POLITECNICO DI MILANO
Scuola di Ingergneria Industriale e dell'Informazione
Corso di Laurea Magistrale in Mathematical Engineering

Individual semantic modeling for music information retrieval

Image and Sound Processing Group

Relatore: Prof. Augusto Sarti
Correlatore: Dott. Massimiliano Zanoni

Tesi di Laurea di:
Pietro Ansidei, matr. 835964

Anno Accademico 2016-2017

Contents

List of Figures

List of Tables

Sommario

De gustibus non disputandum è un celebre modo di dire. Questo è soprattutto vero nella musica, campo in cui gli individui presentano gusti differenti e li esprimono perlopiù con l'utilizzo di termini diversi, anche nel caso le preferenze coincidano. Poiché uno dei più importanti obiettivi dell'industria musicale è la raccomandazione di contenuti audio, questo pone un problema di modellazione di contenuti e di coerenza tra sistemi di rappresentazione appartenenti a persone diverse. Lo scopo di questa tesi è pertanto di costruire un modello personalizzato di descrizione dei contenuti musicali e renderlo tale da garantire la comparabilità tra individui differenti.

Questo lavoro sarà compiuto considerando un dizionario per ogni individuo, creato sulla base dei termini che quest'ultimo utilizza per descrivere i suoi gusti musicali, ed estrarre dalle caratteristiche acustiche delle canzoni una misura specifica, che tenga conto della similarità tra ogni parola del dizionario. Questo processo si svolgerà attraverso una procedura di *machine learning*, la quale implementerà un algoritmo teso a derivare dai dati una correlazione di tipo non lineare. Una volta riuniti i modelli soggettivi nella forma di *componenti principali*, sarà possibile paragonarli e, di conseguenza, mettere in connessione profili che mostrino interessi simili, anche qualora questi si manifestassero attraverso sistemi semantici differenti.

La procedura qui descritta sarà utile, in un passaggio successivo, per creare un nuovo sistema di raccomandazione, basato sul *collaborative filtering*, la cui resa sarà migliore rispetto a un approccio solamente basato sulla corrispondenza nello storico degli ascolti individuali. Inoltre, potendo integrare questo modello con metadati relativi agli utenti, sarà possibile affrontare due problemi noti dei sistemi di raccomandazione allo stato dell'arte: il problema del *cold start*, che consiste nell'assenza di dati iniziali per la modellazione di un utente nuovo al sistema, e il comportamento nella *long tail*, inteso come la possibilità che il sistema non sia in grado di produrre suggerimenti di ascolto agli utenti con un buon grado di innovatività rispetto al loro storico.

Abstract

De gustibus non disputandum is a popular common saying. This is most true in music, with people having different tastes and expressing most of them by using different words. Often different users use the same words to descrive slightly different concepts. Since one of the major tasks in music industry is recommendation of songs, this poses a problem of concept modeling and of coherence between the representation models of different people. The aim of this work is thus to model the way people personally describe music and make it in such a way to grant comparability between different individuals.

This will be done by considering a dictionary for each person, made by the words he or she uses to describe music, and extracting from the computable characteristics of songs a specific measure of their similarity related to each term in the dictionary. The process will be exploited through a machine learning procedure implementing an algorithm which derives a non-linear correlation out of the data. Once gathered the subjective models in the shape of principal components, it will be possible to compare them and connect people showing similar interests, even when these are addressed with different semantics.

The described procedure can be useful in a successive step to generate a new recommender system based on collaborative filtering, which is supposed to improve with respect to an approach based solely on the correspondance of the songs listened. Moreover, by integrating the models with users metadata, it is possible to smoothen two known issues of the architecture: the *cold-start* problem, which consists in the lack of data for modeling new users, and the behaviour in the *long tail*, meaning the possibility of the system failing to provide enough innovation in its suggestions.

Chapter 1

Introduction

Since the beginning of the twentieth century, the western world has been changing habits with respect to music consumption: the widespread diffusion of new technologies, like the radio or the phonograph, allowed it to become popular and commercial. In particular, the improved economic conditions following the end of World War II facilitated the creation of a wealthy music industry that conceives not only the artistic aspect but also looks to the audience as consumer.

At its early time, the music market deeply involved interactions between listeners themselves as well as with experts: the discovery of new artists or genres happened either through word of mouth or the media. The record stores played a major role in this situation, because their task consisted in *intercepting* the tastes of the customers and *provide* them with good advice by understanding their input. The Digital Revolution brought this market along since the late nineties with the creation of content-sharing web platforms like *Napster*: the unexpected success of the internet as a cheap and fast delivery method made the industry exploit it for its business. More in general, the widespread diffusion of e-commerce and streaming services made the market itself change: its consumers themselves are able to look in any moment for the product they desire and obtain it without moving from their desk. A few relevant examples for that can be *Amazon*[1] for shopping, *Youtube*[2] for video entertainment and *Spotify*[3] for music contents.

Supposing the reader already came into contact with one of these services, it is immediate to notice a difference with the offline way of doing shopping or musical discovery: automatic systems substitute the human interaction in collecting information about the listeners' consumption. This knowledge will be used to elaborate tastes, habits and musical attitudes in order to provide the user of the system with accurate suggestions. The reciprocal need

[1] https://www.amazon.com/
[2] https://www.youtube.com/
[3] https://www.spotify.com/

between customers and suppliers of having someone improving satisfaction has been answered in the digital world with the conception of *recommending softwares*, which analyse customers' choices and provide suggestions for new content (called *items*) to be made use of. Explicitely focusing on the audio sector, in order to do so they exploit models which have been studied and implemented by the *music information retrieval* (*MIR*) field of research.

