Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Features for the Classification and Clustering of Music in Symbolic Format

Alexandre Miguel Entradas Bernardo

Mestrado em Engenharia Informática

2008

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática

Features for the Classification and Clustering of Music in Symbolic Format

Alexandre Miguel Entradas Bernardo

DISSERTAÇÃO

Dissertação orientada pelo Prof. Dr. Thibault Langlois

Mestrado em Engenharia Informática

2008

Resumo

Este documento descreve o trabalho realizado no âmbito da disciplina de Projecto em Engenharia Informática do Mestrado em Engenharia Informática da Faculdade de Ciências da Universidade de Lisboa.

Recuperação de Informação Musical é, hoje em dia, um ramo altamente activo de investigação e desenvolvimento na área de ciência da computação, e incide em diversos tópicos, incluindo a classificação musical por géneros.

O trabalho apresentado centra-se na Classificação de Pistas e de Géneros de música armazenada usando o formato MIDI.

Para resolver o problema da classificação de pistas MIDI, extraímos um conjunto de descritores que são usados para treinar um classificador implementado através de uma técnica de Máquinas de Aprendizagem, Redes Neuronais, com base nas notas, e durações destas, que descrevem cada faixa.

As faixas são classificadas em seis categorias: Melody (Melodia), Harmony (Harmonia), Bass (Baixo) e Drums (Bateria).

Para caracterizar o conteúdo musical de cada faixa, um vector de descritores numérico, normalmente conhecido como "shallow structure description", é extraído. Em seguida, eles são utilizados no classificador — Neural Network — que foi implementado no ambiente Matlab.

Na Classificação por Géneros, duas propostas foram usadas: Modelação de Linguagem, na qual uma matriz de transição de probabilidades é criada para cada tipo de pista midi (Melodia, Harmonia, Baixo e Bateria) e também para cada género; e Redes Neuronais, em que um vector de descritores numéricos é extraído de cada pista, e é processado num Classificador baseado numa Rede Neuronal.

Seis Colectâneas de Musica no formato Midi, de seis géneros diferentes, Blues, Country, Jazz, Metal, Punk e Rock, foram formadas para efectuar as experiências. Estes géneros foram escolhidos por partilharem os mesmos instrumentos, na sua maioria, como por exemplo, baixo, bateria, piano ou guitarra. Estes géneros também partilham algumas características entre si, para que a classificação não seja trivial, e para que a robustez dos classificadores seja testada.

As experiências de Classificação de Pistas Midi, nas quais foram testados, numa primeira abordagem, todos os descritores, e numa segunda abordagem, os melhores descritores, mostrando que o uso de todos os descritores é uma abordagem errada, uma vez que existem descritores que confundem o classificador. Provou-se que a melhor maneira, neste contexto, de se classificar estas faixas MIDI é utilizar descritores cuidadosamente seleccionados.

As experiências de Classificação por Géneros, mostraram que os Classificadores por Instrumentos (Single-Instrument) obtiveram os melhores resultados. Quatro géneros, Jazz, Country, Metal e Punk, obtiveram resultados de classificação com sucesso acima dos 80%

O trabalho futuro inclui: algoritmos genéticos para a selecção de melhores descritores; estruturar pistas e musicas; fundir todos os classificadores desenvolvidos num único classificador.

PALAVRAS-CHAVE:
Classificação de Pistas MIDI, Recuperação de Informação Musical, Classificação por Géneros Redes Neuronais, Modelação de Linguagem.

Abstract

This document describes the work carried out under the discipline of Computing Engineering Project of the Computer Engineering Master, Sciences Faculty of the Lisbon University.

Music Information Retrieval is, nowadays, a highly active branch of research and development in the computer science field, and focuses several topics, including music genre classification.

The work presented in this paper focus on Track and Genre Classification of music stored using MIDI format,

To address the problem of MIDI track classification, we extract a set of descriptors that are used to train a classifier implemented by a Neural Network, based on the pitch levels and durations that describe each track. Tracks are classified into four classes: Melody, Harmony, Bass and Drums. In order to characterize the musical content from each track, a vector of numeric descriptors, normally known as shallow structure description, is extracted. Then they are used as inputs for the classifier which was implemented in the Matlab environment.

In the Genre Classification task, two approaches are used: Language Modeling, in which a transition probabilities matrix is created for each type of track (Melody, Harmony, Bass and Drums) and also for each genre; and an approach based on Neural Networks, where a vector of numeric descriptors is extracted from each track (Melody, Harmony, Bass and Drums) and fed to a Neural Network Classifier.

Six MIDI Music Corpora were assembled for the experiments, from six different genres, Blues, Country, Jazz, Metal, Punk and Rock. These genres were selected because all of them have the same base instruments, such as bass, drums, piano or guitar. Also, the genres chosen share some characteristics between them, so that the classification isn't trivial, and tests the classifiers robustness.

Track Classification experiments using all descriptors and best descriptors were made, showing that using all descriptors is a wrong approach, as there are descriptors which confuse the classifier. Using carefully selected descriptors proved to be the best way to classify these MIDI tracks.

Genre Classification experiments showed that the Single-Instrument Classifiers achieved the best results. Four genres achieved higher than 80% success rates: Jazz, Country, Metal and Punk.

Future work includes: genetic algorithms; structurize tracks and songs; merge all presented classifiers into one full Automatic Genre Classification System.

KEYWORDS:
MIDI Track Classification, Genre Classification, Music Information Retrieval, Neural Networks, Machine Learning, Language Modeling.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Music Information Retrieval is, nowadays, a highly active branch of research and development in the computer science field, and focuses several topics such as beat tracking, music genre classification, melody extraction, score-following to name a few.

There are a lot of known applications that use this technology for some extent: the new generation media players, that organize music in an intelligent way, based in the music itself, and generates, for example, dynamic playlists; Internet radio stations, which builds a playlist based on the user's taste; score following; finding similarities between songs in a large database.

The work presented in this paper focus on music stored using MIDI format. Electronic instruments use this format to communicate and synchronize themselves. The format consists in a number of tracks were each track represent the sequence of notes (pitch level and duration) played by one instrument. MIDI files also contain some metadata such as the instrumentation or key. One of the advantage of the MIDI format is its compactness. Many musical resources using this format are freely available on the Internet.

Previous work in Music Information Retrieval using MIDI format includes music genre detection where several approaches have been proposed. Some researchers use similarity measures based on Kolmogorov complexity estimates in conjunction with a classical Machine Learning technique like k-Nearest Neighbors [15], Support Vector Machines [10] or clustering [4]. Cataltepe [3] compares the performance of obtained with the Normalized Compression Distance approach on MIDI and audio files with the ad-hoc features extraction and Machine Learning approach proposed by McKay [12].

[7] developed a technique called Bayesian Aggregation, which uses the output predictions of different classifiers and aggregates them in such a way as to take advantage of the hierarchical nature of the predictions to improve classification accuracy.

Other researchers proposed to extract a set high level features from MIDI files and perform a genre classification using Neural Networks [12] [8] or Support Vector Machines in conjunction with dimensionality reduction techniques [9]. Basili [1] made a comparison of various Machine Learning techniques on a musical genre classification task.

Another approach is to perform automatic melody detection. Rizo et al. [5] [6] has proposed a set of features to characterize each MIDI track and used a Random Forest classifier to identify tracks that contain melody. In [11], an information-theoretic complexity measure and an estimate of the local entropy are used to recognize melody tracks.

A different approach, proposed by [13], is to apply text classification techniques, such as Machine Learning and Pattern Recognition, establishing a music equivalent to the words in texts.

1.1 Objectives

In this work we address the problem of MIDI track and genre classification. Midi track classification consists in devising a system able to assign an instrument (a class) to a track. Two Machine Learning approaches, Neural Networks and k-Nearest Neighbors, are used for classifying each track into four classes: Melody, Harmony, Bass and Drums.

Genre classification addresses the problem of classifying a midi song into a genre. Six genres are defined: Blues, Country, Jazz, Metal, Punk and Rock. Two approaches are used: Neural Network and Language Modeling. The Neural Network approach makes use of machine learning techniques and numerical vectors known as shallow structure description. Language Modeling addresses the problem through transition probability matrices.

