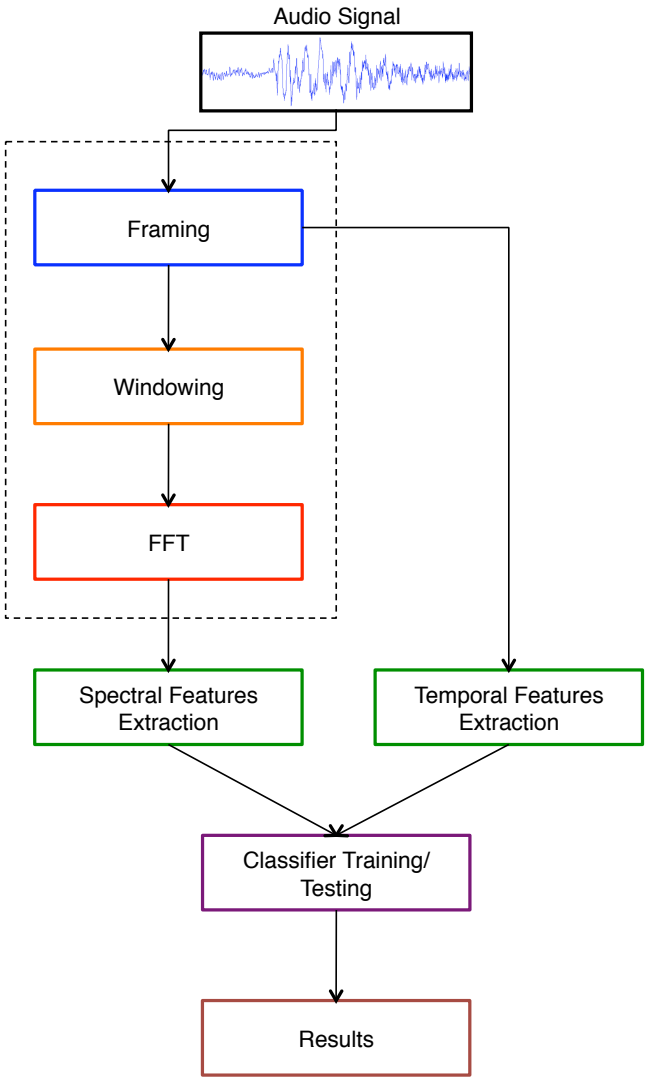


Figure 2.4: Common analysis for genre classification tasks: the dotted square represents the processes done by the STFT. Temporal features are extracted from the audio frames, while spectral features are taken from the frequency spectrum of the audio frames, obtained by the FFT. These features are used to train a classifier and test for its accuracy in predicting a target label.

2.5.1 Nonlinear Time Series Analysis Techniques

One common nonlinear time series analysis technique is known as *State-Space Embedding*. In real-world physical systems, all the factors or *variables* that contribute to the temporal evolution or *dynamics* of the system cannot be accessed straightforwardly. State-space embedding consists in creating a multidimensional space from delayed sets of a time series that defines the temporal evolution of a variable, giving a topological similarity to the dynamics of the system where all the variables are full-known [34].

Assuming that musical audio signals are time series describing a physical system allows to create a different representation of its temporal evolution. As a consequence, information describing its nonlinearities, which might not be given by usual audio analysis tools such as the FFT, can be obtained. State-space embedding is scantily suggested by [15] to discriminate between rock/pop songs and classical songs, where the state variables have smoother changes in the latter case. In [16], the state-space embedding is used on time series of low-level features to obtain NTSA features, based on the resulting trajectory of the state-space.

Another important NTSA tool is the *Recurrence Plot* [23]. It is a technique implemented to measure patterns or repetitive behaviors on the trajectory defined by a state-space embedding [14]. This technique has been used in [25] as a method to detect cover songs, which are versions of a previously existent songs possibly made by a different artist from the original, usually with the same musical arrangements and tonality.

A speech recognition application is explained in [29], where a periodicity histogram is built with recurrence information extracted from the state-space. By knowing the time when this recurrences occur, an estimate of the fundamental frequency of the audio can be found [5]. In [33], a combination of the state-space embedding followed by recurrence plot analysis is done over time series of extracted chromagrams to create new visualization tools that help users to identify structure in music.

The application of these techniques on this research is described in chapter 4. Additional information on nonlinear time series analysis techniques can be found on [10], [34] and [23].

There is little work done in this audio anaysis approach, but it has been shown that NTSA applied over audio signals can bring interesting new results in the feature extraction and audio classification fields of MIR.

Chapter 3

Methodology

This chapter describes the audio files processing scheme used in this research for evaluating common classification methodology. It also details the feature selection procedure, and lists the classifiers used for accuracy evaluation.

3.1 Database

The audio files used for the evaluation come from a specific database provided by George Tzanetakis. It is divided in 10 genres: Rock, Pop, Reggae, Metal, Hip Hop, Classic, Country, Jazz, Disco and Blues. Each genre consists of 100 song excerpts of 30 seconds in duration, excepting Reggae genre with 93 excerpts, making a total of 993 audio files. The files were provided in wav format, mono channel and sampled at 22,050 Hz.

3.2 Audio Processing

The process described in this section is done for both common classification methodology (based on frequency spectrum features) and the nonlinear audio recurrence analysis (described on the next chapter) independently. For the common methodology, the STFT is applied over the audio files with the following parameters: frames

of 2048 samples long, using 50% of overlapping between frames, zero padding of 2048 samples and a Blackman-Harris 92dB window. The FFT is then applied on 4096 samples, intented to create a frequency spectrum of 2048 bins for the frequencies up to the Nyquist frequency f_n.

The STFT is calculated using MIRtoolbox for MATLAB [13]. Developed at the University of Jyväskylä by members of the Finnish Centre of Excellence in Interdisciplinary Music Research, MIRtoolbox is a set of functions developed for MATLAB, dedicated to the extraction of low-level and high-level features from audio for Music Information Retrieval tasks. It is designed as a modular framework where each block develops a particular duty. These blocks can be parametrized by the user and can be interconnected to achieve different purposes. MIRtoolbox version 1.2.3 is used on MATLAB R2009a. On this methodology, the functions used to calculate the STFT of an audio file are subsequently mentioned. Unless stated otherwise, the default parameters of MIRtoolbox functions are used:

1. **miraudio()**. Extracts the audio from a 'wav' file as samples.

2. **mirframe()**. Divides the audio samples into frames of length and overlap given as parameters.

3. **mirspectrum()**. Calculates the spectrum of every frame, applying the window given as a parameter and using the MATLAB FFT algorithm. The zero padding is added by this function internally. The frequency resolution obtained using the FFT parameters described above is 5.3833 Hz/bin.

3.3 Spectral Features

The following features, described in section 2.3.3, are extracted from the frequency spectrum on common methodology, and from the histograms described on sections 4.5 and 4.6 for the nonlinear audio recurrence analysis. The feature extraction is

done using particularly created functions for each feature, contained in MIRtoolbox for MATLAB [12]:

- Statistical moments: mean, variance, skewness and kurtosis.

- Mel Frequency Cepstrum Coefficients (MFCC): the Discrete Cosine Transform of the logarithm of the spectrum, calculated over Mel bands. Represents the shape of the spectrum in a few coefficients. Using a bank of 50 filters, 20 coefficients are computed for the evaluation.

- Chromagram: distribution of the spectral energy on the 12 semitones of the chromatic scale, without discrimination of the octave they belong to. Consequently, 12 values are computed.

- Brightness: percentage of the spectral energy located above a certain frequency threshold. Employed value is 3000 Hz.

- Roll off: frequency value up to which 85% of the spectrum energy is located.

- Spectral Centroid: geometric center of the spectral distribution.

- Spectral Spread: also known as standard deviation, it measures the dispersion of the spectrum around the spectral centroid.

- Spectral Flatness: determines the smoothness of the spectrum. Values close to 1 indicate a noisy signal and close to 0 indicate pure tonality.

A total of 41 features are computed for each frame. To obtain values significant to the whole audio file, the mean and the variance of each time series of features is calculated, giving a total of 82 features per audio file. This setup remains the same for both common methodology and nonlinear audio recurrence analysis.

3.4 Feature Selection and Classification

The processes from this section are done over the dataset of features extracted from common methodology and over the datasets of features extracted from the nonlinear audio recurrence analysis in different combinations, as will be seen on chapter 5. In the order they are mentioned, three feature selection processes are applied on the dataset of spectral features to achieve effective results on the classification task. This feature selection is achieved via the filter implementations on WEKA Explorer. WEKA is a collection of machine learning algorithms for data mining. It contains tools for data pre-processing, classification and clustering, among others [8]. For this methodology, version 3.6.2 is used. The functions used for feature selection are mentioned next. Unless stated otherwise, the default parameters of these functions are used:

1. Attribute Selection: Supervised processing where the most correlated features to a genre are selected. Using the following parameters:

   (a) Evaluator: **cfsSubsetEval**. Evaluates the features by considering individual predictability and global redundancy.

   (b) Search: **BestFirst**. Searches the best features in descending order, starting with the first extracted feature to the last one.

2. Principal Components: Linear and weighted combinations of selected features that reduce multidimensionality of data. Each combination is called a 'component' [28]. Using the following parameters:

   (a) Maximum Attributes: -1. Indicates no limit in the number of features taken for creating each component.

   (b) Variance Covered: between 0.96 and 0.99. The value is changed inside this range until 30 principal components are created, which is the number of principal components taken to analyze the baseline.

3. Normalization: The values of a given feature are normalized from a maximum of 1 to a minimum of 0.

After this stage, the number of selected features for classification is 30. The classification task is done on WEKA Experimenter using the dataset of selected features. Then, different classifiers are employed to ensure the results are not based on one specific classification technique. The default parameters of each classifier are kept unless stated otherwise:

1. Zero Rule Classifier (**0R**): Algorithm that classifies according to the majority genre. The result of this classifier corresponds to a classification based on a random guess. Thus, it represents a theoretical baseline to be surpassed by any other classifier.

2. One Rule Classifier (**1R**): Classification based on a single feature, characterized for having the minimum prediction error. The feature that individually discriminates the most between genres is selected for the task[1].

3. Naïve Bayes (**Bayes**): Probabilistic classifier based on Bayes' theorem. Assumes the presence of a particular feature on a genre as completely unrelated to the presence of any other feature [32]. The classification is based on a combination of individual feature probabilities[2].

4. K Nearest Neighbors (**IBk**): It is an algorithm whose classification is based on the vicinity of genres for a given combination of features. The parameter KNN (Number of taken nearest neighbors) is set to 5.

5. Multilayer Perceptron (**MP**): Classifier constructed over a back-propagation neural network. Depending on the inputs, each element of the network, called a *neuron*, is altered with a learning rate parameter in order to fit a given output. The order in which neurons are modified is from the last layer (closer

[1] Sael Sayad. Classication - basic methods, 2010. http://chem-eng.utoronto.ca/∼datamining/
[2] Naïve Bayes Classifier, September 2002. http://en.wikipedia.org/wiki/naive_bayes_classifier

to the output) to the first layer (closer to the input). The 'back-propagation' term is originated from this characteristic of the network [32]. The learning rate parameter is modified to 0.6.

6. Random Forest (**Forest**): Classification based on a group of decision trees, where groups of features are randomly selected at each node. The final output is the mode of the individual tree output [3]. The number of trees used in the classifier is modified to 100.

7. Support Vector Machines : It is an instance-based algorithm that selects boundary points, known as support vectors, to differentiate one genre from another [32]. Two different kernels are used to create different functions that maximally separates genres:

   - **PolyKernel** (**SVP**): Polynomial function.
   - **RBFKernel** (**SVR**): Radial-based function. Parameter **gamma**=0.6 [4].