Music can be defined as *humanly organized sound* [12]. This characterization entails the integration of multiple aspects within the concept of music: a first regarding *sound* as a phisical phenomenon, a second concerning the *human perception* of music events and a third involving the cultural elaboration of music knowledge. Thus, the purpose of music information retrieval is to learn how to automatically extract knowledge from music by conducting analysis on those different layers of abstraction. The obtained information spans several aspects, aimed to model the different ways in which people perceive and describe music. The resulting models may eventually be used to automate a wide range of processes, like music recommendation, transcription or emotion recognition. The task involves several fields of research, including statistics and signal processing as well as musicology and psychology. Indeed, it comes easily apparent how a scientific approach to the rules of music has to be matched to a world made of subjective perceptions, feelings and interpretations.

This heterogeneous community works with tools given by informatics: sound can be digitised as an audio data stream, then processed in order to elaborate and possibly modify its properties. These not exhaustively include rhythmic, timbric, melodic and emotional aspects, which are encoded as *features*. A *feature* is a measurable quantity which can be derived from the audio data to numerically capture one of the characteristics of sound. Some of them, referred to as *low-level*, are objective because strongly related with the physics of sound, so they appear easy to be computed from the audio signal but lack of abstraction, for instance the amplitude of an audio signal. Others, dealing with more structured and subjective concepts, are easier to understand by a listener but harder to define in a rigorous mathematical way, because they refer to elaborations over perceptual elements: they are thus called *high-level* features, an example of which is the rhythm of a song. For this reason, it is common in literature to connotate features with those different *levels* of abstraction. An intermediate category of *mid-level* features can be identified for concepts which are useful in order to raise the level of abstraction. In particular, lower level input features are transformed into representations that have some desireable properties such as compactness, sparseness or statistical independence [35]. For instance, the amplitude of an audio signal may be analyzed in its temporal patterns (mid-level feature) in order to identify the rhythm of a music excerpt.

Howbeit, when talking about mid- and high-level features, signal processing should leave the field free for contaminations by other disciplines,

more capable to describe or infer ideas which are closer to the human understanding of music. In particular, the integration between objective and subjective aspects of music makes necessary to recognise how much individual perception affects the scene, in order to put the accent on the personal interpretation of musical concepts. This gives the motivation for which also psychology and psychoacoustics are disciplines to be looking to: they will allow the interpretation of the observations and put a basis on the work to be done. Only statistical techniques, however, can help in solving the issue of *organising* data into regular patterns and select which of them are *reliable* in order to achieve the higher level representations which will deal with perceptions and interpretations. A big improvement in these directions has been given by the developments in the field of machine learning, the usage of which is fundamental across the work done by MIR community because of the higher level of abstraction it is able to capture. In particular, it constitues the implementation environment for recommender systems.

The creation of a recommending engine requires two main steps: a first devoted to model the available data about the users and the music content; a second about the generation and delivery of suggestions. The modeling phase, which is the one this work is mostly focused on, consists in the data analysis steps which examine the music content in order to investigate the rules of music abstraction. The process starts indeed from low-level observations and proceeds towards computing or infering data correlation structures, which define the higher-level properties. The same rules will then be replicated in a synthesis step, devoted to select the content matching the modeled behaviour of the users for recommending. The matching is measured by some definition of similarity between modeled objects: the various modeling approaches differ on the kind of collected data and the definition of similarity itself. More and more complex models have been built during time, along with the discovered issues and, consequently, the evolution of the hypothesis made by professionals. Those mostly concern which kind of information is actually relevant in order to raise the level of abstraction, from consumption data to actual sound properties and context information.

Important examples of issues addressed in this work are the so-called *cold-start* and *long-tail* problems: these can be explained as *what to do if no information is given about a new user or item in the system* and *how to characterize items in a way that is independent of their popularity.* The first question finds answer into an accurate profiling of users, in order to exploit any information about them, even if not related with music consumption itself. The second problem drops instead a hint about how to manage the users' listening histories in order to determine suggestions. An easy but widespread example of recommender indeed assumes that users experiencing the same items will prosecute in doing so: in this way, items that *co-occur* in the consumption histories of many users immediately become a reference

for other people, generating a *rich-gets-richer* kind of dynamic.

An important aspect to analyse in order to solve these issues is the need of personalization: the modeling algorithm should consider the characteristics of the individual user as the parameters for computing his own model, in order to encapsulate the personal tastes into different realizations of the high-level features of music. The importance of individuality, rather ignored in MIR's early times, has been underlined in many occasions, among which is noticeable the intervention by Arthur Flexer in [22].

Starting from this observation, the model described in this work considers a new way of exploiting users' music consumption data. As mentioned above, the layers of abstraction in music need to be modeled according to personal characteristics in order to acquire the individual nuances into the high level representations, which will identify the users' models. The method described in this work will exploit semantic processing for shaping user tastes. Starting from clusters of songs, identified with a label by each of the users on the basis of a common musical meaning, we will analyse the low-level acoustic features of the audio excerpts included in each group. This will allow to extract a model of the high-level representation for the given musical meaning which will intrinsecally be personalized. The approach detaches from the usual modeling of co-occurrences between songs listened by users, because each user will be free to define and aggregate songs in the way he prefers. These aggregations, which are outcome of independent classification processes and might even be disjoint to each other, will be examined towards defining which of the characteristics of songs are actually relevant for every user's listening trends.

The mentioned procedure will allow to define an acoustic model of labels, identifying a user by the set of his generated label models. It will thus be necessary to identify concepts of similarity linking to each other the labels and the users respectively: our work will compare the metrics identified on the low-level features space for each label of each user, with a similarity metric defined on linear projections. Finally, user models will be compared considering a concept of similarity based on* two main aspects:

1. the presence of similar labels showing different acoustical meaning;

2. the presence of different labels showing similar acoustical meaning.