1.2 Document organization

- MIDI symbolic representation
- Musical Genre definition
- Methodology
  - Track classification
    - MIDI track description
    - Classifiers
    - Track selection

  - Genre classification
    - Language Modeling Approach
      Single-Instrument Beat Similarity
      Single-Instrument Shallow Structure Beat Similarity
      Multi-Instrument Shallow Structure Beat Similarity
    - Neural Network Approach

- Experiments

  Music Corpora

  Track Classification Experiments

  Genre Classification Experiments

  Approaches Comparison

- Future Work

- Conclusion

1.3 Work Plan

Begin	End	Description
October 2007	October 2007	Bibliography research
November 2007	January 2008	Feature comparative study on musical genres
February 2008	May 2008	Prototype Development
June 2008	June 2008	Thesis writing and Results divulgation

Table 1.1: Work Plan

Chapter 2

MIDI symbolic representation

MIDI (Musical Instrument Digital Interface) is an industry-standard communication technology which connects electronic musical instruments and electronic equipments, enabling a musical piece to be executed, transmitted or manipulated by any equipment which recognizes the MIDI standard. Technically, MIDI is a protocol, but it is generally used as a term to designate the different system components, such as adaptors, cables, files, etc.

Unlike other formats, such as Wav or MP3, a MIDI file doesn't contain any audio, but "event messages" such as pitch, velocity, volume, vibrato, panning, etc, so that a MIDI capable equipment can interpret the MIDI file and reproduce it.

The MIDI format has several possible ways to organize each track: all the instruments on the same MIDI track but on different MIDI channels, and unfortunately, there is no real standard, because there are numerous ways, MIDI sequencers, to create a MIDI song, and each MIDI sequencer may create a MIDI file in a different way. This can lead to several problems when interpreting the MIDI. Rosegarden, an open-source MIDI sequencer, was used to normalize all the MIDI files used in the experiments, so that all share the same structure.

The Midi format was chosen, over audio, for several reasons: smaller file size; instrument, pitch, velocity information (among others) is an integral part of the Midi file; several Internet databases, from various genres, provide free Midi files. Audio has other advantages, such as: energy; original audio material (instead of a Midi symbolic representation) - but for the proposed tasks, Midi features more benefits.

It is important to note that, contrasting to many other previously published studies, our approach does not use any metadata present in the MIDI file (such as instrumentation).

Chapter 3

Musical Genre definition

A musical genre can be thought of as a category to which musical sounds (songs) belong, given the fact that all songs in that particular category share some common elements, such as the techniques, the styles, the context, the themes (content, spirit) and geographical origin. For example, baroque is defined chronologically, Indian is defined geographically, Math-rock is defined mainly by technique and complexity, Post-rock is a genre defined by Simon Reynolds, a music critique, and Pop is only used for commercial reasons. Grouping musical pieces into genres is not a straightforward process, mainly because it relies in an individual personal understanding and musical knowledge.

In this work, effort has been made to assemble various musical pieces which are universally representative of a given genre. It's universally accepted that John Coltrane's "Bluebird" is Jazz, and that Bill Haley's "Rock Around the Clock" is Rock'n'Roll. As such, all Midi files collected, are representative of its genre.

One could argue that rock is a ramification of blues, and that punk is a ramification of rock. It's true, but nevertheless, each of the genres assembled have unique characteristics, although they also share some. For example, Metal and Punk, both have fast tempos, but are different in chord progressions and harmony. Rock uses Blues scales and progressions extensively, but has unique vocal melodies, and more dynamic drums, and also different structure.

Chapter 4

Methodology

4.1 Track classification

In order to characterize the musical content from each track, a vector of numeric descriptors, normally known as shallow structure description, is extracted. Then they are used as inputs for the classifiers — Neural Network and k-Nearest Neighbors. The implementation has been made in the Matlab environment. Also, the MidiToolbox Matlab toolbox was used for handling the MIDI files, and the Netlab toolbox was used for the Neural Network implementation.

4.1.1 MIDI track description

Each MIDI track is characterized by a numeric descriptors vector, such as pitch, note and silences information, which summarize the track musical content and provides a statistical overview of the track.

The descriptors chosen, capture, mainly, melodic aspects of music, unlike other works in this area [12], where large sets of features are used to capture several aspects of music (melodic, rhythmic, instrumental, etc). The goal is to achieve track and genre classification using a small set of features.

Based on other similar works in this area, twenty seven descriptors, plus twelve more that represent the pitch intervals histogram, have been defined, and are presented in Table 4.1. There are seven descriptors for track information, used to represent the track as a whole, and thirty two other descriptors for specific characteristics, which are subdivided into seven categories. Normalized values are computed for all descriptors, except the Intervals Histogram, so there's a proportional relation between all tracks from the same MIDI file.

The first category, Track Information, has seven descriptors: duration, the track duration in beats; number of notes; number of significant silences, which are silences greater than a tick (1/16 beat) - smaller silences are not considered silences, as they are almost imperceptible and non-significant; occupation rate, which is the

Category		Descriptors
Track Information (TI)	1	Duration
	2	# Notes
	3	# Significant silences
	4	Occupation rate
	5	Polyphony rate
	6	Consonance rate
	7	Dissonance rate
Pitch (P)	8	Highest
	9	Lowest
	10	Mean
	11	Standard Deviation
Pitch Intervals (PI)	12	# Different intervals
	13	Largest
	14	Smallest
	15	Mean
	16	Mode
	17	Standard Deviation
Note Durations (ND)	18	Longest
	19	Shortest
	20	Mean
	21	Standard Deviation
Silences Duration (SD)	22	Longest
	23	Shortest
	24	Mean
	25	Standard Deviation
Syncopation (S)	26	# Syncopated notes
Repetitions (R)	27	# Different n-grams
Intervals Histogram (semitones) (IH)	28	(0)Perfect Unison
	29	(1)Minor Second
	30	(2)Major Second
	31	(3)Minor Third
	32	(4)Major Third
	33	(5)Perfect Fourth
	34	(6)Augmented Fourth, Diminished Fifth
	35	(7)Perfect Fifth
	36	(8)Minor Sixth
	37	(9)Major Sixth
	38	(10)Minor Seventh
	39	(11)Major Seventh

Table 4.1: Descriptors

proportion of the track occupied by notes; polyphony rate, a proportion of the track occupied by two or more simultaneous notes; Consonance rate is the proportion of consonant notes, and Dissonance rate, the proportion of dissonant notes. These last two are not trivial and are explained later on. Pitch descriptors refer to the actual MIDI note value, ranging from 0 (C-2) to 127 (G8). Pitch interval is the difference between two consecutive notes, and gives important feedback about the track melody/harmony progression, namely, in respect to the number of different two-note intervals. n-grams descriptor also reflects the number of different pitch intervals, but on a three or four (depending on the track meter) consecutive notes basis. Note durations descriptors are self explanatory, as Silences durations, and are computed in beats. Syncopation is a rhythmic descriptor which reflects the number of notes whose onset is after the beat, normally in between beats, and is very frequent in jazz, and is therefore an important aspect to consider. Pitch intervals histograms show the frequency of the intervals semitones, giving valuable information about the musical scale and the kind of melody, or harmony, of the track. To maintain a proper relation between every pitch intervals histogram, and to enable the classifier to properly relate them, each track is transposed to C, and therefore, a pitch value of 0 is C, 1 is C#, 2 is D, etc.

In music, a consonance is a harmony, chord, or interval considered stable, as opposed to a dissonance, which is considered unstable (or temporary, transitional). The strictest definition of consonance may be only those sounds which are pleasant, while the most general definition includes any sounds which are used freely. Dissonance is the quality of sounds which seems "unstable", and has an aural "need" to "resolve" to a "stable" consonance. Both consonance and dissonance are words applied to harmony, chords, and intervals and by extension to melody, tonality, and even rhythm and meter. Understanding a particular musical style's treatment of dissonance is key in understanding that particular genre and also between different instruments (classes). For instance, in the common practice period, harmony is generally governed by chords, which are collections of notes generally considered to be consonant. Any note that does not fall within the prevailing harmony is considered dissonant. Particular attention is paid to how dissonances are approached, even more to how they are resolved, to how they are placed within the meter and rhythm (dissonances on stronger beats are considered more forceful and those on weaker beats less vital), and to how they lie within the phrase (dissonances tend to resolve at phrase's end). Jazz, for example, uses harmonies which may be considered dissonant, specially if compared with other genres, as Rock or Pop witch are mainly consonant.

4.1.2 Classifiers

Two different classifiers have been used to train and test the system, a Neural Network (NN), which is the main classifier, and k-Nearest Neighbors (kNN) for comparison purposes and for validating some decision choices on which descriptors should be used for best results. The kNN results were considered irrelevant, and are not shown.