8. Linear Logistic Model (**SL**): found on WEKA as SimpleLogistic, is a classifier that fits the data of the selected features into a sigmoid curve or logistic function, to calculate the probability of a genre to be predicted [32]. The parameter **useAIC** is set to **True**.

The classifier training and testing is executed using a 3-fold cross validation, iterating 10 times for each classifier. This setup remains the same for both common classification methodology and nonlinear audio recurrence analysis.

[3] Random forest, January 2005. http://en.wikipedia.org/wiki/Random_forest
[4] Radial basis function, July 2005. http://en.wikipedia.org/wiki/Radial_basis_function

Chapter 4

Nonlinear Audio Recurrence Analysis

On this chapter, the nonlinear time series analysis of the audio signal is presented. The different parts conforming this analysis are described as well. Finally, the development of the time recurrence and frequency recurrence histograms is exposed.

4.1 Nonlinear Time Series Analysis Module

The nonlinear time series analysis module replaces the windowing and the FFT stages from the common methodology used for feature extraction. The sequential processing followed inside the module is: audio framing, state-space reconstruction, computation of the recurrence plot, calculation of the recurrence time histogram, and its transformation into the correspondant recurrence frequency histogram. Figure 4.1 shows a graphic version of this module. The following sections will explain in detail the signal processing each step performs on the audio waveform.

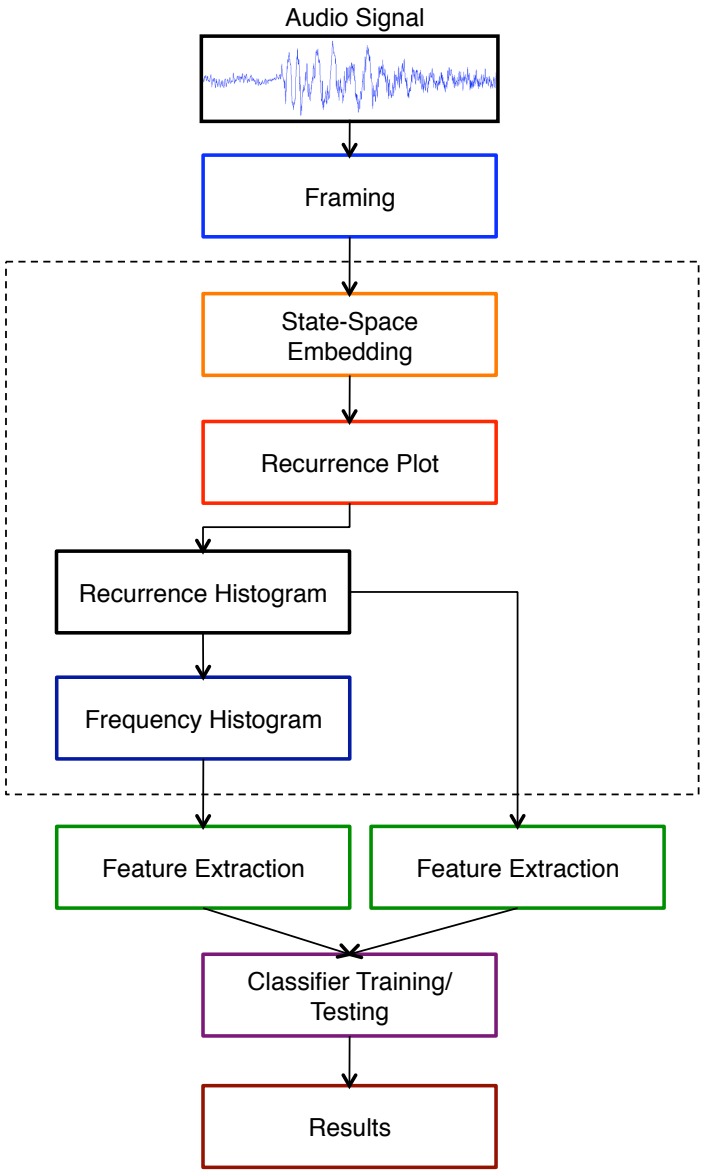


Figure 4.1: The nonlinear analysis module, delimited by the black dotted line, replaces the windowing and the FFT stages to extract features from the resulting recurrence time histogram and recurrence frequency histogram.

4.2 Audio Framing

The FFT calculation from the MIR Toolbox uses zero-padded frames to have the same number of positive frequency bins as number of samples on the original audio frame. This is 2048 frequency bins up to the Nyquist Frequency bin for 2048 samples on the audio frame. Therefore, the audio waveform is divided into frames of 2048 samples to keep the same bin reference when extracting the features. A value of 50% overlapping between frames is used. Different from common methodology, the frames are not windowed. As mentioned in [26] the windowing process tappers the ends of the analyzed data, making the spectrum a smooth function. Since the nonlinear analysis is done over the unaltered audio frame, this step is not needed.

4.3 State-Space Embedding

As primary step to recurrence analysis, a technique known as State-Space Embedding is applied to each audio frame. The process consists in converting each sample of the audio signal into a vectorial form whose dimensions are given as a parameter. This parameter is known as *embedding dimension*. Each vector is known as a *state*, and it describes a point in the multidimensional space. The temporal evolution of states in the multidimensional space results in the development of a trajectory which describes the behavior of the audio signal at specific points in time. The resultant trajectory allows modeling, prediction, and pattern analysis in the signal. This process is applied to individual audio frames, meaning that a state-space reconstruction (and the subsequent processes applied onto it) will be calculated framewise.

For a j-th sample on an audio waveform frame $S(j)$, the resulting m-dimensional state-space $\mathbf{v}$ is calculated by:

$$\mathbf{v} = [S(j), ..., S(j - (m-1)\tau)] \tag{4.1}$$

for $j = \eta, ..., N$ where $\eta = (m-1)\tau$, m is the embedding dimension and τ is the

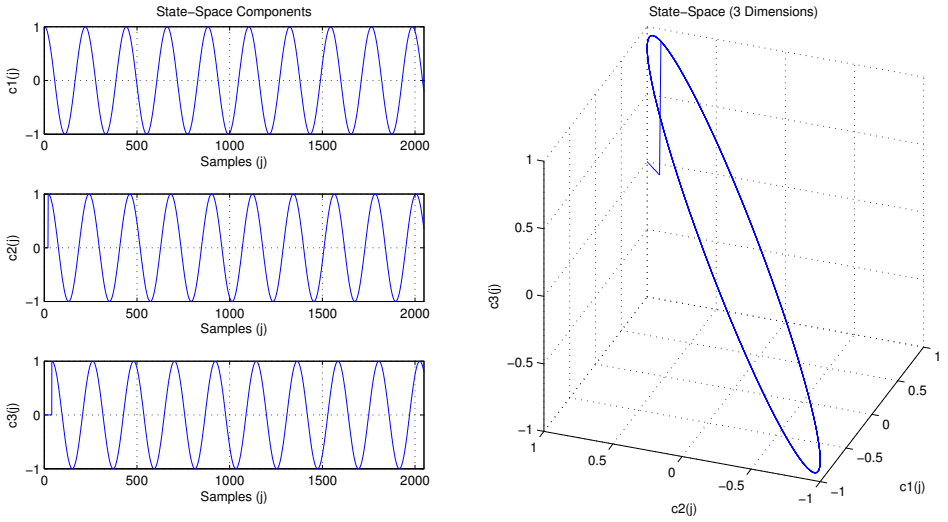


Figure 4.2: State-space reconstruction on a sinusoidal signal using m=3 and τ = 20. The components of each dimension are shown on the left, while the resultant trajectory is shown on the right.

delay time in samples.

A simple example of the construction of the state-space is provided for a sinusoidal signal using m=3 and $\tau = 20$. Figure 4.2 shows the individual components, being $c1(j)$ the original audio frame, and $c2(j)$ and $c3(j)$ the delayed components. The same figure shows the state-space reconstruction on a three-dimensional space. As can be seen, the trajectory of the sinusoidal signal is a circle, which has a periodic behavior due to the periodicity of the signal.

An example that represents the processing done on a musical excerpt using m=3 and $\tau = 20$ is provided on figure 4.3. This state-space diagram corresponds to an audio frame from a song belonging to the Blues genre of the analyzed database. The same method using a different audio frame from a song belonging to the Metal genre is shown on figure 4.4. Thanks to the defined trajectories on the state-space, predictions of future states and recurrence analysis can be achieved easily than analyzing the audio signal per se. The resulting trajectories for the audio frames are not as straightforward as the circle for the sinusoidal signal, so the recurrence analysis is done through a recurrence plot, which is introduced in the following

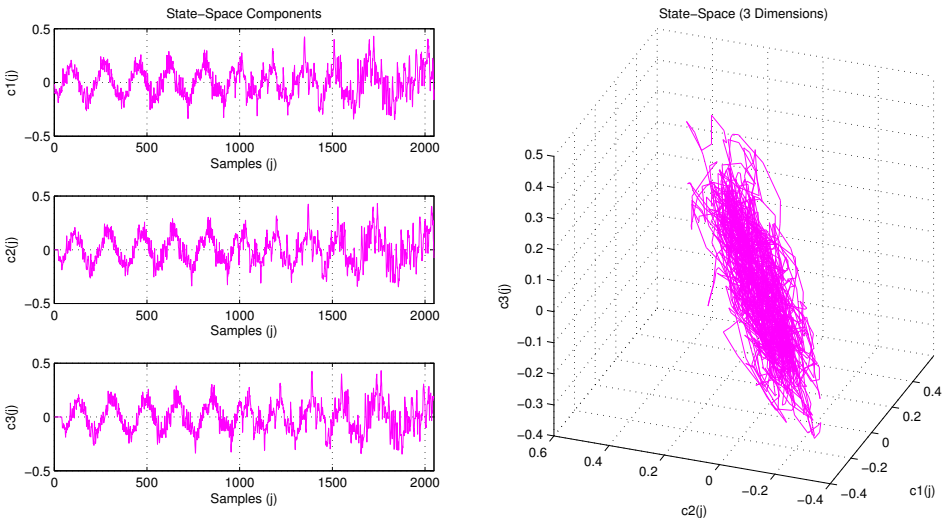


Figure 4.3: State-space reconstruction on a Blues genre audio frame using m=3 and $\tau = 20$. The components of each dimension are shown on the left, while the resultant trajectory is shown on the right.

sections.

Several techniques for obtaining suitable values of m and τ can be found and implemented. Examples of these techniques are false nearest neighbors for m, which is described on [11], and auto-correlation function or the mutual information function for τ, mentioned on [14]. Since one of the goals of this research is to verify how changes on these parameters affect the classification accuracy, the techniques for obtaining suitable values of these two parameters are not applied.

4.4 Distance Matrix

If two points of the state-space trajectory have a small distance value, it is said they correspond to similar states. Therefore, the state similarity between two points can be defined as a recurrence in the signal.