The impact of this approach over recommending quality should be apparent: users' models will no more consider only the presence of songs into categories, but giving also an acoustic explaination about why they fit into, improving the user profile with significant additional information. Moreover, the model here presented is intended to revert a common approach in music tagging: usually, in fact, genre labels like *blues* are looked to as having a kind of universal meaning, derived by the so-called *wisdom of crowd* [21] or out of the opinion of music experts [30]. This reduces personal differences to

be shades in the interpretation of those concepts, whereas in the present work they become the starting point to build an effective individual model, following a need which is strongly perceived in the MIR field [16, 23, 24].

The model has been built and tested on a real listening setting by exploiting CAL500, a well known music database in MIR literature [20]. The model showed important results in assessing both user similarities and label relationships by considering also different individuals, towards the directions explained above. This will allow an application within a real recommending environment: a model for building label suggestions has been developped and tested, running in particular for songs that have not been tagged by users. Moreover, simulations have been done exhibiting the capability of the system to distinguish between real user tags and randomly generated ones. This result is particularly surprising, since it proved that individual semantics actually plays a role in music feature modeling. Another important fact is that working on music genres as clusters of songs was a choice due to the huge amount of literature already facing the theme: this method is sufficiently general to be used for any kind of musical personal labeling, thus entailing possible applications in other fields of MIR (e.g. music emotion recognition).

This method could furthermore be exploited in order to have user models linked with metadata coming from other sources. Indeed, user profiles can be grouped into relevant consumption categories, to be possibly correlated with some side information, which are already tipically collected by music streming providers. Once proven this correlation to exist, as it happens in commercial services, the model could provide a first estimate for tastes profile of users even when no listening data is available, thus giving a hint in the direction of solving the cold-start problem. In this setting, the same lack of information would be overcome regarding content data: songs feed the system only as a collection of their acoustical features, which are always available for any item in a catalogue, in contrast to listening data. Songs importance in defining the model is not weighted anymore on the number of listenings they receive, thus being uncorrelated with respect to their popularity, addressing also the long-tail problem. The described algorithm owes a user-distributed structure with satisfactory computational performances, making the method efficient in both running time and memory. Those explained are the reasons to suppose that such a system could easily find application in state-of-the art commercial services for music recommending purposes.

The work is organized as follows: chapter 2 shows an overview of the state of the art for recommender systems inside and outside the MIR field, describing different approaches and applications of similarity models. Moreover, the growing importance of studies in the semantic direction and the role played by personalization in music research will be examined. Chapter 3 contains a detail of all the technical tools which have been exploited for the development of the present work, focusing on both signal processing and

mathematical aspects, with a detailed description of the nature of the collected data as well as the main statistical framework. Chapter 4 will then explain the model with a full mathematical formalization along with the issues faced within data collection and management. This lead to the experimental results contained in chapter 5. A last section is devoted instead to the conclusions and the next steps which could possibly be performed towards model improvement and implementation in a relevant environment.

Chapter 2

State of the art

2.1 Music Information Retrieval: foundations and trends

Music information retrieval *(MIR)* is a discipline which aims to understand and use of music data through computational approaches and tools. In particular, it deals with the research, development and application of models for music description based on combining theories and techniques from a wide range of disciplines like musicology, computer science, signal processing and cognition.

Music information for retrieval is encoded in the shape of *features*, which are variables representing sound properties with different levels of abstraction. At the lowest level, they consist of numeric values which are output of functions applied to the audio data stream. They can capture basic but fundamental observations about the sound signal energy and shape, which only sometimes translate into an understandable music property. The introduction of a layer of abstraction in the analysis leads to grouping these outputs into more complex data structures, known as mid-level features. These are usually computed by observing the dynamics of a plurality of low-level features during time, aiming to a deeper description of the signal. This characterization can be translated into fundamental music elements like *notes, frequencies, intensities*, thus owing a first perceptual meaning. At their time, mid-level features are grouped or elaborated through mathematical models in order to build semantically higher level content. This is strictly linked to human representation of music in any of the perceptual, knowledge or emotional points of view. For instance, a succession of notes or chords may be embedded in the concepts of *melody* and *harmony* respectively [4], which characterization (*major* or *minor*) affects the emotion conveyed by a song and, along with their rhythmic pattern, may lead to genre recognition.

Mid- and high-level information assume different understandable shapes for human interpretation: it can be based on scores or directly sound (e.g.

notes), being collected through surveys (genre, emotion), by means of words or symbols, and scores themselves may either be physical or digital (e.g., MIDI). This variety is due to the different objectives in music description, all of which find an application on the respective research branch in MIR. It is as well worth mentioning that many of these objectives represent not just pure theoretical speculation, but find a relevance in correspondant industries which can span from audio identification (*Shazam*[1], *Gracenote*[2]) to music recommendation and playlisting (*Pandora*[3], *Spotify*[4], *Amazon Prime Music*[5]), score following (*Rock Band*[6]), music instruments and reproduction systems (*ROLI*[7], *Steinberg*[8], *Bose*[9]) and many others.