Neural Network

Several Neural Networks (Multi-layer Perceptrons) were created for these experiments with a number of hidden units ranging from 40 to 100, in the search for the best balance between hidden units/computing time/results. Multi-Layer Perceptrons were trained using the scaled conjugate gradient algorithm.

K Nearest-Neighbor

k-Nearest Neighbor approach provides very fast results, given limited data files, and gives us the capacity to steer the experiments in the right direction.

4.1.3 Track selection

A melody track can be interpreted as the leading voice, an instrument solo or simply a monophonic instrument playing its part throughout the song. The melody which we are interested in, is the leading voice. In a Jazz or Blues song there isn't always an obvious melody, but instead, several solos, or a melody and a solo on the same track. Harmony is, normally, provided by instruments such as piano, organ, guitar, or a suite of strings, and are polyphonic, which contrasts with melody or solo tracks, which are mostly monophonic. Harmony tracks may also contain solos, but these are mostly played in accompaniment with chords - a pianist soloing with the right hand, accompanies himself with the left hand - so it's harmony nevertheless. Also, bass and drums are categorized, mostly because they are evidently different from the other instruments, and are, individually and together, very important components in genre definition. We present four classes of tracks: Melody; Harmony; Bass; Drums. Tracks which don't fit in these classes are discarded.

4.2 Genre classification

In the genre classification task, two approaches are used: language modeling, in which a transition probability matrix is created for each type of track (Melody, Harmony, Bass and Drums) and also for each genre; and an approach based on Neural Networks, where a vector of numeric descriptors is extracted from each type

of track (Melody, Harmony, Bass and Drums) and fed to a Neural Network Classifier. In both classifiers, each midi file being classified originates four individual results, one for each type of track (Melody, Harmony, Bass and Drums). Therefore, for each midi file, a $N \times 4$ (N being the number of genres) matrix is generated. For the final classification result, the matrix's columns are summed up, and the confusion matrix calculated. A Neural Network genre classification example, where one Midi file is processed and the resulting matrix is as following:

	Blues	Country	Jazz	Metal	Punk	Rock
Melody	0.68561	-0.15444	0.10184	-0.10081	0.34507	0.12041
Harmony	0.65524	0.35038	-0.037589	-0.045175	-0.15989	0.25046
Bass	0.715	-0.1044	-0.037867	0.21417	0.090746	0.22763
Drums	0.62086	0.20108	0.16822	0.15088	0.0021236	-0.14335
All	2.67671	0.29262	0.194604	0.219065	0.2780496	0.45515

Table 4.2: Genre Classification Matrix

This midi file is classified as Blues as it obtained the highest score. The final classification system for the Language Modeling classifier is analogous.

4.2.1 Language Modeling Approach

Language Modeling techniques[14] are used for the genre classification task, and fundamentally consists in four steps:

1. build a symbol dictionary which is used to represent any musical piece;
2. define a procedure to transform a musical piece into a sequence of symbols;
3. build a model for each musical genre;
4. find a procedure, that given a set of models and a sequence of symbols, determine the best model which fits this sequence.

Single-Instrument Beat Similarity

In this approach, all the tracks are split into beats. A beat dictionary is assembled, from the unique beats of all the beat split tracks. A probability transition matrix is then created for each type of track, where each beat is compared to each other through a similarity function.

Beat Division A beat is the basic time unit of a piece of music, and denotes the complete time interval between two consecutive instants in time. In this experiment, a beat contains the onset, duration and velocity for each note comprised between instant t and instant $t + 1$.

There may be some very short notes, which may introduce some unwanted "noise" to the beats. To prevent this, we eliminate notes smaller than 1/16 of a beat. Each track is then transposed to C so that all beats are on the same key, and can be properly compared. Transposing is done using the Krumhansl-Kessler algorithm.

Each track, is then split into an array containing all the beats, as shown in Figure 4.1.

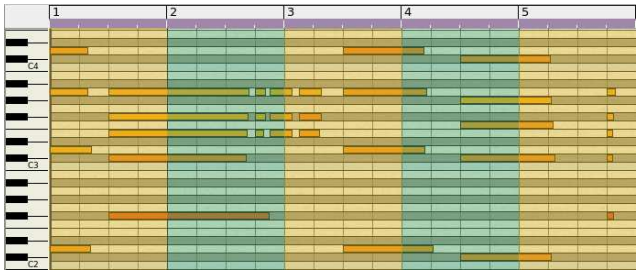

Figure 4.1: A track split by beats (beats are shown in yellow and green).

Similarity Function The similarity function A compares two beats based solely on the note content of each beat. First, each beat is measured (end of last note minus first note onset time), and then the mean of both beats measure is calculated so that only beats with approximately the same length are compared. The threshold value is also taken into account when comparing both beats, so that the higher the threshold, the larger the measure difference. Then, both beats are scanned in parallel with a 1/8 beat window, for chords or single notes. The chords are compared, as explained below, first by comparing the chord lowest pitch. If they're equal, the chord notes are ordered and compared, resulting in a boolean vector (1 equal, 0 different). When the whole beat is processed, the resulting vector mean is calculated obtaining the similarity value. This similarity value is then compared with the threshold value to validate,or not, the two beats similarity.

Each pair of beats is compared through the similarity function, which looks for similar notes or chords. Comparing two single notes is obviously simple: they're similar if the note key is the same; the duration difference between the two notes is proportional to the similarity value. Comparing two chords can be much more complicated: they're also similar if the chords have the same notes and are on the same key. Problems arise when two chords have the same notes, but aren't on the

same key. C-E-A is different from A-E-C, although they both have the same notes. Another problem is for chords which differ only in one note. For example, a chord C-E-G is similar to C-E-G-B, on the other hand, C-E-G-B isn't similar to E-G-B, as they are different chords made up from the same notes, as shown in Figure 4.2.

One simple solution is to use the lower note of the chord as the chord key, and therefore, compare only chords on the same key. Chords on different keys are considered different, as shown in Figure 4.2.

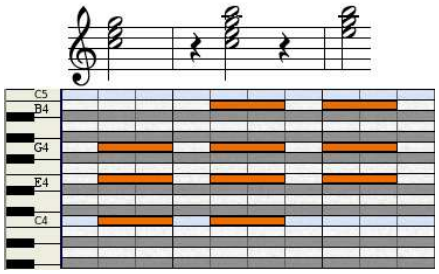

Figure 4.2: The first and the second chords are similar, the third is different.

The similarity function compares two beats using a window of 1/8 of a beat to find the chords or single notes that make up the beat. The similarity variable varies between 0 (equal) and 1 (different).

Building Beat Dictionaries The creation process of a beat dictionary, per instrument, consists in three steps:

1. extract all beats from all tracks, of a single instrument, from the training set;
2. compare all beats between themselves using the similarity function described in 4.2.1;
3. build the dictionary with the resulting unique beats.

The dictionary size varies from genre to genre. Harmony dictionaries are larger than Melody dictionaries, because harmony beats are more complex than melody beats, and thus more diverse. The same principle applies to genre. A Jazz dictionary is larger than a Punk dictionary, for any instrument, because, simply put, jazz is more complex than punk. A problem arises from this fact: a very large dictionary, with length D, is not desirable, as it represents $D \times D$ possible transitions, therefore the transition probabilities are very low. A smaller dictionary doesn't have this problem because each transition occurs more often. To cope with this problem, a higher similarity threshold, per instrument, has to be used, to build dictionaries ranging from 1000 to 3000 symbols. Also, the fact that some genre dictionary is larger than some other genre, is meaningful and important as it promptly highlights the difference between the genres.

Classification of music files The classification of a musical piece is done by translating it into a sequence of symbols and computing the probability that each model generates from this sequence. Given a model M, the probability generated from the sequence $S = s_1, s_2, ...s_n$ is:

$$P_M(s_1) \prod_{i=2}^{n} P_M(s_i|s_{i-1}) \tag{4.1}$$

which is better calculated as

$$\log(P_M(s_1)) + \sum_{i=2}^{n} \log(P_M(s_i|s_{i-1})) \tag{4.2}$$

This score is computed for each model M_j, and the highest scoring one has its corresponding class assigned to the sequence of symbols.

Single-Instrument Shallow Structure Beat Similarity

In this approach, the "shallow structure description" technique, described in 4.1.1, is used to characterize each beat.

Beat Division Identical to Section 4.2.1.

Two-stage clustering The first stage extracts the k most representative beats from each musical piece, using the k-means clustering algorithm. These k beats are the number of clusters per musical piece, and are called k_1. This set of $n \times k_1$ vectors, where n is the number of musical pieces from the training set, is called $\mathcal{F}_1$.