From the state-space embedding, squared Euclidean distance is calculated between pairs of points that conform the trajectory. The intention is to know how close these points are from one another. To calculate the squared Euclidean dis-

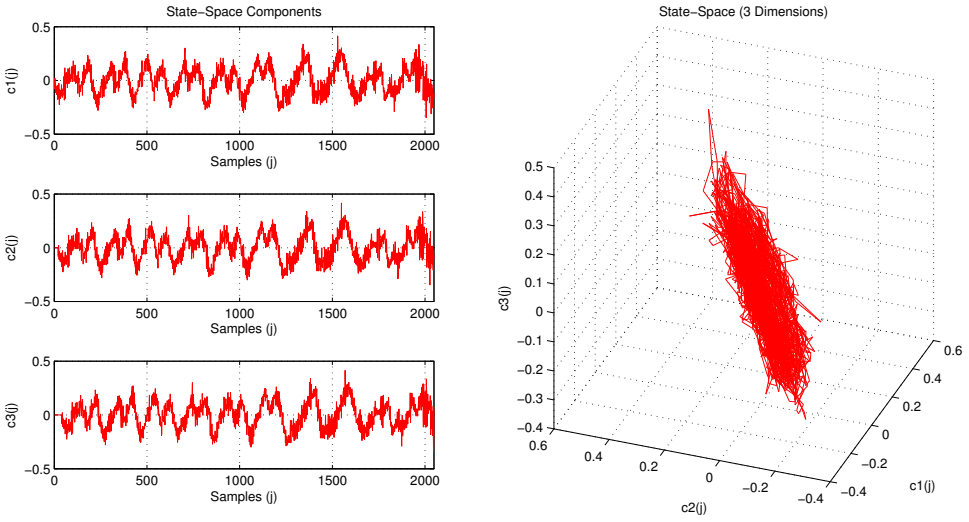


Figure 4.4: State-space reconstruction on a Metal genre audio frame using m=3 and $\tau = 20$. The components of each dimension are shown on the left, while the resultant trajectory is shown on the right

tance between two points the following equation is used:

$$D_{a,b} = \sum_{r=1}^{m} (v_{b,r} - v_{a,r})^2 \tag{4.2}$$

where $D_{a,b}$ is the distance matrix holding the distance values between all the a-th and b-th positions on the phase-space trajectory. A consideration to take when making this calculation is that small distance values are also valid for consecutive points on the same trajectory of the dynamics, which cannot be considered as recurrences since they belong to the development of close states. As a consequence, a window that excludes the processing of adjacent points on the trajectory must be applied. A parameter known as the Theiler correction window can be introduced on equation 4.2, by restricting the values of b from $a+1+w$, where w is the value of rejected consecutive points on the trajectory, to N, the audio frame length. These values of b are kept throughout this chapter. Figure 4.5 shows the distance matrix for the Blues genre audio frame and the Metal genre audio frame.

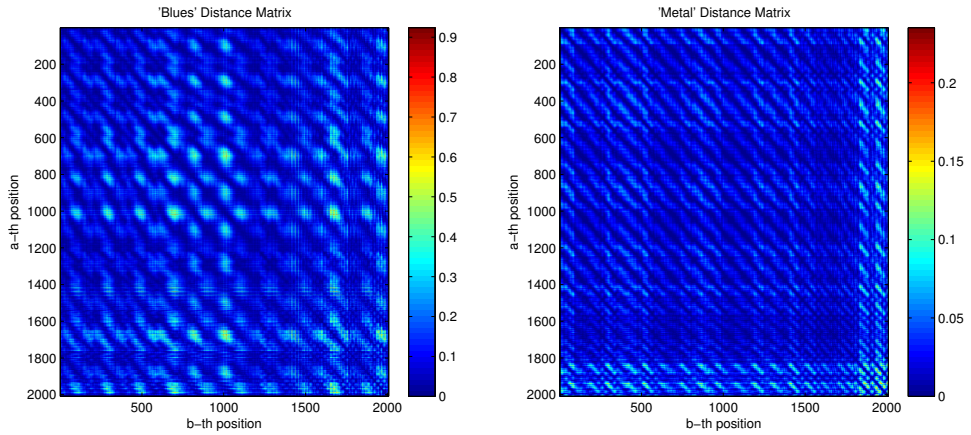


Figure 4.5: Distance matrices for the Blues genre audio frame on the left and for the Metal genre audio frame on the right. A repetitive behavior or pattern can be seen on both audio frames by the overall shape and diagonal lines of each distance matrix.

4.5 Recurrence Plot

A threshold is then defined as a discriminator for high distance values. The calculation of the threshold allows it to change dynamically depending on the distance values for a specific frame. To obtain the threshold, a percentage of the mean of all distances on the signal frame (shown on the recurrence plot) is taken:

$$\varepsilon = p\frac{\sum_{a=1}^{N}\sum_{b=a+1+w}^{N} D_{a,b}}{N(N-1-w)} \tag{4.3}$$

Where ε is the threshold value and p is the proportion of the mean of the distance matrix, whose value can be adjusted as a parameter from 0 to 1.

Since the time separation between points in the trajectory can be given in samples, the recurrences can be compared to integer-valued sample lags in what is known as a recurrence plot. The recurrence plot is a visual aid to identify the repetitive points in a given state-space representation. It is useful to detect a recurrent behavior in the analyzed signal. The recurrences are shown in a squared matrix form

where the axes represent the a-th and b-th positions on the trajectory.

A comparison between the distance matrix and the threshold value outputs a new matrix given by:

$$R_{a,b} = \Theta(\varepsilon - D_{a,b}) \tag{4.4}$$

Where $R_{a,b}$ is a matrix holding the recurrences taken and Θ is the Heaviside function, where $\Theta(y) = 1$ when $y > 0$ and 0 otherwise.

The previous processing will return the recurrence plot filled with ones and zeros exclusively. It indicates which pairs of points in the trajectory are taken as recurrences and which ones are left apart respectively. Graphic examples of recurrence plots can be seen on figure 4.6, where the same audio frames analyzed so far are being used. The parameters used for plotting these figures are m=3, τ=20, w=10 and p=0.3. Further analysis on the variation of these parameters and its influence on the recurrence plot is done on chapter 5.

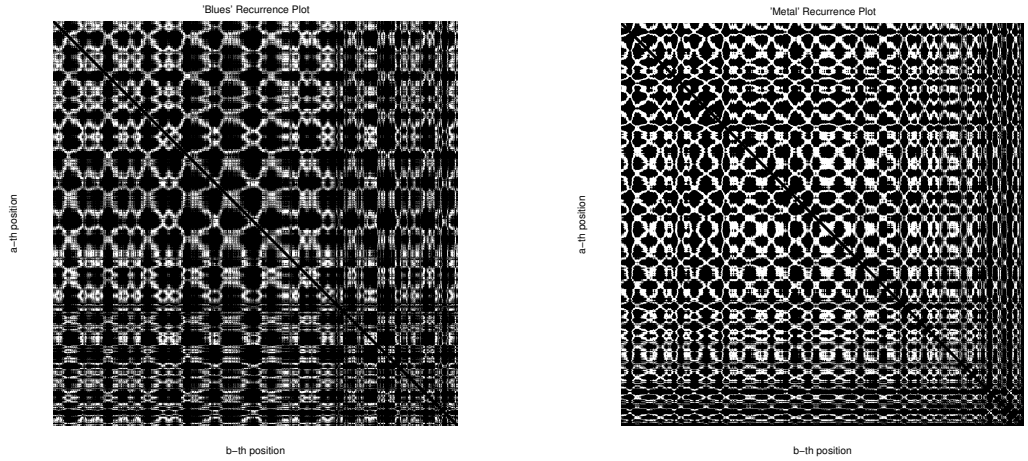


Figure 4.6: Examples of recurrence plots for the Blues genre audio frame on the left and for the Metal genre audio frame on the right. The binary nature of the matrices indicate whether a pair of points is taken as a recurrence (white) or if it is left out of the analysis due to high distance between the points (black). The repetitive behavior is seen clearer than in the distance matrices.

4.6 Recurrence Time Histogram

Following the guidelines stated in [29] and [30], a recurrence time histogram H_t is built as a previous step towards a recurrence frequency histogram creation:

$$H_t(k) = \sum_{a=1}^{N-k} R_{a,a+k} \tag{4.5}$$

Where k is the number of bins on the histogram, which also represents the time difference in samples between two points in the trajectory considered as a recurrence. This value will be refered as *sample lag* on future sections.

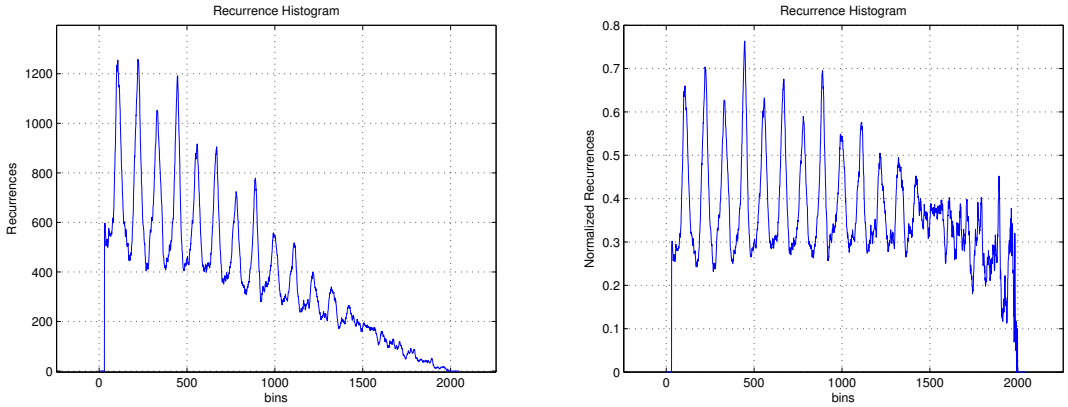
Since the limits of the summation in equation 4.5 decrease when k increases, normalization must be done in order to eliminate the decreasing tendency on the histogram. This can be achieved by dividing the recurrence counts on each bin by the number of total possible counts on that bin. The normalized histogram is then calculated as:

$$\bar{H}_t(k) = \frac{1}{N-k} \sum_{a=1}^{N-k} R_{a,a+k} \tag{4.6}$$

Figure 4.7 shows examples of the recurrence time histogram without normalization and after it has been normalized. The same methods used for obtaining spectral features described on section 3.3 will be used on the recurrence time histogram.

4.7 Recurrence Frequency Histogram

The building of a recurrence frequency histogram (H_f) departs from knowing two fundamental parameters: the sampling frequency of the audio signal and the sample lag of the found recurrences. The former is given by the audio files, while the latter is obtained from the k-th bin of the recurrence time histogram as explained on section 4.6. To obtain the corresponding frequency of a sample lag k having a sampling



(a) Unnormalized H_t (b) Normalized H_t

Figure 4.7: The recurrence time histograms before normalization (a) and after normalization (b) are shown. The decreasing tendency of H_t caused by the increasing k is eliminated when dividing by all possible recurrence on the correspondant bins. Tne normalized values go from 0 to 1.

frequency f_s , we use:

$$f(k) = \frac{f_s}{k} \tag{4.7}$$

Two facts can be observed from the last equation: first, high sample lags correspond to small values of frequency and vice versa. Second, the function has an inverse proportional behavior, meaning low frequencies will be spaced closer than high frequencies, which translates into a better resolution for high sample lags. Figure 4.8 shows the behavior of the function for the correspondent values of k .

As mentioned in section 2.2, the frequency binning of the FFT is a proportion of the frame length, equivalent to dividing the sampling frequency by the number of bins. On H_f, the binding is an inverse proportion of the sample lag, equivalent to dividing the sampling frequency by the sample lag$k$. Since the features to extract are developed for frequency spectrums obtained through FFT analysis, a frequency fitting is required. This frequency fitting consists in changing frequency values from the inverse proportionality given by equation 4.7 into equally-spaced frequency binding given by the FFT.