An example could be useful in order to understand the contaminations between disciplines happening in the MIR field, dealing with both mathematic and technical issues as well as humanistic and psychological aspects. Consider the problem of genre recognition, meaning the identification of a high-level feature, called genre, which is able to capture and describe the acoustical properties of songs through labels [3]. This connotation should attribute a semantically meaningful word for how the songs *sound like*. By the point of view of raw data processing, this can be translated into the detection of a specific behaviour in the energy of the signal, including rhytmic patterns describing *the displacement in time of the loudest instants of the song* along with timbric, melodic or harmonic characterization. This is clearly insufficient in order to define properly the concept of genre, which is influenced also by emotional and cultural aspects. Moreover, a song is an organic whole of performances coming from different sources, the behaviour of which may considerably differ from each other: genre is deeply influenced by this partecipation, think for instance to a brazilian *samba*. Thus, the proper way in order to obtain a full knowledge of the acoustic phaenomenon results into the application of statistic techniques embedded into machine learning models. This allows a computer to make automatic inference on the music data and to give the desired answers in a human-readable way. Particular importance has been recently acquired by data representation learning techniques, the usage of which is more and more spreading across the MIR community [13, 14, 15].

Back to the genre recognition example, it is possible to understand how much individual perception affects the scene. It happens, for instance, to disagree on the mood perceived while listening a song. That happens because

[1] https://www.shazam.com/
[2] http://www.gracenote.com/
[3] https://www.pandora.com/
[4] https://www.spotify.com/
[5] https://www.amazon.it/gp/dmusic/promotions/AmazonMusicUnlimited/
[6] https://www.rockband4.com/
[7] https://roli.com/
[8] https://www.steinberg.net/en/home.html
[9] https://www.bose.it/

people do not have the same perceptions and reactions because of what they listen to. This gives the motivation for having also psychology, and psychoacoustics in particular, as disciplines to be looking to while working. Here a personalization of the algorithms is needed, since the training of a machine learning model should include also variables related to the subject of the experiment, as a parameter of the problem to be solved. These variables could include for instance some individual preference expression as well as perceptual or emotional characteristics of the human user.

The recent achievements in computer science in terms of computability, meaning memory availability and processing speed, allow to manage a great amount of data. This enlarged the possibilities of research in MIR, by using the tools of machine learning, and this reflected into a growing interest in the field. The International Society for Music Information Retrieval Conference (ISMIR) is an annual conference that focuses specifically on the area, being an excellent source of cutting edge MIR research. Also of particular interest, the Music Information Retrieval Evaluation eXchange (MIREX) is an annual competition associated with ISMIR where various approaches and algorithms are compared using the same set of data. Due to its highly multidisciplinary character, MIR research is also published in a wide variety of other conferences and journals that were the source of inspiration for this work.

The current applications of MIR include manipulation and creation of music: along with recommender systems, it is involved in track separation and instrument recognition from recordings, which consists in splitting the individual sources of sound in a song; automatic music transcription, meaning the transscription of an audio excerpt into a musical score, music categorization on the basis of cultural and emotional aspects and music generation, which is a task similar to rhythm detection but furthermore includes the analysis and reproducibility of melodic and armonic patterns.

2.2 Recommender systems in MIR

A recommender system is a software which proposes to *users* of a digital service the *items* they may like through the usage of data analysis techniques. Items is a general word meaning any content, a product to be bought on a shopping website as well as a song or a video to be played. These systems operate by focusing on two main approaches:

- **content-based** systems are the ones which exploit a concept of similarity between the available objects based on their properties. They suggest objects which resemble or are connected the most to others that the user already experienced in his interaction with the system itself;

- **collaborative filtering** methods exploits instead similarity between the users, obtained by grouping people because of the interest they showed on the same objects. These systems tend to suggest items that were already experienced by compatible users, based on the context.

Recent research showed that a combination of these two approaches could improve the efficiency of the recommendation. This happens usually by unifying within the same predictive model content analysis and user profiling. These systems are referred to as *hybrid*.

Whatever the choice of the method, the evaluation of similarity may not depend only on the objective properties of the items, but it can also deal with a preference expressed by the customers themselves. This can come from a two-level liking feedback, with the user declaring whether he liked the item or not, as well as a multi-level ranking scale quantifying how good that experience was.

Up to now, the described methods involve an active participation of the user, who is supposed to answer an *explicit* question posed by the software. For instance, consider the 5 stars ranking by several services on the web or the like/dislike feedback on a video. This kind of direct interaction often happens to be uncomfortable or even impossible. Hence, other *implicit* ways of collecting informations should be found. A new perspective regarding this has been given in [18], where the authors suggested that the quality of recommendations could be improved by considering also the interest elicited on the user instead of just the given ranking. In this case, a good index for the pleasantness of an item is represented by the stimulus to interact induced on the user.

It is important to give a glance to a possible implementation of a recommender system. This will also allow to understand the main kind of troubles that may come out in its design.

Consider the sets $\mathcal{U} = \{u_1, u_2, \ldots, u_{N_u}\}$ of all the users and $\mathcal{I} = \{i_1, i_2, \ldots, i_{N_i}\}$ of all the items, where $N_u = |\mathcal{U}|$, $N_i = |\mathcal{I}|$, both finite. Their cartesian product set $\mathcal{U} \times \mathcal{I}$ is made out of all the possible user-item pairs (u, i), so it is possible to associate a value to each of this pairs, representing (if it exists) the evaluation that user u gave to item i. For instance, we can define a two levels function $eval : \mathcal{U} \times \mathcal{I} \to \{0, 1\}$ such that

$$eval(u, i) = \begin{cases} 1 & \text{if } u \text{ experienced } i \\ 0 & \text{otherwise} \end{cases} .$$