The second stage consists in extracting a set of k_2 beats, the most representative in $\mathcal{F}_1$, using the k-means algorithm again. This new set, $\mathcal{F}_2$, contains the k_2 centroids obtained from the clustering, and for each one, a symbol is assigned, thus creating a dictionary $\mathcal{D}$, composed of k_2 symbols.

To translate a new musical piece into a set of symbols, three steps are needed:

1. Process each musical piece MIDI track to obtain the correspondent numeric descriptors vector.
2. Compute the mean of all the tracks numeric descriptors vector.
3. Compute the first nearest neighbor in $\mathcal{F}_2$, for each beat, and assign the corresponding symbol.

Estimation of n-grams The following phase is the estimation of a language model for each genre into which we want to classify the musical pieces.

For each musical genre, the probability of each bigram is computed by processing every sequence of symbols, from the training set, and counting the occurrences of the

symbols transitions. The result is a transition probability matrix which contains, for each pair of symbols (s_i, s_j), the symbol s_i probability $P(s_j|s_i)$ to be followed by the symbol s_j. In a genre classification task context, this transition probability matrix represents a model, which is estimated for each genre by processing the n-grams from the files which belong to a specific genre.

As the training sets used to estimate the models are finite and relatively small, there are many transitions which never occur, therefore, with zero probability. This can lead to a musical piece, being tested, to achieve a zero probability, needing only to observe a single transition which has not been seen before in the training set. To avoid this erroneous behavior, a small constant $\epsilon =$ is added to each non-observed transition.

In the Beat Division case, explained in 4.2.1, a similarity function to encode beats into symbols is used, and it is possible to find symbols in a test sequence that have never been seen during the training phase. To deal with this aspect a special symbol is added to the dictionary that matches any previously unseen symbol. The transitions probabilities from this special symbol to the other symbols of the dictionary are set to be equiprobable.

Note that this problem does not arise with the n-gram approach because symbols are assigned to the nearest centroid in $\mathcal{F}_2$.

Classification of music files Identical to Section 4.2.1.

Multi-Instrument Shallow Structure Beat Similarity

Similar to its Single-Instrument counterpart, described in section 4.2.1. Each track (melody, harmony, bass and drums) is processed using the technique described in Section 4.1.1 and the mean of all the tracks is computed, resulting in one numeric descriptors vector, per beat.

4.2.2 Neural Network Approach

A Neural Network approach for genre classification is used for its simplicity and mainly for comparison purposes for the main classifier. It's a simple classification technique which resumes each midi files Melody, Harmony, Bass and Drum tracks, using the technique described in Section 4.1.1, and feeds these resulting feature vectors into a neural network.

Chapter 5

Experiments

All experiments were done using the n-fold cross validation method, where the data set is divided into n subsets, and the classification method is repeated n times. Each time, one of the "n" subsets is used as the test set and the other "n-1" subsets are put together to form a training set. Then the average classification results across all n trials is computed. Different n values were used for the track and genre classification experiments.

5.1 Music Corpora

Six MIDI Music Corpora were assembled for the experiments, as depicted in Table 5.1, from six different genres, Blues, Country, Jazz, Metal, Punk and Rock. These genres were selected because all of them have the same base instruments, such as bass, drums, piano or guitar. Also, the genres chosen share some characteristics between them, so that the classification isn't trivial, and tests the classifiers robustness.

Corpus	# Midis	# Tracks
Blues	72	240
Country	69	260
Jazz	93	320
Metal	98	340
Punk	78	290
Rock	77	290
All	487	1740

Table 5.1: Music Corpora for Track and Genre Classification

For the Track Classification Experiments, all the tracks from all the midi songs were used. For the Genre Classification Experiments, sixty six midi songs, from each genre, were used.

5.2 Track Classification Experiments

Early experiments showed that using all descriptors gives poor results, leading us to experiment with one descriptor category at a time, such as Pitch or Notes, or a combination of two categories. This approach led us to another problem. Which category combination was better? And which single descriptor combination? Testing every possible combination couldn't even be an hypothesis, for a very large number of combinations can be made out of all the thirty seven single descriptors.

A very simple algorithm solved the problem. The set of descriptors is built by testing each descriptor individually and joining iteratively more descriptors to the set while the performance increases. Section 5.2.3 shows some results obtained with different sets of descriptors.

The n value for the n-fold cross validation method used in these set of experiments is calculated from the following formula:

$$n = \frac{k}{10} \tag{5.1}$$

where k is the number of tracks from a given genre.

The Matlab environment was used to implement the system and to perform the experiments. An additional toolbox was used, the MidiToolbox for helping with the handling of MIDI files in the Matlab environment. Matlab was chosen because it already sports a vast array of functions, classifiers, graphics, plots, etc, which help in analyzing the MIDI files.

5.2.1 Track Classification: all descriptors

In the first set of experiments, all thirty nine descriptors were used, which proved to be a good, although naive, approach. Several NN were used, with hidden units ranging from 40 to 80, and they all presented basically the same results, varying only in 3%, so the best network was used, with 40 hidden units. The confusion matrices are shown in tables 5.2.

Melody	30	0	1	0
Harmony	2	66	0	0
Bass	0	1	70	0
Drums	0	4	0	66

Table 5.2: Blues Confusion Matrix. NN: all descriptors. Classification Rate: 96,7%

Using all descriptors, is a good, but naive approach, because some descriptors may successfully distinguish between two different classes, but another descriptor

Melody	59	1	1	0
Harmony	0	60	1	4
Bass	1	1	64	1
Drums	0	4	1	62

Table 5.3: Country Confusion Matrix. NN: all descriptors. Classification Rate: 94,2%

Melody	47	2	0	0
Harmony	2	86	1	0
Bass	0	0	91	0
Drums	0	0	0	91

Table 5.4: Jazz Confusion Matrix. NN: all descriptors. Classification Rate: 98,4%

Melody	55	2	1	0
Harmony	0	92	3	1
Bass	1	6	84	0
Drums	0	2	1	92

Table 5.5: Metal Confusion Matrix. NN: all descriptors. Classification Rate: 95%

Melody	63	2	1	0
Harmony	2	73	0	1
Bass	2	0	70	1
Drums	0	4	1	70

Table 5.6: Punk Confusion Matrix. NN: all descriptors. Classification Rate: 95,2%

Melody	59	1	0	0
Harmony	4	69	2	2
Bass	4	0	73	0
Drums	0	1	3	72

Table 5.7: Rock Confusion Matrix. NN: all descriptors. Classification Rate: 94,1%

may distinguish the same classes in an opposite way, and confuse the final classification.

All the tracks, from all genres, had basically the same score, which means that the descriptors used are enough for successfully classifying tracks into these four classes.

Melody	244	12	7	0
Harmony	8	373	4	10
Bass	3	5	374	7
Drums	0	20	8	365

Table 5.8: All Corpus Confusion Matrix. NN: all descriptors. Classification Rate: 94,2%

5.2.2 Track Classification: single descriptor category

A different approach was used in the following experiments. Instead of using the full set of descriptors, six sets of descriptors were used, corresponding to the descriptor categories, and also some combinations of the best scoring sets. A single NN was used with hidden units set to 40. The results are shown in table 5.9.

Set	Blues	Country	Jazz	Metal	Punk	Rock	All
TI	82,9	81,9	90,6	85,9	88,3	80,3	79,9
P	65,4	60,8	83,1	65	76,9	63,1	66,8
PI	59,6	63,8	76,9	70	74,5	54,8	63,8
ND	59,2	57,7	72,8	57,4	50,3	58,3	54,4
SD	49,2	46,2	52,5	46,5	45,9	39,3	43,3
S	62,5	60	63,8	71,2	72,1	61	51
R	40,8	42,7	68,1	48,5	39,7	39	45,3
IH	75,4	64,2	66,6	62,1	56,2	58,3	57,4
TI+P	**90,4**	**88,5**	**96,9**	**87,4**	**93,4**	**88,6**	**88,8**
P+PI	81,7	77,3	89,1	80	89,3	73,8	80,8

Table 5.9: Single Categories Classification Rates

The experiments results were, as expected, worse than those achieved using all the descriptors, but nevertheless presented interesting results. The TI set alone, provided good results, giving the impression that it is the set of descriptors which better describes a track's characteristics, or that it's descriptors are the most important for this classification experiment. All the other single sets yielded worse results and are clearly confusing the classifier, and should not be used, at least not in this naive way. The combined sets, TI+P and P+PI, also presented good results. Given its good results, TI+P could be used solely in a scenario where processing time is crucial, as the processing of a mere 11 descriptors doesn't require much processing power, therefore less time.