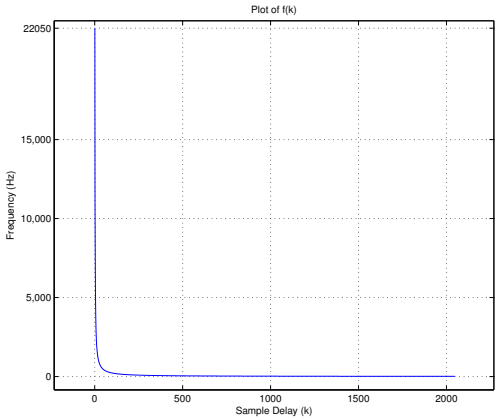


Figure 4.8: Frequency values as a function of k. The high frequency values are narrowed into a small area of the function, meaning these will have lower resolution than low frequencies when making the fitting on the recurrence frequency histogram.

The proposed fitting can be achieved by obtaining the frequency $f(k)$ from the sample lag of a found recurrence, and comparing it to the frequency values of the FFT binding. The smallest difference between $f(k)$ and the FFT frequency values will indicate the FH bin where $f(k)$ fits the best. Steps taken towards the frequency fitting are described next:

1. A vector H_f of length N is first initialized to zero. Since the value of N does not change over the analysis, this is calculated only once.

2. All the possible FFT positive frequency values for N bins can be calculated by:

$$F_i = \frac{f_s}{2N} i \tag{4.8}$$

   where the FFT bin index $i = 1, ..., N$. Since the value of N does not change over the analysis, this is calculated only once.

3. Starting from a recurrence on $R_{a,b}$, the value of k can be obtained by:

$$k = b - a \tag{4.9}$$

By equation 4.7, the frequency value for this recurrence is known.

4. The comparison between F_i and $f(k)$ is done to obtain all the differences between the FFT frequency binning values and the frequency as a function of the sample lag:

$$I_i = |F_i - f(k)| \tag{4.10}$$

5. The smallest value in I represents the closest location on the FFT frequency values where the frequency $f(k)$ can be adjusted to. Therefore, the H_f bin α is retrieved by:

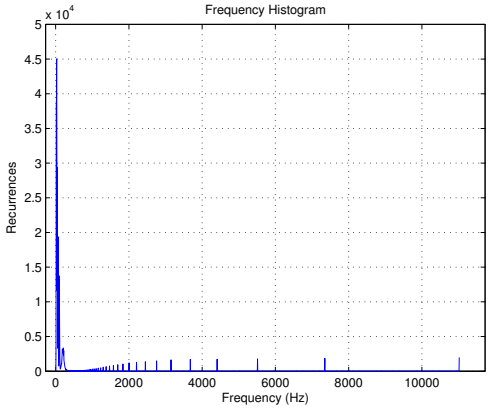
$$\alpha : I_\alpha = min(I) \tag{4.11}$$

6. The element $H_f[\alpha]$ is incremented by 1, meaning a recurrence with a frequency $f(k)$ has been fitted on binα of an FFT frequency binding.

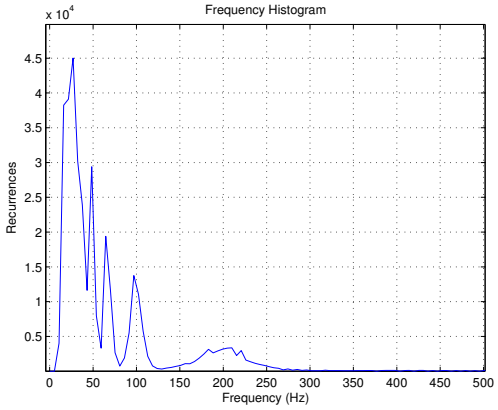
The previous process is then repeated for all recurrences. The normalization function follows the same procedure, but instead of using recurrences only, all the values from R are taken into account, whether being recurrences or not. The output for the normalization curve will be a vector S, so the normalized recurrence frequency histogram is calculated by:

$$\bar{H}_f[\alpha] = \frac{H_f[\alpha]}{S[\alpha]} \tag{4.12}$$

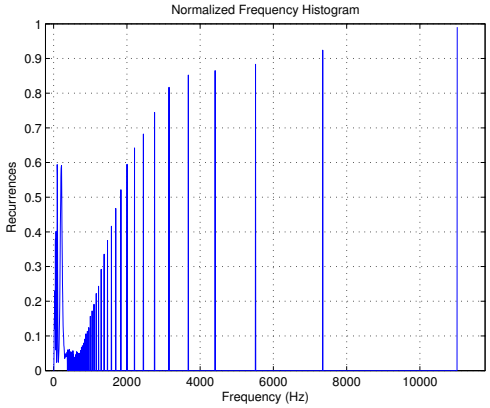
The frequency binning on the recurrence histogram, as in the frequency spectrum, is initially defined by the sampling frequency of the data. On the former, the values are given by an inverse proportionality, while the latter is equally divided into the number of samples used in the frame. Since the calculation of the frequencies



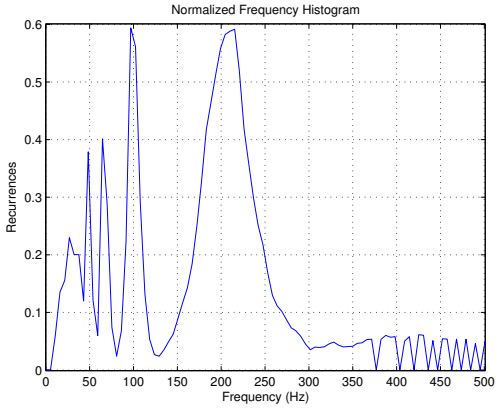
(a) H_f before normalization.



(b) Zoom on low frequencies in (a).



(c) Normalized H_f.



(d) Zoom on low frequencies in (c).

Figure 4.9: The recurrence time histogram translation into frequency outputs the frequency recurrence histogram on (a). By zooming at the low frequency section of the histogram a continuous behavior can be seen, which spreads out as the frequency increases and eventually creating peaks. When normalizing H_f, the high frequency peaks rise, due to the low resolution and the high amount of recurrences assigned to those specific bins. On the other hand, the continuous low frequency section, after normalization, show peaks similar to those of a frequency spectrum.

using sample lags might not derive in an exact frequency bin on the spectrum and can only be done with integer numbers, the rounding of the values will leave empty frequency bins for every calculated frame. For example: using 22,050 Hz as f_s and $N = 2048$, bin number 71 on the spectrum corresponds to 376.8311 Hz, which corresponds to a sample lag of 58.5143 samples. Since only integer values can be taken, 58 samples correspond to 380.1724 Hz, whose closest value in the spectrum values is bin number 72 (382.2144). On the other hand, taking 59 as sample delay (373.7288 Hz) results in assigning the recurrence on bin 70 (371.4478), which is the closest difference between the recurrence frequency and the equally spaced frequency values. The efect of the rounding can be observed in figure 4.9.

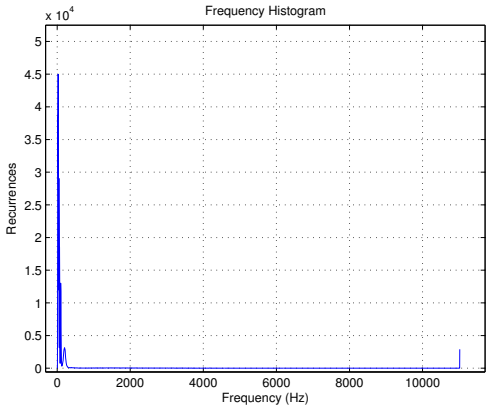
Given the high resolution of low frequency values given by equation 4.8, more values of *f(k)* will be fitted to the first bins of H_f. Therefore, H_f will present a continuous behavior on low frequencies and a spread non-continuous behavior as the frequency increases. To eliminate this effect, a random value ranging from -0.5 to 0.5 is added to k in equation 4.7. This action will spread the values of $f(k)$ horizontally, distributing the high frequency peaks in broader bin ranges and keeping low frequency peaks in shorter ranges. Consequently, H_f will have a continuous aspect on all fequencies.

Even if the same process is applied on the normalization function, the number of total possible recurrences on a bin will not be proportional to the considered recurrences belonging to that same bin, due to different random values added to k and to the normalization function. This effect is more influential on high frequencies, where the spread of the peaks is wider and the uncertainty of matching the same bin is higher. Therefore, the high frequencies are eliminated from the normalized H_f using a high value of w, taking into consideration the analyzed frame length and the time equivalent this length represents.

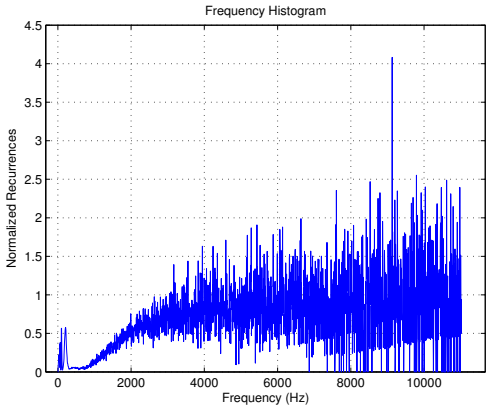
In figure 4.10a the effect of the added random vaue can be seen as a dispersion of the high frequency peaks in figure 4.9a. When normalizing this distributed H_f in figure 4.10b, the high frequency region rises with a random behavior due to the rea-

sons stated on the previous paragraph. Finally, when applying a Theiler correction window of 30, the high frequency values are eliminated, keeping the continuous low frequency region to be used on feature extraction.

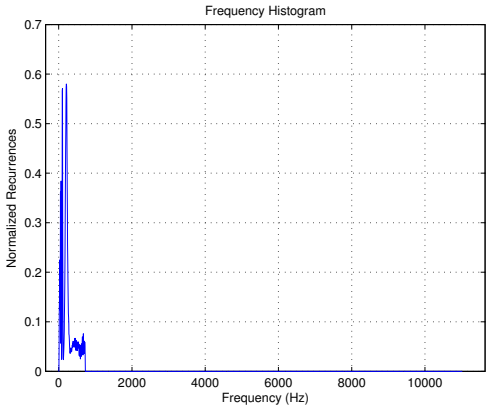
Comparisons between the frequency spectrum obtained through the FFT and the frequency recurrence histogram can be observed in figure 4.11. The same audio frames used to create the state-space embedding in section 4.3 are used for this purpose. It can be seen peaks are positioned on similar frequency values, while additional peak information can be found on the RH description of the audio frame. These are examples of recurrence frequency histograms where features described on section 3.3 will be extracted from.



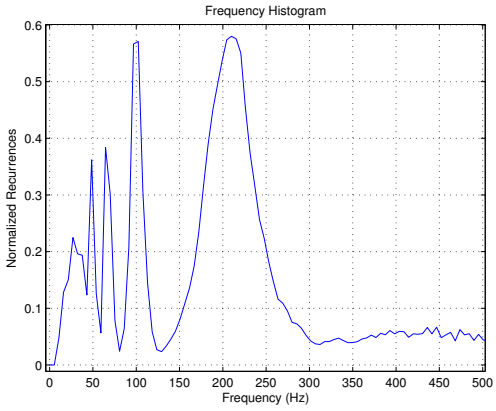
(a) Unnormalized distributed H_f.



(b) Normalized distributed H_f.

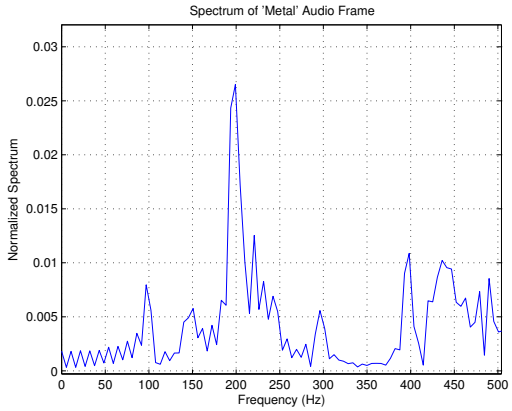


(c) Normalized distributed H_f with w=30.