The most immediate way to visualize the data is to store them into a user-item matrix; each of its rows represents a user and each column stands for an item. In this case, the pair (u, i) represents the index of the elements. The ranking matrix will be used to produce a list of recommended items for

a given user: the algorithm must be able to predict a numerical value expressing the predicted likeness of any item that the active user has not rated, within the same scale as the provided opinion values. At this stage, the definition of a similarity measure becomes necessary and this will be the basis for rating every (u, i) pair. The items that owe the highest similarity values, called *neighbors* of the rated elements, will then be provided to the final user through a suitable interface. Different techniques are implemented in literature: for instance, content-based filtering could use vector space models such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier [6], Decision Trees [7] or Neural Networks [8] to model the relationship between different items within a corpus. Collaborative filtering algorithms instead matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations; in this case the neighborhood is built out of similar users thanks to two techniques called *memory-based* and *model-based* [9]. Memory-based techniques calculates similarity between users by comparing their ratings on the same item, and it then computes the predicted rating for an item by the active user as a weighted average of the ratings of the item by users similar to the active user where weights are the similarities of these users with the target item [10]. Model-based techniques quickly recommend a set of items for the fact that they use pre-computed model and they have proved to produce recommendation results that are similar to neighborhood-based recommender techniques. Examples of these techniques include Dimensionality Reduction technique such as Singular Value Decomposition (SVD), Matrix Completion Technique, Latent Semantic methods, and Regression and Clustering. Model-based techniques analyze the user-item matrix to identify relations between items; they use these relations to compare the list of top-N recommendations [5].

It is easy to figure out that a user could have experienced and ranked only few of the objects which are present in the system, so we can suppose the ranking matrix to be highly sparse. Moreover, when considering large environments, the number of users and items runs about millions, while the number of entries of the matrix is the equal to $N_u \cdot N_i$. So, the quantity of zeroes (missing informations) within the system could easily be around thousands of billions. This is useless and detrimental, since it represents a waste of memory and computational power, which translates into worst performances for the algorithm. The speed in providing an answer and updating the system is in fact one of the most important issues that goes beyond the accuracy of the recommendation, thus this kind of *data sparsity* is to be considered necessarily in the design.

One of the problem children of data sparsity is the so called *cold-start*. It consists in the difficulty of generating a trustable recommendation for a user who has no ranking history in the system. It happens because any similarity measure is computed starting from data, that in this case are not available.

Another still open issue linked to poorness of data is the behaviour of a recommender with respect to the *long tail*. In order to clarify what the long tail is, consider a database made of all the music ever written, each piece of which is labeled with the number of times it had been listened to. Some popular songs will have great listening values, while many others, not to say almost every, would rather have little, if not zeroes. However, the fact that *many* songs have little individual impact means as well that their contribution as a mass is relevant with respect to the overall sum of the values. This is to be taken care of mostly when using the recommender system for discovering new items, since a collaborative filtering method could suggest to many users things that already have great individual impact. This can create a rich-get-richer effect for popular products known as *bias towards popularity*: a good recommender should improve serendipity in case the user shows interest to explore new, unpopular items.

2.2.1 The similarity rush

All of the recommender systems work by expressing and using a definition of similarity, which can be referred either to the users or the items. Of course, different definitions may lead to different estimations and thus to different results. This poses a problem in terms of finding a good similarity relation, which means that an investigation on this concept has to be performed.

The similarity is based on an hidden assumption, which consists in the belief that *entities behaving in the same way will continue to do so*. Actually, this statement is essential in order to produce forecasts for the future: it would not be otherwise possible to infere from data a brand new behaviour, lacking any relation with the past. This already gives an hint about a stability property of the function of data to be used, but as well poses another issue to be discussed: what amount of stability is actually desirable and how much this represents instead an unwanted constraint to flexibility?

The first music recommenders expressed similarity in terms of *simultaneous occurence* [34]: two songs are defined as similar if many users either listen to both of them or group them together, as well as two users listening to the same songs are supposed to have the same tastes. The *cosine similarity* between two items i and j can be defined as follows [11]:

$$s(i,j) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\| \left\|\vec{j}\right\|} = \frac{\sum_{u \in \mathcal{U}} r_{u,i} r_{u,j}}{\sqrt{\sum_{u \in \mathcal{U}} r_{u,i}^2} \sqrt{\sum_{u \in \mathcal{U}} r_{u,j}^2}}, \tag{2.1}$$

where $r_{u,i}$ is the ranking given by user u to item i and $\mathcal{U}$ is the set of all the users.

Even if this kind of definition may appear reliable and coherent with data at first sight, it is true that this statement appears to be somehow misleading. This happens when confusing the concept of simultaneous occurrence with

correlation: we do not have an a priori clue to state that two songs have an objective reason to be grouped together just because some user did it. In this way, we can neither assume that songs listened by the same users actually sound similar, nor that different users listen to the same songs for the same motivations. Thus how can we be sure about the coherence durig time of such a system? An example of similarity model solving this issue is the *Pearson correlation*, which is expressed by the following formula [10]:

$$s(i,j) = \frac{\sum_{u \in \mathcal{U}} (r_{ui} - \overline{r_u})(r_{uj} - \overline{r_u})}{\sqrt{\sum_{u \in \mathcal{U}} (r_{ui} - \overline{r_u})^2} \sqrt{\sum_{u \in \mathcal{U}} (r_{uj} - \overline{r_u})^2}}, \tag{2.2}$$

where $\overline{r_u}$ is the average rating of the user u and $\mathcal{U}$ is the set of users that rated both items i and j.