These results showed that, for an optimum classification, the descriptors have to be carefully selected, not only by combining categories, but combining single descriptors.

5.2.3 Track Classification: best descriptors

Using the algorithm described previously, a possible best descriptors set was found for each genre and for all genres combined. The results are presented in Table 5.10 (numbers are correspondent to 4.1) for NN using 40 hidden units.

Genre	Rate (all)	Rate (best)	Best Descriptors
BLUES	96,7%	97,9%	[2 3 4 5 7 8 10 13 17 19 21 22 23 24 25 26 27 28 29 31 33 34 35 38]
COUNTRY	94,2%	96,9%	[1 3 4 5 6 7 9 10 13 14 15 21 22 26 35 37]
JAZZ	98,4%	99,4%	[4 5 8 10 11 12 15 18 21 23 24 28 29 31 32 35 36 37 38]
METAL	95%	96,5%	[1 2 3 4 5 6 7 8 10 11 12 14 15 16 17 19 20 21 22 24 25 26 28 29 30 31 32 33 36 39]
PUNK	95,2%	97,9%	[2 5 7 10 14 17 21 26 27 33 34 35 36]
ROCK	94,1%	96,2%	[2 3 4 5 8 10 17 18 19 20 22 33 35 36 38]
ALL	94,2%	95%	[3 4 5 7 8 10 11 12 14 15 17 18 19 20 21 22 23 26 27 28 29 30 31 32 33 34 35 36 38]

Table 5.10: All Descriptors Vs Best Descriptors Classification Rates

As expected, all genres improved their classification score. Some interesting observations can be drawn from these results, namely, from the best descriptors sets found.

Polyphony Rate (#5) and Mean Pitch (#10), are present in all sets. Polyphony is important in distinguishing between different instrument tracks. A bass is mainly monophonic, with a very low polyphony rate, as melody tracks. Harmony is highly polyphonic, as Drums. Mean Pitch is also very important, as it characterize a track's pitch scope. A bass track has a lower mean pitch than a melody track. These two descriptors alone, allow to differentiate a large number of simple tracks. A track with a low polyphony rate a low Mean Pitch is probably a Bass track, but with a high Mean Pitch is probably a Melody track. A track with a high polyphony rate and a medium Mean Pitch is probably a Harmony track, but, most probably, not a bass or a melody track.

Perfect Fourth (#33), Minor Sixth (#36) and Perfect Fifth (#35) are present in most sets, and according to music theory, these intervals are one of the most consonant, because they have simple pitch relationships resulting in a high degree of consonance, which is perfect for distinguishing between, for example, a simple Melody from a complicated Harmony.

5.3 Genre Classification Experiments

5.3.1 Single-Instrument Beat Similarity experiment

For this experiment, the Language Modeling technique explained in 4.2.1 is used. The similarity threshold values used are: Melody = 0.49; Harmony = 0.60; Bass = 0.49; Drums = 0.49.

Classification using the four extracted tracks is achieved through the technique explained in 4.2. The results are displayed in Table 5.11.

BLUES	37	4	14	0	1	10	56,1%
COUNTRY	3	57	4	0	1	1	86,3%
JAZZ	0	0	66	0	0	0	100%
METAL	0	0	0	56	2	8	84,8%
PUNK	1	0	0	5	49	11	74,2%
ROCK	2	2	1	13	2	46	69,7%

Table 5.11: Confusion Matrix: Single-Instrument Beat Similarity (78,5%).

5.3.2 Single-Instrument Shallow Structure Beat Similarity experiment

For this experiment, the technique described in 4.2.1, is used. K_1 and k_2 values are 30 and 300.

Classification using the four extracted tracks is achieved through the technique explained in 4.2.

The results are displayed in Table 5.12.

BLUES	28	5	7	2	5	19	42,4%
COUNTRY	1	48	5	0	5	7	72,7%
JAZZ	2	7	54	0	0	3	81,8%
METAL	0	0	0	47	2	17	71,2%
PUNK	3	2	0	2	51	8	77,3%
ROCK	3	8	3	10	3	39	59,1%

Table 5.12: Confusion Matrix: Single-Instrument Shallow Structure Beat Similarity (67,4%).

5.3.3 Multi-Instrument Shallow Structure Beat Similarity experiment

In this experiment, the same technique from the last experiment is used, but instead of using separate tracks, all tracks are "merged" together through a mean operation on all track's descriptors vector. K_1 is 20 and k_2 is 60.

The results are displayed in Table 5.13.

BLUES	34	6	20	1	0	5	51,5%
COUNTRY	7	40	8	4	0	7	60,6%
JAZZ	5	0	60	0	0	1	90,9%
METAL	0	0	1	51	3	11	77,3%
PUNK	2	3	2	27	14	18	21,2%
ROCK	9	5	9	8	4	31	47%

Table 5.13: Confusion Matrix: Multi-Instrument Shallow Structure Beat Similarity (62%).

5.3.4 Neural Network: Single-Instrument Shallow Structure

The technique used is described in 4.2.2. Four networks were used, one for each instrument, with 40 hidden units. Classification using the four extracted tracks is achieved through the technique explained in 4.2.

The results are displayed in Table 5.14.

BLUES	46	4	7	2	1	6	69,7%
COUNTRY	4	59	2	0	0	1	89,4%
JAZZ	3	1	61	0	0	1	92,3%
METAL	3	0	0	56	2	5	84,8%
PUNK	1	3	1	2	53	6	80,3%
ROCK	6	2	1	7	11	39	59,1%

Table 5.14: Confusion Matrix: Single-Instrument Shallow Structure (79,3%).

5.3.5 Neural Network: Multi-Instrument Shallow Structure

In this experiment, the same technique from the last experiment is used, but instead of using separate tracks, all tracks are "merged" together through a mean operation on all track's descriptors vector. A single network, with 40 hidden units, was used. The results are presented in Table 5.15.

BLUES	33	5	12	4	6	6	50%
COUNTRY	4	51	5	2	3	1	77,3%
JAZZ	6	2	57	0	0	1	86,4%
METAL	4	2	2	36	10	12	54,5%
PUNK	5	8	4	5	39	5	59,1%
ROCK	12	5	2	10	14	23	34,8%

Table 5.15: Confusion Matrix: Multi-Instrument Shallow Structure (60,3%).

5.4 Approaches Comparison

Table 5.16 resumes all the experiments results. The best results are presented in bold.

	5.3.1	5.3.2	5.3.3	5.3.4	5.3.5
BLUES	56,1%	42,4%	51,5%	**69,7%**	50%
COUNTRY	86,3%	72,7%	60,6%	**89,4%**	77,3%
JAZZ	**100%**	81,8%	90,9%	92,3%	86,4%
METAL	**84,8%**	71,2%	77,3%	**84,8%**	54,5%
PUNK	74,2%	77,3%	21,2%	**80,3%**	59,1%
ROCK	**69,7%**	59,1%	47%	59,1%	34,8%
TOTAL	78,5%	67,4%	62%	**79,3%**	60,3%

Table 5.16: Experiments Comparison.

From the three Language Modeling experiments (5.3.1 5.3.2 5.3.3), Single-Instrument experiments 5.3.1 and 5.3.2, achieved the best results, namely in Jazz with 100% success rate, followed by Country with 86,3% and Metal with 84,8%, all above 80% success rate, which are good results. For the Neural Network experiments, again, the Single-Instrument experiment 5.3.4, achieved better results than 5.3.5. Four genres achieved higher than 80% success rates: Jazz, Country, Metal and Punk.

Evidently, Blues and Rock didn't perform very well, probably because of a badly assembled music corpora, as other genres performed good enough, proving the classifiers potential and robustness. Another reason for the Blues and Rock low scores is the fact that these two genres share a lot of characteristics with the other genres, and also between themselves, as can be observed from the confusion matrixes. One could, in a naive exercise, believe in the classifiers misclassification and conclude that, for example, Blues share some characteristics with Jazz, and Punk share some characteristics with Rock, and that wouldn't be a false conclusion. In fact Blues and Jazz share a common musical background, and Punk derives from Rock. It's no surprise that Jazz performed so well, as it has some unique characteristics which

the other genres do not have, at least not at the same extend of Jazz. Dissonance, for example, is largely used in Jazz, as are Syncopated rhythms.

Chapter 6

Future Work

6.1 Genetic algorithm

A genetic algorithm is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics and are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Genetic algorithms provide a better search tool to find the best possible descriptors set, in opposition to the hill climbing algorithm used previously.