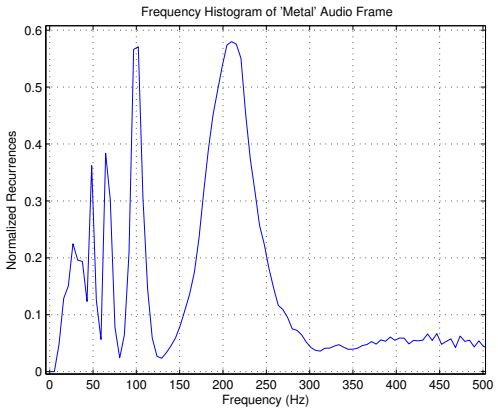


(d) Zoom on low frequencies in (c).

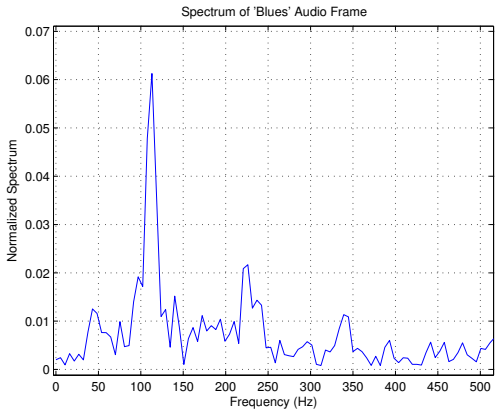
Figure 4.10: Adding a small random value at the sample lag when calculating the frequency fitting results in the distribution of high frequency peaks along the histogram distribution. However, different random values are added to the normalization function, which results in a random behavior on high frequencies after normalization. A considerable value in the Theiler correction window parameter w eliminates all the information from this part of H_f, making the continuous low frequency section, which remains the same, the only part of H_f providing concrete information about the audio signal.



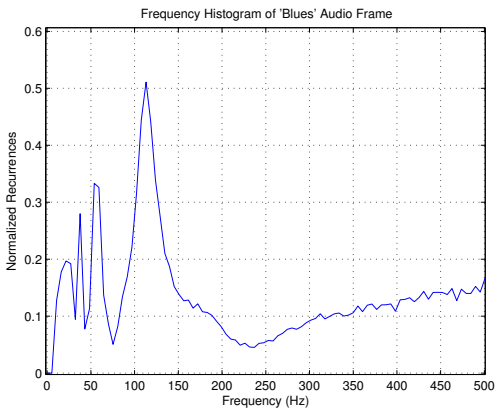
(a) Metal genre frequency spectrum.



(b) Metal genre H_f.



(c) Blues genre frequency spectrum.



(d) Blues genre H_f.

Figure 4.11: Comparison between frequency spectra and frequency recurrence histograms from Metal and Blues genre audio frames. The x-axis on four figures is a zoom on the low frequency region. The same range of low frequencies is compared, showing high peaks at similar frequency values, while showing different information on the rest of the frequency range, specially below the highest peaks.

Chapter 5

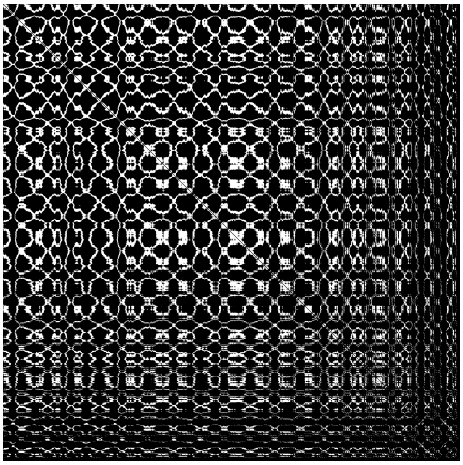
Results

This chapter explains the selected parameters for the classification task based on the effects they have towards the construction of the recurrence time and recurrence frequency histograms. It also presents and compares the accuracy percentages of the baseline-trained classifiers, as well as the ones trained using the extracted features from the proposed nonlinear time series analysis, and different combinations of them.

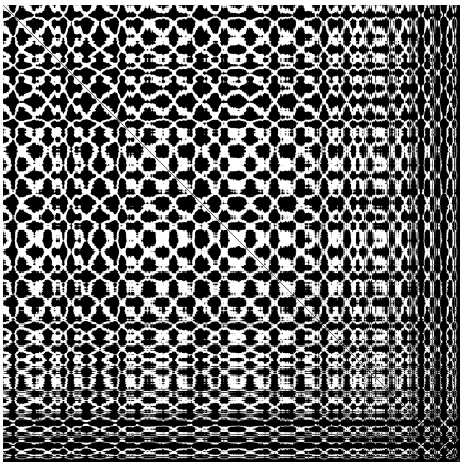
5.1 Parameter Assesment

Two important parameters for the construction of the recurrence plot are: the proportion p of the distance matrix mean, used for calculating the distance threshold, and the Theiler correction window w. If the parameter p is high, more pairs will be taken as recurrences. In addition, if w is high, more consecutive points will be left apart of the analysis. Examples of the effects of these parameters can be seen on figures 5.1 and 5.2 respectively, where the process is applied on the Metal genre audio frame analyzed in the previous chapter. The parameters used on each plot are indicated on the caption of each subfigure, using bold highlights on the changed parameters.

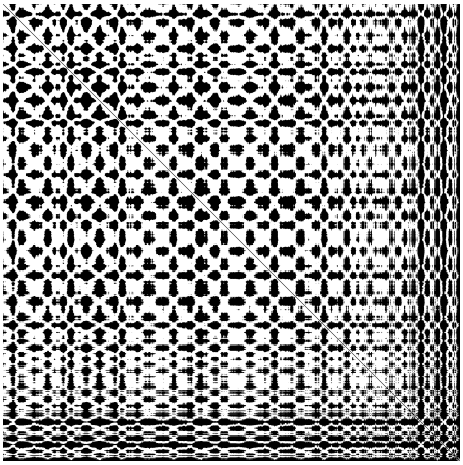
Figures 5.3 and 5.4 show different calculated recurrence time histograms for the Metal genre audio frame. The parameters used on each plot are indicated on the



(a) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.2**.

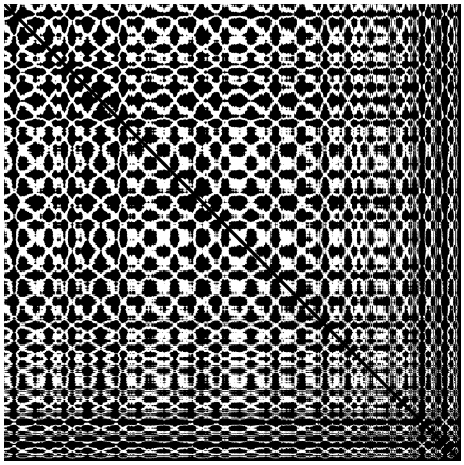


(b) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.3**.

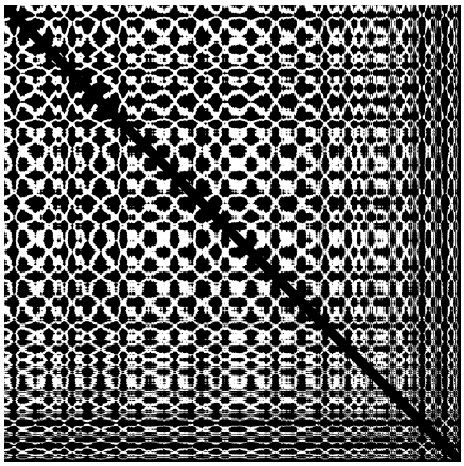


(c) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.7**.

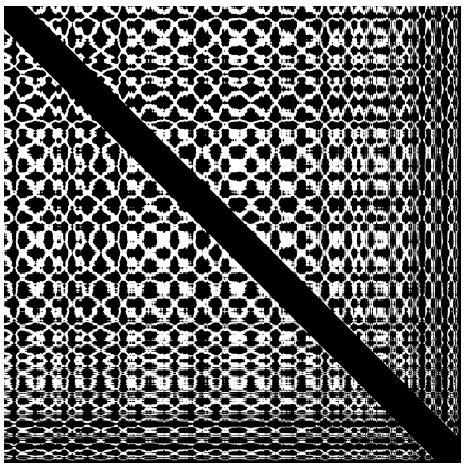
Figure 5.1: Effects of the threshold parameter p on the recurrence plot. As the value increases more pairs of points are taken as recurrences, disrupting a clear view of patterns or repetitive behaviors.



(a) m=3, $\tau = 20$, $\boldsymbol{w}$**=10**, p=0.2.



(b) m=3, $\tau = 20$, $\boldsymbol{w}$**=30**, p=0.2.



(c) m=3, $\tau = 20$, $\boldsymbol{w}$**=100**, p=0.2.

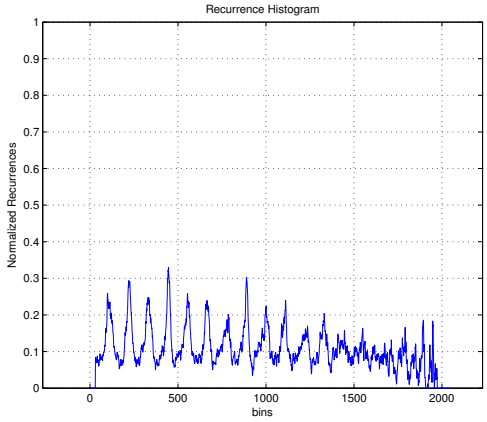
Figure 5.2: Effects of the Theiler window parameter w on the recurrence plot. As the value increases, more consecutive points are taken out of the analysis, creting a black diagonal line representing the non-taken pairs of points.

caption of each subfigure, using bold highlights on the changed parameters. Figure 5.4 shows that changes in τ and m do not considerably affect the shape of the recurrence time histogram. The only noticeable variations can be seen on high-valued bins, where recurrences are not taken into account, and in the average value of the recurrence histogram, which tends to decrease as m increases; both are due to the effect of $N - \eta$ as an upper limit value when creating the state-space. Figure 5.4 shows examples of recurrence time histograms where features described on section 3.3 are extracted from.

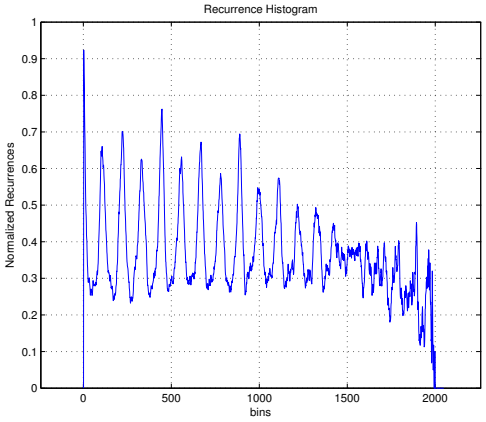
After visual inspection of the influence of different parameter values on several audio frames and based on the amount of useful information those parameters bring to the extracted features, the selected parameter values for the recurrence histogram and frequency histogram processes are the following:

- m: 3, 7, 12.
- τ: 1, 3, 7.
- p: 0.2, 0.3, 0.7.
- w: 30.