Arthur Flexer in [23, 25] faces the point of inter-rater agreement in music similarity and precisely the general notion of "sounding similar" is a central point of criticism in his papers. Those assess that the idea of similarity is actually very different between people and almost individual, since depending on complex multi-dimensional notions like "timbre, melody, harmony, tempo, rhythm, lyrics, mood, etc.". This goes in contrast with a simple, universal definition of similarity like the one coming out of correlation analysis. Indeed, Flexer declares most of studies exploring music similarity as being restricted to simple overall similarity judgments, even when using human listening tests, thus *"blurring the many important dimensions of musical expression"*. In order to prove this concept, he showed that there is a low inter-rater agreement due to the coarse concept of music similarity. Impressively, there exists an upper bound of performance that can be achieved by algorithmic approaches to music similarity: this has been already reached and never surpassed. His conclusion points out that it would be necessary for research to focus on what music similarity actually means to human listeners.

The concept of similarity is strictly linked with the one of distance, which can be conceived as its opposite. This means that a change of perspective is possible, where, instead of looking to what can make entities *similar*, it is possible to consider what generates difference or, better, *distance*. This can be really useful, considering that the distance is a well-known and formalized mathematical concept, whereas similarity is not. In [21], for instance, the authors face the issue of similarity learning from collaborative filtering, by exploiting the so-called "*wisdom of crowd*" and trying to define a distance metric on items by exploiting user generated knowledge. This paper points out that there are no assumptions of transitivity or symmetry in the learned similarity, even if starting from a distance, which of course owes these properties. It concludes with an improvement of usual content-based similarity models. An important issue about these kind of definitions is the usual carelessness in using the distance properties: as it will be shown in Chapter 3, Mahalanobis' metric is the most comfortable to implement and manage,

but it owes characteristics which make it not that general and unsuitable to model some kind of phaenomena. Unluckily, a deep analysis on how the properties of a distance translate into the ones of a related similarity metric is often unmanageable because of nonlinearities.

Wolff and Weyde's [19] deals with modeling music similarity with respect to user perception. Actually, the work makes no effort in the direction of individuality, since never mentioning a model of perceived similarity, but it follows the main principle of adapting the similarity metric to subjective data. Nevertheless, it is worth to mention that the authors declare a similarity metric to be "not necessarily a linear, positive definite and symmetric function which satisfies the triangle inequality". This considerations allow to fancy a deeper exploitation of personal usage data to model similarity.

2.2.2 Semantic modeling

A first step for automatic learning systems towards interaction with users' perception has been reached by giving a meaning to particular combinations of musical features. In fact, once recognised the approach solely based on co-occurrences is insufficient, a deeper analysis of the music content is needed. This reflects into considering measurable acoustical features as the real data to be looked at, while similarity between songs becomes actually a search into the behavioural rules of music according to listeners.

Statistical analysis over music data lead to the concepts of middle- and high-level musical features. These characteristics capture complex elements in music like timbre, rhythm, melody or harmony. The feedback to this process can only be perception-based and it is just in this phase that actual signification is provided to the mathematical definitions, since human experience becomes a formal description of data. This fact introduces the presence of a semantic aspect in recommending, and the relations between concepts and their meaning should be translated into either a similarity or distance formalization. In particular, when the recommending paradigm is based on tagging songs with labels, like in genre or emotion recognition, this opens new possibilities for model design.

Orio and Piva in [28] present a study on the contribution of timbric and rhythmic features for semantic music tagging, starting with the assumption "that acoustically similar songs have similar tags". This work operates with a given universal dictionary for music. In particular, it assigns multiple concept labels to songs by weighting how much those concepts are compliant with the acoustic properties of the audio excerpts. This may sound reductive, since they also state that labeling is not a universal procedure, but relies on subjective judgements. Starting from this principle, Bogdanov and Herrera [26] work with a collection of music chosen by the final user, called *preference set*, in order to infere a set of high-level semantic descriptors to be correlated with songs metadata, meaning information that is not related with the audio

signal, for instance the performer of a song. In order to do so, they compare a content-based distance, based on those descriptors, with another considering also rhythmic and timbric properties. Finally, they decide to exploit artist based metadata within the model. Even if proving the convenience of such an approach, it comes apparent how the usage of artist metadata gives a restriction to the acoustical variability that can be noticed among songs by the same author.

[20] describes instead semantic issues and purposes a similar solution still based on a predetermined dictionary. The vocabulary is connected to songs by weights coming out of a multinomial distribution. This paper faces also the problem of querying by tagging, e.g. making an inquiry on a database by using labels; in this case the semantic of the query is also inspectioned, while the goodness of the response is computed by comparison of the semantical weights for the query and the song. This sounds optimal regarding predictions, but keeps open the issue about modeling, since it also assesses the existence of a problem due to the predefined taxonomy. Another open point for discussion is the assumption that feature vectors are conditionally independent given the label, since it seems clear that it is just feature dependency generating the labels. This paper is as well rather important among the MIR community, becuase it describes *CAL500*, a music dataset which became a reference standard for research in the field and for this reason is used for the development of this work.

2.2.3 The issue of personalization

Considering the impact of human interaction with MIR systems, it comes necessary to care about the final user in order to obtain proper recommendations. It is again Arthur Flexer in [22] pointing out that, whenever computational models are used to describe the human perception of music, the existence of an objective reference *ground truth* is assumed at least to evaluate the models' performances. This objectivity however argues with the principle of individual similarity stated above, thus an efficient system shoud be aware of the aspects influencing what a person perceives as similar.

A personalized system should incorporate information about the user into its data processing part and this information should rely on music content, music context and user context. The latter is an extremely important contribution of Flexer's work, since it justifies the discussion about the temporal adaptivity of the system: this should reflect the dynamic behaviour of users' contexts. All of this makes pretty difficult to state which variables could be most important in affecting similarity perception [23]; the problem then shifts in the definition of a user model which can implement with flexibility even complex factors.