6.2 Track, Song and Genre Structure

Different musical genres normally have specific song structures which contribute to the uniqueness of the genre. Jazz music has a typical ABA structure, in which A is the beginning and ending theme/melody of the song, and B is the large solo part. Rock, on the other hand, can have a ABABCB or a ABABB structure. The goal is to structurize each track/song into sections and search for differences between genres and similarities between songs from the same genre.

6.3 Merging Track and Genre Classification

A full automatic Genre Classification System, should be able to classify any track, without any preprocessing, such as track tagging. The Track Classification system, given it's excellent results, could be used for this task. A midi file would have it's instruments separated by the Track Classification system, and then, each track would be forwarded to a Single-Instrument Genre Classification system (4.2.1, 4.2.1 and 4.2.2). This way, there would be no gap between track and genre classification.

An interesting task would be to merge all the classifiers, using a technique similar to the one described in 4.2.

Chapter 7

Conclusion

An Automatic Classification of Midi Tracks system, and two Genre Classification approaches were developed and presented. Track Classification uses MIDI files from six genres, Blues, Country, Jazz, Metal, Punk and Rock, and classifies the tracks in four classes, Melody, Harmony, Bass and Drums. A neural network is used to process thirty nine descriptors extracted from each MIDI track, which has been previously tagged in the four classes. The experiments showed that using all descriptors is a wrong approach, as there are descriptors which confuse the classifier. Using carefully selected descriptors proved to be the best way to classify these MIDI tracks.

In Genre Classification, two approaches were used: language modeling, in which a transition probabilities matrix is created for each track and also for each genre; and an approach based on Neural Networks, where a vector of numeric descriptors is extracted from each track and fed to a Neural Network Classifier. Experiments showed that the Single-Instrument classifiers, 5.3.1, 5.3.2 and 5.3.4, achieved the best results, mainly above 80% success rate for each genre, which are good results. Four genres achieved higher than 80% success rates: Jazz, Country, Metal and Punk.

Future work: genetic algorithms which provide a better search tool to find the best possible descriptors set, in opposition to the hill climbing algorithm used previously; structurize each track/song into sections and search for differences between genres and similarities between songs from the same genre; merge all presented classifiers into one full Automatic Genre Classification System.

Two papers were written during this research, which served as the basis to this thesis. The first, "Automatic Classification of Midi Tracks", written by myself and Prof. Thibault Langlois, was accepted at ICEIS2008 [2], and focused on Midi Track Classification. The second, "A Language Modeling Approach for the Classification of Midi and Audio Music" focused on Genre Classification of both Audio and Midi, by Prof. Thibault Langlois, Gonçalo Marques, and myself.

Globally, the research was highly successful, as results are concerned, and also as an important step for my academic, personal and professional development. My

contribution for this research area, Music Information Retrieval, was, in my opinion, very positive and valuable, as the work done provides new research directions. My hope is that the work developed helps future researchers in reaching far beyond than where I've reached.

Bibliography

[1] Roberto Basili, Alfredo Serafini, and Armando Stellato. Classification of musical genre: a machine learning approach. In *ISMIR*, 2004.

[2] Alexandre Bernardo and Thibault Langlois. Automatic classification of midi tracks. In José Cordeiro and Joaquim Filipe, editors, *ICEIS (2)*, pages 539–543, 2008.

[3] Zehra Cataltepe, Yusuf Yaslan, and Abdulah Sonmez. Music genre classifcation using midi and audio featres. *Journal on Adances in Signal Processing*, 2007, 2007.

[4] Rudi Cilbrasi, Paul Vitányi, and Ronald de Wolf. Algoritmic clustering of music based on string compression. *Computer Music Journal*, 29(4):49–67, 2004.

[5] Rizo D., Ponce de León P. J., Pérez-Sancho C., and Iñesta J. M. Pertusa A. A pattern recognition approach for melody track selection in midi files. In Tindale A. Dannenberg R., Lemström K., editor, *Proc. of the 7th Int. Symp. on Music Information Retrieval ISMIR 2006*, pages 61–66, Victoria, Canada, 2006. ISBN: 1-55058-349-2.

[6] Rizo D., Ponce de León P.J., and Iñesta J.M. Pertusa A. Melodic track identification in midi files. In *Proc. of the 19th Int. FLAIRS Conference*. AAAI Press, 2006. ISBN: 978-1-57735-261-7.

[7] C. DeCoro, Z. Barutcuoglu, and R. Fiebrink. Bayesian aggregation for hierarchical genre classification. In *ISMIR*, 2007.

[8] Yo-Ping Huang, Guan-Long Guo, and Chang-Tien Lu. Using back propagation model to design a midi music classification system. In *International Computer Symposium*, pages 253–258, Taipei, Taiwan, December 2004.

[9] Ming Li and Ronan Sleep. Improving melody classification by discriminant feature extraction and fusion. In *Proc. of the 5th Int. Symp. on Music Information Retrieval ISMIR 2004*, 2004.

[10] Ming Li and Ronan Sleep. Melody classification using a similarity metric based on kolmogorov complexity. In *Sound and Music Computing*, Paris, France, October 2004.

[11] S. T. Madsen and G. Widmer. A complexity-based approach to melody track identification in midi files. In *International Workshop on Artificial Intelligence and Music (MUSIC-AI 2007)*, Hyderabad, India, January 2007.

[12] Cory McKay and Ichiro Fujinaga. Automatic genre classification using large high-level musical feature sets. In *ISMIR*, 2004.

[13] C. Pérez-Sancho, J.M. Iñesta, and J. Calera-Rubio. A text categorization approach for music style recognition. In *IbPRIA (2)*, pages 649–657, 2005.

[14] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

[15] Adi Ruppin and Hezy Yeshurun. Midi music genre classification by invariant features. In *ISMIR*, pages 397–399, October 2006.

Appendices

Appendix A

Beat Similarity Function

$l1$ = length($beat1$);
$l2$ = length($beat2$);
if $l1 > l2$ **then**
  $onsetDiff = l2/l1$;
else if $l1 < l2$ **then**
  $onsetDiff = l1/l2$;
else
  $onsetDiff = 1$;
end if
$onsetDiff = (onsetDiff + 1)/2$;
if $onsetDiff >$ (0.95 + (1 - $similarityThreshold$))/2 **then**
  out = empty set;
  while $beat1$ is not empty AND $beat2$ is not empty **do**
    a = get first note or chord from $beat1$;
    b = get first note or chord from $beat2$;
    if (lower pitch from a) is equal to (lower pitch from b) **then**
      p_a = sort pitches from a;
      p_b = sort pitches from b;
      $ml = p_a$ is equal to p_b;
    else
      $ml = 0$;
    end if
    out = concatenate out with ml;
    $beat1$ = $beat1$ except first note or chord;
    $beat2$ = $beat2$ except first note or chord;
  end while
  out = 1 - mean(out);
else

$out = onsetDiff;$
end if

Appendix B

ICEIS2008 Paper, Automatic Classification of Midi Tracks.

AUTOMATIC CLASSIFICATION OF MIDI TRACKS

Alexandre Bernardo, Thibault Langlois
Universidade de Lisboa, Faculdade de Cilncias, Departamento de Informtica, Portugal
ab.mail@gmail.com, tl@di.fc.ul.pt

Keywords: MIDI Track Classification, Music Information Retrieval, Neural Networks, *k*-Nearest Neighbors.

Abstract: This paper presents a system for classifying MIDI tracks according to six predefined classes: Solo, Melody, Melody+Solo, Drums, Bass and Harmony. No metadata present in the MIDI file is used. The MIDI data (pitch of notes, onset time and note durations) are preprocessed in order to extract a set of features. These data sets are then used with several classifiers (Neural Networks, *k*-NN).

1 INTRODUCTION

Music Information Retrieval is, nowadays, a highly active branch of research and development in the computer science field, and focuses on several topics such as beat tracking, music genre classification, melody extraction, score-following, to name a few.

There are a lot of known applications that use this technology for some extent: the new generation media players, which organizes music in an intelligent way, based in the music itself, and generates, for example, dynamic playlists; Internet radio stations, which builds a playlist based on the user's taste; score following; finding similarities between songs in a large database.

The work, presented in this paper, focuses on music stored using MIDI format. Electronic instruments use this format to communicate and synchronize themselves. The format consists in a number of tracks where each track represents the sequence of notes (pitch level and duration) played by one instrument. MIDI files also contain some metadata, such as the instrumentation or key. One of the advantages of the MIDI format is its compactness. Many musical resources using this format are freely available on the Internet.