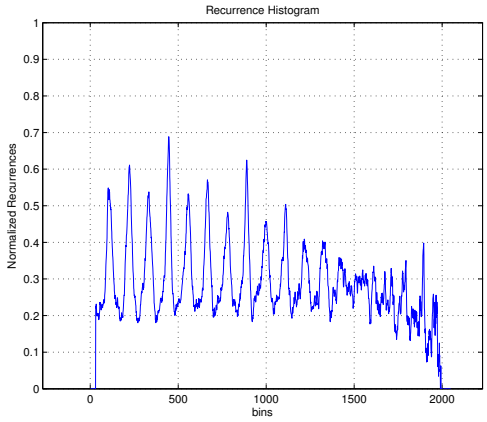
On the following sections, the effect of chosen parameter values on the classifiers training and testing will be shown. Classifiers are trained and tested in six different and independent ways: using features extracted from recurrence time histogram (H_t) only, from frequency time histogram (H_f) only, from the combination of these two, from the combination of each H_t and H_f features with the features obtained through common methodology (CM), and a complete combination of all H_t, H_f and common methodology features. The keywords used to identify each classifier have been defined on section 3.4. On every classification section, the shown results are the average of the 10 iterations of the 3-fold cross validation performed on each classifier.



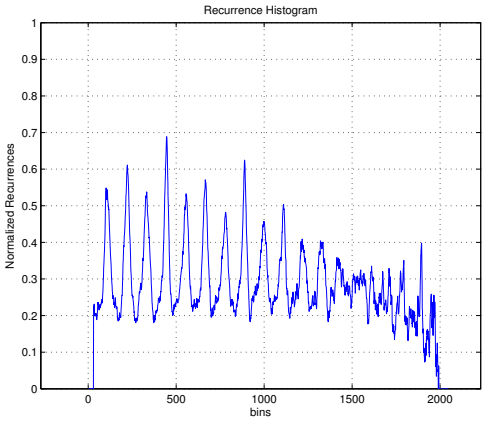
(a) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.05**.



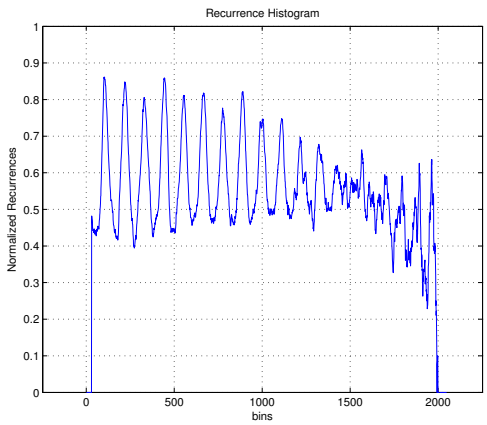
(b) m=3, $\tau = 20$, $\boldsymbol{w}$**=0**, p=0.2.



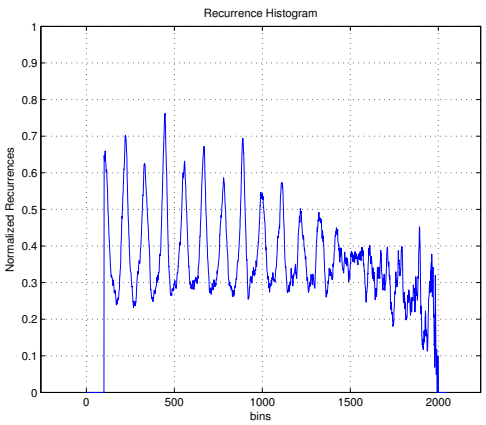
(c) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.2**.



(d) m=3, $\tau = 20$, $\boldsymbol{w}$**=30**, p=0.2.

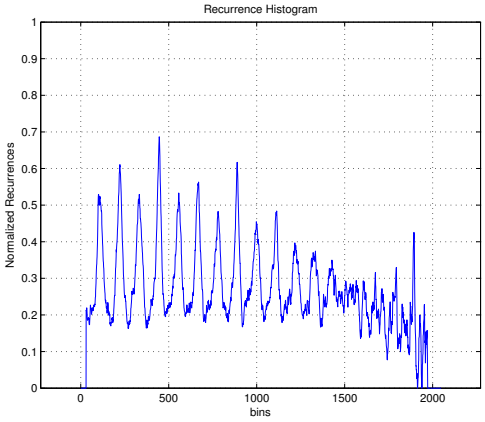


(e) m=3, $\tau = 20$, w=30, $\boldsymbol{p}$**=0.7**.

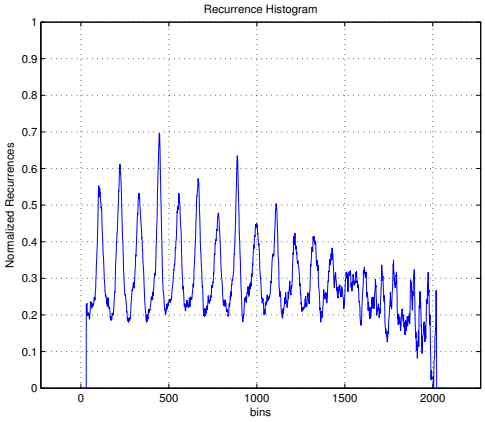


(f) m=3, $\tau = 20$, $\boldsymbol{w}$**=100**, p=0.2.

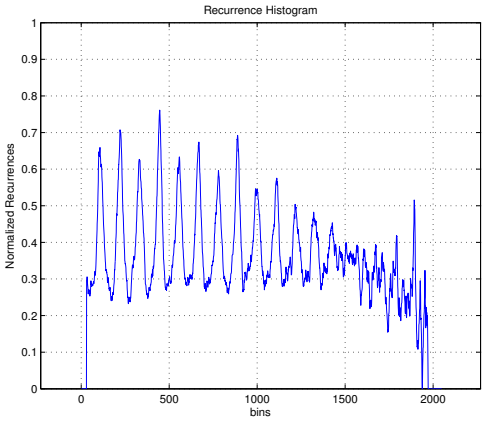
Figure 5.3: A comparison between normalized and non-normalized H_t can be seen on (a) and (b). The increment in parameter p brings an increment in the average value of H_t, caused by raising the distance threshold and taking more recurrences consequently. The increment in parameter w causes the rejection of close points from the analysis, resulting in elemination of recurrences on the first bins.



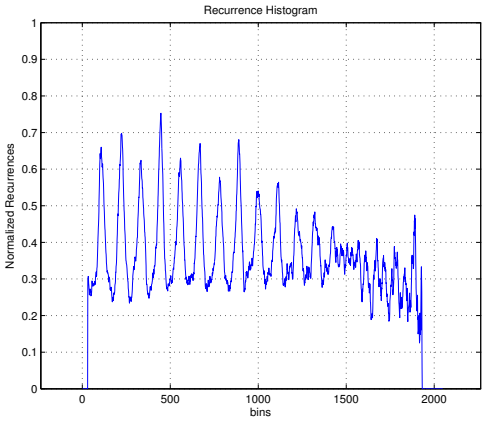
(a) $\boldsymbol{m}$**=4**, $\tau = 20$, w=30, p=0.2.



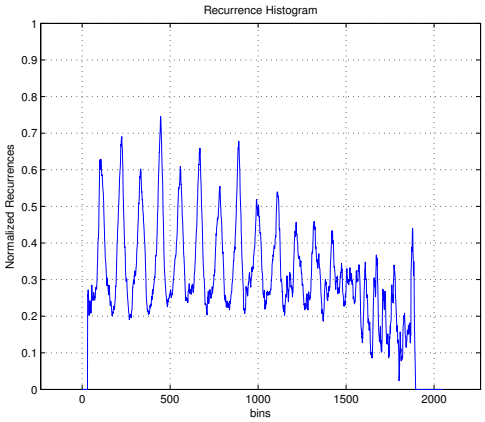
(b) m=3, $\tau =$ **7**, w=30, p=0.2.



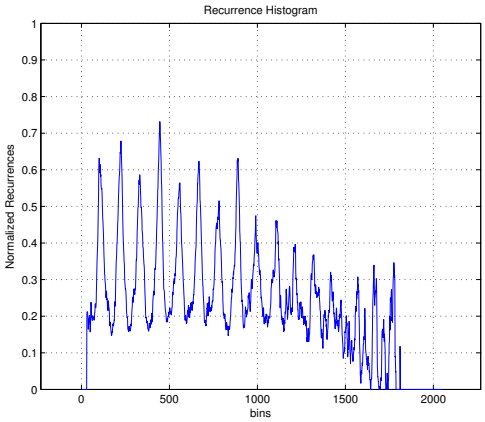
(c) $\boldsymbol{m}$**=7**, $\tau = 20$, w=30, p=0.2.



(d) m=3, $\tau =$ **30**, w=30, p=0.2.



(e) $\boldsymbol{m}$**=12**, $\tau = 20$, w=30, p=0.2.



(f) m=3, $\tau =$ **50**, w=30, p=0.2.

Figure 5.4: Effect of state-space parameter variations on H_t. When the embedding dimension m is increased, the average value of the H_t decreases, showing an analogous effect such as of incrementing p; this is caused by the limit of samples $N - \eta$ taken from the audio frame when creating the state-space. The increment of τ does not considerably affect the average value, but remains ignoring recurrences from the higher bins given the η samples not taken from the audio signal.

5.2 CM Classification

The accuracy of the trained classifiers using common methodology features is shown on table 5.1.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
-	-	-	10.0	23.5	56.4	50.4	62.00	62.3	63.2	66.0	66.0

Table 5.1: Accuracy results for common methodology classification.

5.3 H_t Features Classification

The accuracy of the trained classifiers using features extracted from the recurrence time histogram is shown in table 5.2. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	23.6	41.3	38.9	46.5	52.8	45.3	48.1	52.3
3	7	0.20	10.0	24.0	45.6	**48.1**	51.5	56.1	51.7	53.5	54.6
3	1	0.30	10.0	23.9	45.2	40.7	49.6	55.1	47.9	51.5	53.4
3	1	0.70	10.0	**24.8**	44.9	39.6	51.2	54.8	48.5	52.2	55.1
3	7	0.70	10.0	**24.5**	**49.5**	**49.9**	**51.7**	**59.4**	**54.9**	**57.5**	56.5
7	1	0.20	10.0	21.5	41.5	38.5	46.7	51.9	44.4	46.6	53.0
7	1	0.30	10.0	19.5	43.3	38.9	49.6	54.3	46.6	49.9	**55.5**
7	1	0.70	10.0	**24.0**	45.6	**48.1**	51.5	56.1	51.7	53.5	54.6
7	3	0.30	10.0	24.0	42.1	42.3	47.5	53.0	47.4	49.8	51.7
7	3	0.70	10.0	23.8	46.1	45.7	49.2	56.0	52.3	54.5	54.6
12	1	0.30	10.0	19.8	40.2	38.7	48.1	49.7	45.5	49.0	53.4
12	1	0.70	10.0	21.5	43.4	42.0	50.5	52.2	49.0	52.2	54.5
12	3	0.70	10.0	21.7	**46.2**	6.9	**51.9**	**56.7**	**53.3**	**55.5**	**57.4**
12	7	0.70	10.0	23.0	**47.7**	45.9	**53.4**	**56.8**	**55.9**	**57.5**	**57.0**

Table 5.2: Accuracy results for H_t features classification.

As can be seen, the overall percentage is reduced when comparing to the baseline values. Values drop between 11% and 17 % depending on the classifier. One should

note that the spectral features used on H_t are not designed to describe the behavior of this type of information, they are designed to describe the frequency spectrum. Consequently, the classification using the H_t features by themselves could be expected not to be as accurate as the baseline classification. Another tendency is the increment of classification accuracy when m is increased. As seen on figure 5.4b the increment on this parameter leads to a reduction in the number of compared points, reducing the average value of the H_t. this leads to a better definition of the H_t, without rejecting too much information. The increment of p leads to a better classification on low values of m by increasing between 1 and 2%. The increment of τ in small steps insignificantly reduces the classification acurracy.