Music recommenders have been linked to the research on music tastes in social psychology [31]. In particular, music taste profiles had been generated

in research in order to assess independent dimensions of users' preferences and their individual dependence on musical attributes. This lead to the conclusion that tastes were dependent both on demographic and social variables – like *gender*, *age*, *country* or *social class* –, but also on individual characteristics like *personality traits* or *beliefs*. While the first are already known in the field of recommending [29] and exploited in current market services, the exploration of the personal perception has not gone too far up to now.

If the user interaction is collected by meaningful tags, semantics could provide a solution for the issue of modeling, since it is the human way of providing a denotation to abstract concepts. When providing labels, every user is already able to define what is similar (since it is defined with the same label) and what is not by his own perception, because the user characteristics of both cultural and individual nature reflect themselves into the choice of the items to be similar. This leads to the possible generation of a profile based on the distance the user itself defines between items, not taking care of how this distance was built. By adopting a context-based approach on the items, it will then be possible to understand how songs relate to each other according to the user, thus building an appropriate ontology for recommending purposes. In [17], Maleszka et al. conceive user profiles as weighted hierarchical thesaurus. The analysis of connections between concepts is provided in a tree-shaped way: these maps depict ontologies based on the relationship of *belonging* holding between a child node and its root. The tree may vary from user to user; similarity between individuals and possibly user grouping are based on the distance among the weights related to the same tag.

The introduction of a given thesaurus although does not solve an issue proposed by the same authors, which is that user queries could not reflect the real individual information needs. Additionally, the same query may have different meaning for different users. This fact is well known in MIR field and has been deeply studied in the field of Natural Language Processing. Considering the genre recognition problem, for instance, many machine learning experiments owe ground-truths consisting in labels either authored by experts or obtained through knowledge integration. Many authors among the cited [22, 31, 32] agree that genre is an ill-defined concept, while [30] contains an evaluation of non-expert annotations applied to common-sense word relations, showing that the reliability of individual quality in tagging is low. This point should be considered when recommending is based on labels, since music consumers usually are not experts and subjectivity in tagging process is raised because of this, which acts like a bias and can be corrected only by aggregation, which is the principle in collaborative filtering.

All of this contributions lead the way to conceive a new model for hybrid recommendation system, which should at a time:

1. consider properly personalization, by defining individual models for

label similarity;

2. compare individual models to deliver inter-user similarity and proceed with collaborative methods;

3. be content-based in order to relate personal choices to objective music features;

4. exploit the power of semantics in order to build with flexibility the necessary connections.

Semantics here lays in the label which is subjectively assigned to a set of concepts. These are built from data in the shape of aggregating maps, which are individual and independent to each other as clustering rules and thus may be different between people, as well as being completely separate or also overlapping.

The importance of individual semantics has been foreseen in MIR even if not studied yet. The difficult dialog between this field and music cognition has been told in [16], an excerpt of which is reported here: “Whether "rock" is indeed *rock* or *jazz* does not matter – actually, we want algorithms that have the flexibility to also learn that *jazz* is *rock* if we like them to. However, we also have the second research goal of being useful to electronic music distribution systems. Now, in this world, defining a unique ground truth is suddenly very relevant, but you soon realize it is also close to impossible: we have plenty of examples where what some call *rock*, others will call *pop* or *jazz* and so on. [...] Most of our recent research tries to address this paradox: for instance, how tags learned on one dataset generalize to other datasets, how to personalize music recommendations or even letting users define their own personal categories in interaction with the system.”.

Chapter 3

Theoretical review

This chapter introduces a detailed description of the technical tools used to develop the model. These involve both algorithms for sound processing, which constitute the data acquisition part of this work, and the statistical theory and methods which had been applied for data cleaning, processing and modeling. In particular, the first sections are devoted to methods for collecting music features and to a characterization of those data based on both signal processing and perceptual concepts; in a squent section, a full definition of the mathematical and statistic techniques is provided in order to justify the chosen methods. Finally, a section is devoted to the discussion of the mathematical issues faced during this work along with their possible solutions provided in literature.

3.1 Data management

Data management is closely related to the implementation of data mining systems. Although many research papers do not explicitly elaborate on data management, it is extremely important for the correct usage of data. A good preprocessing ensures indeed the data format and quality as well as it improves the efficiency and simplify the subsequent processing. For instance, an accurate feature extraction plays a critical role in music data mining, as we will see in the next section. The actual mining tasks possibly involve data visualization, association mining, classification, clustering, and similarity search. Finally, a postprocessing step is needed to organize and evaluate the knowledge derived from the previous stage. Since postprocessing mainly concerns nontechnical work such as documentation and evaluation, this section will concern the first two parts and will briefly review data management in this context.

Data management concerns the mechanism and structures of how the data are accessed, stored and managed. It focuses on data quality, involving data cleansing, data integration, data indexing, and others [55]. Data *cleans-*

ing refers to cleaning the data by filling in missing values or removing non significant samples, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. For instance, there might be the necessity to set a default value for any missing data to proceed further in the analysis. Once ensured the quality of data, their *integration* consists in combining data obtained from different sources and providing users with an unified view of such data. This process plays a significant role in music data, for instance, when performing genre classification using both acoustic features and consumption data. Data *indexing* refers to the problem of storing and arranging a database of objects so that they can be efficiently searched for on the basis of their content. Particularly for music data, data indexing aims at facilitating efficient content management. Due to the nature of music data, indexing solutions are needed to efficiently support similarity search, due to the high-dimensional nature of the data to be organized and the complexity of the similarity criteria used to compare objects.