Previous work in Music Information Retrieval using MIDI format includes music genre detection where several approaches have been proposed. Some researchers use similarity measures based on Kolmogorov complexity estimates in conjunction with a classical Machine Learning technique like *k*-Nearest Neighbors (Ruppin and Yeshurun, 2006), Support Vector Machines (Li and Sleep, 2004b) or clustering (Cilbrasi et al., 2004). Cataltepe (Cataltepe et al., 2007) compares the performance of obtained with the Normalized Compression Distance approach on MIDI and audio files with the ad-hoc features extraction and Machine Learning approach proposed by McKay (McKay and Fujinaga, 2004).

Other researchers proposed to extract a set of features from MIDI files and perform a genre classification using Neural Networks (McKay and Fujinaga, 2004) (Huang et al., 2004) or Support Vector Machines in conjunction with dimensionality reduction techniques (Li and Sleep, 2004a). Basili (Basili et al., 2004) made a comparison of various Machine Learning techniques on a musical genre classification task.

Another approach is to perform automatic melody detection. Rizo et al. (D. et al., 2006a) (D. et al., 2006b) has proposed a set of features to characterize each MIDI track and used a Random Forest classifier to identify tracks which contain melody. In (Madsen and Widmer, 2007), an information-theoretic complexity measure and an estimate of the local entropy are used to recognize melody tracks.

In this paper we address the problem of MIDI track classification. Based on the pitch levels and durations which describe each track, we extract a set of

features that are used to train a classifier. It is important to note that, in contrast to many other previously published studies, our approach does not use any metadata present in the MIDI file (such as instrumentation). Tracks are classified into six classes: Solo, Melody, Melody+Solo, Drums, Bass and Harmony. Two Machine Learning approaches are compared. The rest of the paper is organized in the following way: section 2 describes the data and the different sets of descriptors and the classifiers that were used. Section 3 reports the experiments and the results obtained. Finally, section 4 concludes and discusses future directions of research.

2 METHODOLOGY

In order to characterize the musical content from each track, a vector of numeric descriptors, normally known as shallow structure description, is extracted. Then they are used as inputs for the classifiers — Neural Network and *k*-Nearest Neighbors — which were implemented in the Matlab environment. Also, the MidiToolbox Matlab toolbox was used for handling the MIDI files.

2.1 MIDI track description

Each MIDI track is characterized by a vector of numeric descriptors, such as pitch, note and silences information, which summarizes the track musical content and provides a statistical overview of the track. Based on other similar works in this area, twenty five descriptors, plus twelve more that represent the pitch intervals histogram, have been defined, and are presented in Table 1. There are five descriptors for track information, used to represent the track as a whole, and thirty two other descriptors for specific characteristics, which are subdivided into seven categories. Normalized values are computed for all descriptors, except the Intervals Histogram, so there's a proportional relation between all tracks from the same MIDI file.

The first category, Track Information, has five descriptors: duration, the track duration in beats; number of notes; number of significant silences, which are silences greater than a tick (1/16 beat) - smaller silences are not considered silences, as they acknowledgments are almost imperceptible and non-significant; occupation rate, which is the proportion of the track occupied by notes; polyphony rate, a proportion of the track occupied by two or more simultaneous notes. Pitch descriptors refer to the actual MIDI note value, ranging from 0 (C-2) to 127 (G8).

Table 1: Descriptors

Category		Descriptors
Track Information (TI)	1	Duration
	2	# Notes
	3	# Significant silences
	4	Occupation rate
	5	Polyphony rate
Pitch (P)	6	Highest
	7	Lowest
	8	Mean
	9	Standard Deviation
Pitch Intervals (PI)	10	# Different intervals
	11	Largest
	12	Smallest
	13	Mean
	14	Mode
	15	Standard Deviation
Note Durations (ND)	16	Longest
	17	Shortest
	18	Mean
	19	Standard Deviation
Silences Durations (SD)	20	Longest
	21	Shortest
	22	Mean
	23	Standard Deviation
Syncopation (S)	24	# Syncopated notes
Repetitions (R)	25	# Different n-grams
Intervals Histogram (semitones) (IH)	26	(0)Perfect Unison
	27	(1)Minor Second
	28	(2)Major Second
	29	(3)Minor Third
	30	(4)Major Third
	31	(5)Perfect Fourth
	32	(6)Augmented Fourth, Diminished Fifth
	33	(7)Perfect Fifth
	34	(8)Minor Sixth
	35	(9)Major Sixth
	36	(10)Minor Seventh
	37	(11)Major Seventh

Pitch interval is the difference between two consecutive notes, and gives important feedback about the track melody/harmony progression, namely, in respect to the number of different two-note intervals. *n*-grams descriptor also reflects the number of different pitch intervals, but on a three or four (depending on the track meter) consecutive notes basis. Note durations descriptors are self explanatory, as Silences durations, and are computed in beats. Syncopation is a rhythmic descriptor which reflects the number of notes whose onset is after the beat, normally in between beats. Syncopation is very frequent in jazz, and is also an important aspect to consider. Pitch intervals histograms show the frequency of the intervals semitones, giving valuable information about the musical scale and the kind of melody, or harmony, of the track.

2.2 Classifiers

Two different classifiers have been used to train and test the system, a Neural Network (NN), which is the main classifier, and *k*-Nearest Neighbors (*k*NN) for comparison purposes and for validating some decision choices on which descriptors should be used for best results.

2.2.1 Neural Network

Several Neural Networks (Multi-layer Perceptrons) were created for these experiments with a number of hidden units ranging from 40 to 100, in the search for the best balance between hidden units/computing time/results. Multi-Layer Perceptrons were trained using the scaled conjugate gradient algorithm.

2.2.2 K Nearest-Neighbors

k-Nearest Neighbors approach provides very fast results, given limited data files, and gives us the capacity to steer the experiments in the right direction.

2.3 MIDI file format

The MIDI format has several ways to organize each track, and unfortunately, there is no real standard, because there are numerous ways, MIDI sequencers, to create a MIDI song, and each MIDI sequencer may create the MIDI file in a different way. This can lead to several problems when interpreting the MIDI (i.e. all the instruments on the same MIDI track but on different MIDI channels). Rosegarden, an open-source MIDI sequencer, was used to normalize all the MIDI files used in the experiments, so that all share the same structure.

2.4 Track selection

A melody track can be interpreted as the leading voice, an instrument solo or simply a monophonic instrument playing its part throughout the song. The melody which we are interested in, is the leading voice. In a Jazz song there isn't always an obvious melody, but instead, several solos, or a melody and a solo on the same track. Harmony is, normally, provided by instruments such as piano, organ, guitar, or a suite of strings, and are polyphonic, which contrasts with melody or solo tracks, which are mostly monophonic. Harmony tracks may also contain solos, but these are mostly played in accompaniment with chords - a pianist soloing with the right hand, accompanies himself with the left hand - so it's harmony nevertheless. Also, bass and drums are categorized, mostly because they are evidently different from the other instruments, and are, individually and together, very important components in genre definition. We present six classes of tracks: Melody; Melody+Solo; Solo; Harmony; Bass; Drums. Tracks which don't fit in these classes are discarded.

2.5 Music Corpora

Two MIDI Music Corpora were assembled for the experiments, as depicted in Table 2, from only one genre, Jazz, as it incorporates the most common problems in identifying the different components of a song: jazz hasn't obvious singing voice melodies, has various solos on several tracks and most songs have enough instruments to populate our six classes with different data. This gives us enough different issues to solve in the descriptor extraction and classification methods. As the name implies, the neural networks were trained using the "training" set, and tests using the "test" set. [1]

Table 2: Music Corpora

Corpus	Jazz (training)	Jazz (test)
# Files	40	43
# Tracks	239	252
Melody	37	41
Melody + Solo	23	18
Solo	23	22
Harmony	62	71
Bass	39	43
Drums	39	43

3 EXPERIMENTS

Early experiments showed that using all descriptors gives poor results, leading us to experiment with one descriptor category at a time, such as Pitch or Notes, or a combination of two categories. This approach led us to another problem. Which category combination was better? And which single descriptor combination? Testing every possible combination couldn't even be an hypothesis, for a very large number of combinations can be made out of all the thirty seven single descriptors.

A very simple algorithm solved the problem. The set of descriptors is built by testing each descriptor individually and joining iteratively more descriptors to the set while the performance increases. Section 3.3 shows some results obtained with different sets of descriptors.