5.4 H_f Features Classification

The accuracy of the trained classifiers using features extracted from the recurrence frequency histogram is shown in table 5.3. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	21.7	**43.0**	**40.2**	46.5	49.0	**45.9**	**48.9**	**49.5**
3	7	0.20	10.0	**23.7**	38.1	34.2	42.6	46.2	41.8	44.2	47.2
3	1	0.30	10.0	21.0	42.8	40.2	**47.0**	**50.6**	**45.2**	**48.7**	**50.3**
3	1	0.70	10.0	21.2	**44.4**	**41.8**	**46.8**	**51.6**	**47.3**	**51.2**	**50.7**
3	7	0.70	10.0	**23.7**	39.8	34.9	43.0	45.5	38.1	41.1	46.9
7	1	0.20	10.0	21.8	41.8	38.1	44.4	49.2	44.8	47.1	47.2
7	1	0.30	10.0	20.9	**43.4**	38.8	45.2	48.5	45.0	47.6	47.6
7	1	0.70	10.0	21.5	38.3	34.2	44.3	45.8	39.4	42.7	48.4
7	3	0.30	10.0	21.2	37.2	36.4	42.6	44.8	41.0	43.7	45.7
7	3	0.70	10.0	17.9	38.2	34.2	42.3	45.7	40.6	43.6	45.5
12	1	0.30	10.0	20.1	40.7	35.1	42.3	45.3	42.4	44.9	46.6
12	1	0.70	10.0	20.2	41.4	37.6	45.0	47.4	43.1	46.1	48.7
12	3	0.70	10.0	**21.8**	34.6	30.2	39.5	41.7	36.9	39.6	41.9
12	7	0.70	10.0	21.3	36.0	34.8	40.9	44.9	39.0	42.2	44.1

Table 5.3: Accuracy results for H_f features classification.

On this case, the overall accuracy with respect to the baseline is reduced by 12% on **SL** and almost 16% on classifiers such as **SVP** and **SVR** when compared to the H_t accuracies. Contrary to what can be seen on the previous section, the increasing m reduces the accuracy of this type of classification, making the worst H_f feature-based classification the best H_t feature-based classification for almost all classifiers. The effect of p and τ is similar than in H_t feature-based classification. The recurrence frequency histogram compresses the sample lags into very delimited frequency regions, which might cause a loss of information compared to the FFT.

5.5 H_t + H_f Features Classification

The accuracy of the trained classifiers using a combination of features from the recurrence time histogram and from the recurrence frequency histogram is shown in table 5.4. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	23.1	46.6	44.8	50.7	57.1	50.5	53.0	55.4
3	7	0.20	10.0	24.4	47.5	**48.2**	51.3	56.8	51.2	54.2	54.8
3	1	0.30	10.0	24.6	46.8	43.6	50.6	**58.4**	51.3	54.0	56.3
3	1	0.70	10.0	**25.8**	47.1	45.4	**52.9**	56.8	53.4	**55.9**	**57.1**
3	7	0.70	10.0	23.5	**48.8**	47.6	51.4	**58.0**	**53.7**	**57.0**	55.0
7	1	0.20	10.0	24.4	45.0	42.6	48.0	55.6	47.8	49.8	53.5
7	1	0.30	10.0	24.2	43.8	42.9	50.6	55.7	48.4	50.7	54.5
7	1	0.70	10.0	23.8	45.1	43.2	50.6	55.9	52.1	54.3	55.2
7	3	0.30	10.0	21.1	38.4	39.3	43.2	45.8	42.2	44.2	47.2
7	3	0.70	10.0	**24.9**	47.1	45.9	50.1	56.6	52.0	55.1	54.1
12	1	0.30	10.0	23.2	45.3	44.6	49.5	55.4	48.5	52.0	55.1
12	1	0.70	10.0	22.9	43.2	42.6	48.4	53.6	47.3	49.5	52.6
12	3	0.70	10.0	**25.7**	**48.4**	**47.4**	**52.5**	57.5	**54.0**	55.5	**57.3**
12	7	0.70	10.0	23.5	**49.8**	**49.1**	**53.8**	**59.3**	**56.0**	**58.5**	**58.0**

Table 5.4: Accuracy results for H_t + H_f features classification.

The accuracy percentages of this type of classification does not reach the baseline percentages, but are above H_f classification by 5% on low m and p values and almost

16% on high m values; they are also above H_t classification by 1% on high m and p values, while reaching an increment around 5% on low m values. While selecting the features by means of the Attribute Selection, an average of 74% of the selected attributes is provided by H_t features, while the remaining 26% is provided by H_f features.

5.6 CM + H_t Features Classification

The accuracy of the trained classifiers using a combination of common methodology features and time recurrence histogram features is shown in table 5.5. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	**28.0**	58.2	55.3	65.6	67.2	65. 3	68.1	**70.3**
3	7	0.20	10.0	26.0	59.8	55.4	**66.7**	**67.2**	**66.4**	68.0	69.6
3	1	0.30	10.0	**27.0**	58.1	54.5	65.6	66.29	64.5	67.5	69.7
3	1	0.70	10.0	26.8	57.8	55.3	65.2	65.8	65.1	68.3	69.5
3	7	0.70	10.0	24.9	60.0	**57.6**	**65.9**	**67.1**	66.0	**69.7**	68.7
7	1	0.20	10.0	25.1	57.6	53.0	64.7	65.1	63.9	67.0	68.6
7	1	0.30	10.0	26.6	57.4	53.2	65.9	65.6	64.7	67.1	**70.1**
7	1	0.70	10.0	26.8	57.1	52.9	64.5	65.5	63.6	67.3	68.8
7	3	0.30	10.0	24.8	59.3	54.6	64.6	64.8	64.6	67.6	68.9
7	3	0.70	10.0	24.7	**60.8**	55.6	64.9	65.8	64.9	68.4	69.4
12	1	0.30	10.0	**26.9**	54.5	50.6	64.3	63.7	61.0	64.7	67.8
12	1	0.70	10.0	25.7	57.0	52.6	64.9	64.5	61.9	66.4	68.4
12	3	0.70	10.0	26.5	**61.9**	**55.8**	65.5	66.7	**66.9**	**69.5**	69.1
12	7	0.70	10.0	26.0	**62.2**	**57.8**	**67.7**	**69.0**	**68.5**	**70.9**	**70.5**

Table 5.5: Accuracy results for CM + H_t features classification.

The accuracy percentage increments by almost 5% on **MP**, **Forest**, **SVR** and **SL** compared to the baseline percentages when high values of m are used. This is due to the combination of the best H_t features and baseline features to train and test the classifier. When selecting these features on WEKA by means of the Selection, described on section 3.4, an average of 60% of the selected attributes is provided

CM baseline, while the other 40% is provided by H_t extracted features. This means H_t features help the baseline features to increase the classification accuracy.

5.7 CM + H_f Features Classification

The accuracy of the trained classifiers using a combination of common methodology features and recurrence frequency histogram features is shown in table 5.6. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	24.9	57.5	50.1	61.6	65.2	62.4	65.8	**66.4**
3	7	0.20	10.0	**26.9**	56.2	49.6	61.5	64.8	61.4	64.6	64.5
3	1	0.30	10.0	25.6	58.2	49.2	60.9	**65.3**	62.8	**66.6**	65.8
3	1	0.70	10.0	25.8	**58.4**	**53.0**	**62.3**	64.2	**63.7**	**67.1**	66.0
3	7	0.70	10.0	**28.2**	56.8	49.5	61.4	**65.3**	62.5	65.7	65.3
7	1	0.20	10.0	**26.8**	58.2	51.1	61.9	64.3	62.3	65.9	65.8
7	1	0.30	10.0	26.1	57.7	**52.9**	62.2	64.3	**63.1**	66.5	**66.7**
7	1	0.70	10.0	25.2	**58.9**	51.9	61.7	**65.1**	61.7	64.1	64.4
7	3	0.30	10.0	25.6	58.0	**52.5**	**64.1**	64.7	62.6	66.0	**66.6**
7	3	0.70	10.0	24.0	55.6	50.1	61.8	64.5	61.1	64.6	64.8
12	1	0.30	10.0	26.4	56.4	51.2	61.2	64.2	61.8	65.5	65.7
12	1	0.70	10.0	25.1	**58.3**	50.4	61.8	64.8	62.9	66.1	65.3
12	3	0.70	10.0	25.6	54.6	49.8	61.1	62.9	59.6	62.4	63.7
12	7	0.70	10.0	24.0	57.6	51.8	**63.1**	63.9	**63.6**	**67.1**	66.2

Table 5.6: Accuracy results for CM + H_f features classification.

In this case, the accuracy percentage increases by 1% or less on **MP**, **Forest** and **SVP** , and by 2% on **SVR**. When selecting the features by means of the Attribute Selection in WEKA, an average of 72% of the selected attributes is provided by CM features, while the other 28% is provided by H_f extracted features. It can be seen that the effect of the H_f features is not a big impact on the classification accuracies, given the fact that H_f features have lower classification results when used by themselves, translating to lack of effective information on these features. It must be considered as well the high frequency random behavior eliminated by the Theiler

correction window parameter w, which completely erases unrelevant information from the FH representation of the audio signal.

5.8 Baseline + H_t + H_f Features Classification

The accuracy of the trained classifiers using a combination of all features from baseline, recurrence histogram and frequency histogram is shown in table 5.7. Bold highlights indicate the top three classification accuracies.

m	τ	p	0R	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
3	1	0.20	10.0	27.0	59.2	54.8	**66.7**	66.4	65.2	68.6	69.5
3	7	0.20	10.0	24.2	61.4	56.3	65.9	67.5	65.1	67.7	69.5
3	1	0.30	10.0	**28.6**	59.5	54.8	66.5	67.6	65.8	68.9	69.5
3	1	0.70	10.0	26.2	**63.1**	**57.2**	65.8	**68.5**	64.9	68.3	69.1
3	7	0.70	10.0	26.0	61.1	**58.4**	66.1	**68.1**	**67.4**	**70.7**	**70.1**
7	1	0.20	10.0	26.5	60.7	52.2	65.3	66.4	65.2	68.3	67.9
7	1	0.30	10.0	26.8	60.8	55.4	**67.1**	66.7	66.4	69.4	**70.1**
7	1	0.70	10.0	26.8	59.4	54.4	65.3	67.0	65.0	68.6	68.4
7	3	0.30	10.0	**27.7**	60.8	54.3	66.4	66.7	**67.1**	**70.0**	69.6
7	3	0.70	10.0	27.2	60.6	54.9	64.9	67.0	64.8	68.5	68.8
12	1	0.30	10.0	**27.5**	58.5	52.1	65.9	65.2	63.4	66.4	67.9
12	1	0.70	10.0	27.2	58.0	52.2	65.0	65.0	62.4	66.6	67.7
12	3	0.70	10.0	26.2	**63.2**	53.8	64.1	64.9	61.8	66.0	67.1
12	7	0.70	10.0	26.3	**63.8**	**58.6**	**66.6**	**68.6**	**69.9**	**71.5**	**69.8**

Table 5.7: Accuracy results for Baseline + H_t + H_f.

The percentages shown are not too different from the percentages from section 5.6, changing in less than 1%. From the Attribute Selection, an average of 53% of the selected features is provided by CM features, 36% is provided by H_t features and 11% is provided by H_f features. It is clear that the influence of H_f features in this case brings no consistent change in the accuracy of the classifiers.

The highest classification accuracy is achieved by the combination of all features, resulting in a 71.5% for the Support Vector Machines classifier using a Radial Basis Function as kernel. This represents a 5.5% increment above the highest common

methodology classification, given by the same classifier.

5.9 Summary

A final table gathering the best classification accuracies is shown. Numbers in bold on table 5.8 represent the top three percentages of each classifier above CM classification accuracy. From the same table, it can be seen that the best classifications are achieved when combining the features extracted from H_t and/or H_f with the CM features rather than by themselves. When combining features from H_t with features from H_f, the classification accuracy does not reach the baseline percentage. In terms of parameters, a common value for high accuracies is p=0.7, which indicates the selection of an elevated number of recurrences due to a high threshold on the recurrence plot. Another common parameter value on high accuracies is either a low embedding dimension (m=3) or a very high value of it (m=12). Since low values of τ are used for the whole analysis, nothing concrete can be developed as how it influences the classification on higher values.

Features	m	τ	p	1R	IBk	Bayes	MP	Forest	SVP	SVR	SL
CM	-	-	-	23.5	56.4	50.4	62.0	62.3	63.3	66.0	66.0
H_t	3	7	0.70	24.5	49.5	49.9	51.7	59.4	54.9	57.5	56.5
H_t	12	7	0.70	23.0	47.6	45.9	53.4	56.8	55.9	57.5	57.0
H_f	3	1	0.30	21.0	42.8	40.2	47.0	50.6	45.2	48.7	50.3
H_f	3	1	0.70	21.2	44.4	41.8	46.8	51.6	47.3	51.2	50.7
H_t+H_f	12	3	0.70	25.7	48.4	47.4	52.5	57.6	54.0	55.5	57.3
H_t+H_f	12	7	0.70	23.5	49.8	49.1	53.8	59.3	56.0	58.5	58.0
CM+H_t	3	7	0.70	24.9	60.0	57.6	65.9	67.1	66.0	69.7	68.7
CM+H_t	12	7	0.70	**26.0**	**62.2**	**57.8**	**67.7**	**69.0**	**68.5**	**70.9**	**70.5**
CM+H_f	3	1	0.70	25.8	58.4	53.0	62.3	64.2	63.7	67.1	66.0
CM+H_f	7	1	0.30	**26.1**	57.7	52.9	62.2	64.3	63.0	66.5	66.7
CM+H_f	12	7	0.70	24.0	57.6	51.8	63.1	63.9	63.6	67.1	66.2
ALL	3	7	0.70	26.0	**61.1**	**58.4**	**66.0**	**68.1**	**67.4**	**70.7**	**70.1**
ALL	12	7	0.70	**26.3**	**63.8**	**58.6**	**66.6**	**68.6**	**68.9**	**71.5**	**69.8**

Table 5.8: Summary of the best classification accuracies for a given set of extracted features and parameter combination.

Chapter 6

Conclusions

This work was focused on the development of new information sources for music classification based on nonlinear recurrence analysis tools. The *Recurrence Time Histogram* is obtained by transforming the matrix output of a nonlinear time series analysis technique known as Recurrence Plot into a histogram, where each bin represents the sample delay between recurrent states. The *Recurrence Frequency Histogram* comes after a frequency fitting from the inverse of the recurrence histogram sample delays to the frequency binning of the frequency spectrum, obtained by means of the FFT. These histograms are obtained from segments of audio signal called frames, which have a fixed length and fixed analysis parameters that are kept the same for all the songs from a given database; this gives information about how the variation of parameters affect the outcome of both histograms, and settles a reference for future classification analysis.

The frequency adjustment from the recurrence time histogram and the low density of high frequency bins, derive in low frequency continuity and high frequency peaks on the recurrence frequency histogram. The distribution of the peaks by adding a small random value and the consequent normalization bring an increment on high frequency values, which are eliminated by an extended Theiler correction window. This action will remove apparently unrelevant information above a certain frequency value.

Since the objective of this research is to provide the tools where features for music classification could be extracted from, the same spectral features used to obtain information from the frequency spectrum are used on both recurrence time histogram and recurrence frequency histogram for classification purposes, which include classifier training and testing.

The combination of common methodology features and histogram features resulted in a higher classification accuracy than the common methodology and individual histogram classifications. Features extracted from the recurrence frequency histogram in combination with common methodology features provide an increased classification accuracy, but when combined with recurrence time histogram features they do not reflect a consistent change in the classification accuracy. This supports the idea that the increment in classification accuracies does not come from providing more data about the audio signal, but from providing new and reliable information about it. However, it can be seen that the nonlinear audio recurrence analysis provides additional information that supports the common classification methodology by increasing its accuracy percentage. The best classification resulted in a 5.5% accuracy increment in the highest common methodology acurracy, raising it from 66.0% using common methodology to 71.5%.

In terms of parameters, the highest classifications on each combination of features came from high embedding dimensions, high percentages of the distance matrix mean and considerably low time delay values. Another combination that provided high accuracies was a low embedding dimension, low percentage of the distance matrix and a small time delay. Both of these combinations result in a better definition of the recurrence time histogram, whether by not comparing pairs of points on the state-space trajectory or by taking less recurrences due to threshold restrictions respectively. For a single classifier, the percentage of accuracy varies between 3% and 6%, making the combination of adequate parameters and the best combination of extracted features the key factors for an improved classification.

6.1 Future Work

This research has shown interesting results regarding the possibility of analyzing raw audio signals by nonlinear time series analysis techniques; some of these ideas can be developed in the future for extended applications or by increasing depth in specific sections of this work. Some points that can be addressed as extentions or complements of this thesis are the following:

- **Validation of parameters using a different database**: the use of a different database than the one used in this research is a viable alternative to verify the increment of classification accuracy with a given combination of parameters. To do so, the whole methodology must be applied on this new database, beginning with common methodology classification. A database with a similar number of genres but including complete songs is suggested.

- **Development of specific histogram features**: since the extracted features from both histograms are designed for frequency spectra, and even though the frequency histogram has the same frequency reference as the correspondent spectrum, the methodology by which these are extracted may not fit correctly in decribing its behavior. The creation of features implemented specifically for these histograms is a promising option to extract more meaningful information from the sources here provided.

- **Alternative frequency fitting reference**: the frequency histogram is created with the Fourier spectrum as reference for its frequency values. Because of this, the information extracted from the recurrence histogram appears compressed on the frequency histogram, creating peaks and non-continuous behavior on certain frequency bins. The frequency reference could be changed in order to spread the information on the frequency histogram and extract better information from it. As a consequence, the development of features specifically made for frequency histograms will be required.

- **Frequency fitting without random values**: as seen before, the distribution of the frequency histogram relies on adding random values for the fitting into equally-spaced frequency bins, which leads to the elimination of high frequency values. An alternative in the fitting of these values can be considered to provide more information on the frequency histogram and increase the reliability of the extracted features.

- **Maximization of the recurrence analysis parameters**: the embedding dimension, the time delay, and the distance matrix threshold control a considerable variation of the classification accuracy for a given classifier under certain combination of features. A maximization of these parameters might bring a better accuracy when combining baseline features with recurrence histogram features or when combining all features together; a special emphasis is done on this two cases given that they provided the highest percentages on this research.

Bibliography

[1] Jean-Julien Aucouturier and François Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.

[2] Jean-Julien Aucouturier, François Pachet, and Mark Sandler. "the way it sounds": Timbre models for analysis and retrieval of music signals. *IEEE Transactions on Multimedia*, 7(6):1–8, 2005.

[3] Michael Casey, Remco Veltkamp, and Masataka Goto. Content-based music information retrieval: Current directions and future chanllenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.

[4] Gary Davis and Ralph Jones. *The Sound Reinforcement Handbook*, chapter 1. Hal Leonard Publishing, 1989.

[5] David Gerhard. Audio visualization in phase space. Technical report, Simon Fraser University, Canada, 1999.

[6] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.

[7] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *AES 25th International Conference*, number 6-2, June 2004.

[8] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reute-

mann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[9] Jesper Hojvang Jensen. *Feature Extraction for Music Information Retrieval*. PhD thesis, Aalborg University, Deparment of Electronic Systems, 2010.

[10] Holger Kantz and Thomas Schreiber. *Nonlinear Time Series Analysis*, chapter 2. University Press, 1997.

[11] Matthwe B. Kennel, Reggie Brown, and Henry D. I. Abarbanel. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *The American Physical Society Journal*, 45(6):3403–3411, 1992.

[12] Olivier Lartillot. *MIR toolbox manual*. University of Jyväskylä, Finland, June 2009.

[13] Olivier Lartillot and Petri Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International Conference on Digital Audio Effects*, pages 237–244, September 2007.

[14] Norbert Marwan, Carmen Romano, Marco Thiel, and Jürgen Kurths. Recurrence plots for the analysis of complex systems. *Physics Reports*, 438(5-6): 237–329, 2006.

[15] Ingo Mierswa and Katharina Morik. Automatic feature extraction for classifying audio data. *Machine Learning*, 58(2-3):127–149, 2005.

[16] Fabian Moerchen, Ingo Mierswa, and Alfred Ultsch. Understandable models of music collections based on exhaustive feature generation with temporal statistics. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 882–891, 2006.

[17] Nicola Orio. Music retrieval: A tutorial and review. *Foundations and Trends in Information Retrieval*, 1(1):1–90, November 2006.

[18] Steffen Pauws. Musical key extraction from audio. Technical report, Philips Research Laboratories Eindhoven, 2004.

[19] Geoffroy Peeters. A large set of audio features for sound description in the cuidado project. Technical report, IRCAM, 2004.

[20] Pietro Polotti and Davide Rocchesso. *Sound to Sense, Sense to Sound. A State of the Art in Sound and Music Computing*, chapter 3. Logos Verlag, Berlin, 2008.

[21] Michael J. Roberts. *Signals and Systems*, chapter 7. McGraw Hill, 2004.

[22] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content. *IEEE Signal Processing Magazine*, 23(2):133–141, March 2006.

[23] Thomas Schreiber. Interdisciplinary application of nonlinear time series methods. *Physics Reports*, 308(1):1–64, 1999.

[24] Joan Serrà and Emilia Gómez. Audio cover song identification based on tonal sequence alignment. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 61 – 64, 2008.

[25] Joan Serrà, Xavier Serra, and Ralph Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(093017):1–20, September 2009.

[26] Xavier Serra. *A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition*. PhD thesis, Stanford University, October 1989.

[27] Sigurdur Sigurdsson, Kaare Brandt Petersen, and Tue Lehn-Schioler. Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. Technical report, Technical University of Denmark, 2006.

[28] Agilent Technologies. *Principal Components Analysis Manual*, 2005.

[29] Dmitry Terez. Robust pitch determination using nonlinear state-space embedding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, pages 345–348, 2002.

[30] Dmitry Terez. Methods and apparatus for pitch determination. United States Pattent Application Publication, Millville, NJ. USA., May 2002.

[31] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[32] Ian Witten and Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. Elsevier, second edition, 2005.

[33] Ho-Hsiang Wu and Juan P. Bello. Audio-bases music visualization form music structure analysis. In *SMC Conference 2010*, number 73, 2010.

[34] Hitten Zaveri, Ivan Osorio, Mark G. Frei, and Susan Arthurs, editors. *Epilepsy: The Intersection of Neurosciences, Biology, Mathematics, Physics and Engineering*, chapter contributed by Ralph G. Andrzejak. CRC Press, In Press.

[35] Udo Zölzer. *Digital Audio Signal Processing*, chapter 3. John Wiley and Sons, 1998.

[36] Udo Zölzer, editor. *DAFX: Digital Audio Effects*, chapter 1. John Wiley and Sons, 2007.