The Matlab environment was used to implement the system and to perform the experiments. An additional toolbox was used, the MidiToolbox for helping with the handling of MIDI files in the Matlab environment. Matlab was chosen because it already sports a vast array of functions, classifiers, graphics, plots, etc, which help in analyzing the MIDI files.

[1] This music corpora is available for download at http://www.di.fc.ul.pt/~il/ICEIS2008/

3.1 Track Selection: all descriptors

In the first set of experiments, all thirty seven descriptors were used, which proved to be a naive approach. Several NN were used, with hidden units ranging from 40 to 80, and they all presented basically the same results, varying only in 3%, so the best network was used, with 80 hidden units. *k*NN best *k* value was 7. The confusion matrices obtained with both methods are shown in tables 3 and 4.

Table 3: Confusion Matrix. NN: all descriptors. Classification Rate: 67,8%

Melody	9	2	6	17	7	0
Melody+Solo	0	3	6	8	1	0
Solo	0	3	15	4	0	0
Harmony	0	0	5	61	3	2
Bass	0	0	0	1	42	0
Drums	0	0	0	1	1	41

Table 4: Confusion Matrix. *k*-NN: all descriptors. Classification Rate: 71,8%

Melody	23	5	3	8	2	0
Melody+Solo	2	6	5	5	0	0
Solo	2	5	15	0	0	0
Harmony	5	3	1	57	3	2
Bass	1	1	0	2	39	0
Drums	0	0	1	1	0	41

k-NN gave slightly better results than NN, but not a significant benefit. Using all descriptors, is a naive approach, because some descriptors may successfully distinguish between two different classes, but another descriptor may distinguish the same classes in an opposite way, and confuse the final classification.

Notice the high score in Harmony, Bass and Drums, this is mostly because these tracks are quite different from each other, and specially from the other three classes. Harmony is polyphonic, as are mostly Drum tracks, in contrast to melody or solo tracks which are monophonic. Bass is also monophonic but usually has a lower pitch than melodies or solos, which makes it easy, for the classifier, to distinguish. It seems that the real problem is classifying Melody and Solo, as these are quite similar, and may be confused with Melody+Solo class.

3.2 Track Selection: single descriptor category

A different approach was used in the following experiments. Instead of using the full set of descriptors, six sets of descriptors were used, corresponding to the descriptor categories, and also some combinations of the best scoring sets. Both networks used, with hidden units set to 80 for NN, and *k* values ranging from 1 to 29 for *k*NN, using the best value achieved. The results are shown in table 5.

Table 5: Single Categories Classification Rates

Set	NN Rate(%)	*k*-NN Rate(%)
TI	71,8	63,8
P	56,7	53,9
PI	50,4	44,4
ND	50,4	42
SD	31,3	30,9
S	38,8	37,6
R	42	40,5
IH	47,2	36,9
TI+P	75,4	71,8
P+PI	63	48,8

With NN, surprisingly, the TI set alone provided better results than the whole set of descriptors. Also, the TI+P set provided the best results so far! All the other sets yielded worse results and are clearly confusing the classifier, and should not be used, at least not in this naive way. The *k*NN results, using set TI, were worse than those achieved when using the full set of descriptors, but using TI and P combined gave similar results. As in NN, all the other categories gave worse results. These results proved that the descriptors have to be carefully selected, not only by combining categories, but combining single descriptors, in order to achieve the best results.

3.3 Track Selection: best descriptors

Using the algorithm described previously a possible best descriptor set was found. The best set is composed of descriptors [1 2 4 5 7 8 9 15 18 24 31 34] (numbers are correspondent to table 1) for NN using 60 hidden units, and [3 4 5 6 7 8 9 11 13 18 19 24 25 31 33] for *k*NN with $k = 5$. It's clearly obvious that using a whole category is not the best option. Instead, using only the descriptors that work better together. For NN, which is the main classifier, a significant 16% gain was achieved comparing with the full descriptor set.

As we can see, only the descriptor #3 was not chosen from the TI set, which makes sense, as it was the set that provided the best results alone. From the P set, Highest Pitch was not chosen, but Lowest Pitch was, as it's used for classifying the bass tracks. From the PI set, only the Standard Deviation was used and from the ND set, only Mean was chosen. The SD set was ignored by the algorithm, which means that the silences are not significant and could be discarded. S was also chosen, meaning that the rhythm is also an important feature in distinguishing between classes.

The descriptors chosen from the IH set, were "Perfect Fourth" and "Minor Sixth". According to music theory, there intervals are one of the most consonant,

because they have simple pitch relationships resulting in a high degree of consonance, which is perfect for distinguishing between, for example, a simple slow Melody or a fast complicated Solo.

In respect to the confusion matrix, all the misclassified tracks make sense. A melody track is similar to a bass track, although it has higher pitches. In fact, that one misclassified bass track has higher pitches as well. The Melody+Solo class is the worst performing, mainly because a solo can be made of several melodies, or even harmony at some point, and be misclassified. More training data would definitely improve the performance on this class.

Table 6: Confusion Matrix. NN: best descriptors. Classification Rate: 83,7%

Melody	31	4	2	4	0	0
Melody+Solo	2	13	1	1	1	0
Solo	0	4	17	1	0	0
Harmony	2	0	0	69	0	0
Bass	0	1	0	0	42	0
Drums	0	0	2	1	1	39

Table 7: Confusion Matrix. KNN: best descriptors. Classification Rate: 77,3%

Melody	28	4	3	4	2	0
Melody+Solo	2	10	5	1	0	0
Solo	4	2	14	1	1	0
Harmony	5	1	3	60	1	1
Bass	0	0	0	0	43	0
Drums	0	0	1	1	1	40

4 CONCLUSION AND FUTURE WORK

An Automatic Classification of Midi Tracks system has been implemented and presented. It uses MIDI files from a single genre, Jazz, and classifies the tracks in six classes, Melody, Melody+Solo, Solo, Harmony, Bass and Drums. A neural network is used to process thirty seven descriptors extracted from each MIDI track, which has been previously tagged in the six classes. The experiments showed that using all descriptors is a wrong approach, as there are descriptors which confuse the classifier. Using carefully selected descriptors proved to be the best way to classify these MIDI tracks. Future work, under research now, includes using a larger MIDI database, testing new genres, such as Rock, Pop or Classical, to prove the systems reliability between all genres, so that it can be used as a crucial part of a larger musical genre classification system. Having the tracks identified, as we have presented here, allows different processing specialized to each class.

ACKNOWLEDGMENTS

This work was supported by EU and FCT, through LaSIGE Multiannual Funding Programme.

REFERENCES

Basili, R., Serafini, A., and Stellato, A. (2004). Classification of musical genre: a machine learning approach. In *ISMIR*.

Cataltepe, Z., Yaslan, Y., and Sonmez, A. (2007). Music genre classifcation using midi and audio featres. *Journal on Advances in Signal Processing*, 2007.

Cilbrasi, R., Vitányi, P., and de Wolf, R. (2004). Algoritmic clustering of music based on string compression. *Computer Music Journal*, 29(4):49–67.

D., R., de León P. J., P., C., P.-S., and Pertusa A., I. J. M. (2006a). A pattern recognition approach for melody track selection in midi files. In Dannenberg R., Lemström K., T. A., editor, *Proc. of the 7th Int. Symp. on Music Information Retrieval ISMIR 2006*, pages 61–66, Victoria, Canada. ISBN: 1-55058-349-2.

D., R., de León P.J., P., and Pertusa A., I. J. (2006b). Melodic track identification in midi files. In *Proc. of the 19th Int. FLAIRS Conference*. AAAI Press. ISBN: 978-1-57735-261-7.

Huang, Y.-P., Guo, G.-L., and Lu, C.-T. (2004). Using back propagation model to design a midi music classification system. In *International Computer Symposium*, pages 253–258, Taipei, Taiwan.

Li, M. and Sleep, R. (2004a). Improving melody classification by discriminant feature extraction and fusion. In *Proc. of the 5th Int. Symp. on Music Information Retrieval ISMIR 2004*.

Li, M. and Sleep, R. (2004b). Melody classification using a similarity metric based on kolmogorov complexity. In *Sound and Music Computing*, Paris, France.

Madsen, S. T. and Widmer, G. (2007). A complexity-based approach to melody track identification in midi files. In *International Workshop on Artificial Intelligence and Music (MUSIC-AI 2007)*, Hyderabad, India.

McKay, C. and Fujinaga, I. (2004). Automatic genre classification using large high-level musical feature sets. In *ISMIR*.

Ruppin, A. and Yeshurun, H. (2006). Midi music genre classification by invariant features. In *ISMIR*, pages 397–399.