Data preprocessing describes the operations performed on raw data in order to prepare for the processing procedure. It includes data sampling, dimensionality reduction, feature extraction and transformation. Data *sampling* allows a large data set to be represented by a much smaller random sample (or subset) of the data. For acoustic data, data sampling refers to measure the audio signals at a finite set of discrete times, since a digital system cannot directly represent a continuous audio signal. *Discretization* instead is used to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. In particular, in music data mining, it finds application whenever working with a discrete amount frequencies bands. *Dimensionality reduction* is a further important step in data mining, since many analysis methods become significantly harder to apply as the size of the data increases. The reduction of dimensionality by selecting a subset of attributes is know as *feature selection*: its goal is to find a minimum set of variables such that the resulting probability distribution of the data is as close as possible to the original distribution derived from all of the features. A good selection may significantly improve the comprehensibility of the resulting models, often building algorithms that generalize better to unseen points, allowing the data to be less noisy and visualized in an easier way. Moreover, it is often the case that finding the correct subset of predictive features is an important issue on its own. Some of the most common approaches, particularly for continuous data, use techniques from linear algebra to project the data from a high-dimensional space into a lower-dimensional space, for instance, Principal Component Analysis, which will be described later in this chapter.

Feature *extraction* refers to simplify the amount of resources required to describe a large set of data accurately. For music data, feature extraction involves low-level musical feature measurements (e.g., acoustic features) and inference for higher-level characteristics of sound (e.g., music keys). An

overview of the feature extraction procedure performed in this work will be done from now on. Feature extraction is usually integrated with feature selection in terms of identifying the appropriate features for further analysis. Variable *transformation* finally refers to a map that is applied to all the values of a variable in order to confer some desired properties to the variable itself. For instance, if only the positive magnitude of a variable is important, then its values can be transformed by taking the absolute value. For acoustic data, a transformation consists of any operation or process that might be applied to a musical variable in composition, performance or analysis.

3.2 Feature extraction

Music acoustic features include any acoustic properties of a sound that may be recorded and analyzed: audio feature extraction is the foundation for any type of music data mining. This is the process of distilling the huge amounts of raw audio data into much more compact representations, which capture also semantically relevant information about the underlying musical content. The difference in abstraction of the information captured in those data structures leads to the characterization of music features in *low-level*, *mid-level* and *high-level* ones.
A common way of grouping audio features is based on the type of information they capture. On an abstract level, one can identify different high-level aspects of a music recording: in particular, the hierarchical organization of sounds in time is referred to as *rhythm* and their hierarchical organization in frequency or pitch is referred to as *harmony*. *Timbre* is the quality that distinguishes sounds of the same pitch and loudness generated by different sound sources. To analyze music stored as a recorded audio signal we need then to design representations that roughly correspond to how we perceive sound through the auditory system. At a fundamental level, such audio representations will help determine when things happen in time and how fast they repeat. Therefore, the foundation of any audio analysis algorithm is a representation that is structured around time and frequency, which will be shown later in this section.

As introduced previously, the objective of this work is to derive a model of the individual concept of similarity for users of an audio reproduction system. This will be realized by modeling personalized high-level features. Indeed, the relationship existing between user's perception of music and acoustic features data or their combinations is to be studied, in order to link the individual music experience to observable characteristics. Bogdanov in [27] shows that low-level features are more suitable to model user preferences than high-level descriptors, thus songs will be described by means of a set of simple audio signal descriptors. This will be represented as functions of the digital audio data, which consist in the sampling of the continuous

signal representing changes in air pressure over time. This makes apparent how deep in meaning can be the low-level description of music, thus, in order to collect the fundamental descriptive aspects, each feature used will be described in detail in this section. Features will be characterized in terms of qualitative type and output value and in particular they can be grouped in four different kinds:

- **Energy:** Energy-related features capture information about the energy distribution and evolution within the time evolution of the signal. These features can be deeply affected by how the song sounds like. For example, the different timbre of instruments playing in a song may influence the perception of the song genre based on characteristics like the loudness or the RMS value (Section 3.3.1 will contain details about loudness and RMS) due, for instance, to the usage of sound compressors. Moreover, genre will affect the time evolution of the energy values: jazz music can show rapid energy variations, while pop, rock and commercial songs tend to have a higher and steadier loudness curve.

- **Temporal:** Temporal features analyze aspects of music that are related to the distribution of audio events on time. These can be derived directly from the sampled signal, but also refer to spectral variations in order to detect temporal events, in which case they make particular use of the STFT algorithm described in 3.2.1. The most important feature in this group is tempo, measuring the frequency of the rhythmic accents in a music excerpt. Although this may seem relevant, it was chosen not to exploit them in the development of this work in order to focus on acoustic properties of lower level.

- **Spectral:** Timbrical aspects of music are defined based not only on the energy of the signal, but also on the distribution of this energy over the different frequencies. This distribution is called *spectrum* and it is computed on every frame of the audio signal in order to have a view that which is as istantaneous as possible. The accuracy on time of this representation allows to track the short-time variations in the frequency distribution of energy, denoting both note changes over time and instruments' characteristics. For instance, the spectral inharmonicity feature measures the divergence of the spectrum components from the multiples of the fundamental frequency, and provide information about how much a sound is harmonic. Chromatic features instead define the distribution of notes in each time slice of a song.

- **Waveform:** Some features need no pre-processing in shape of spectrum or energy computation in order to describe sound characteristics, but are extracted directly from the audio waveform. These are most useful to understand the rapidity of variations in a musical piece and

Type	Features
Energy	Energy dip probability, Intensity, Intensity Ratio, Loudness, RMS energy, RMS energy delta
Spectral	Chromagram, Crest, Irregularity J, Irregularity K, MFCC coefficients, Odd-even ratio, Rolloff, Sharpness, Spectral inharmonicity, Spectral centroid, Spectral contrast, Spectral flatness, Spectral flux, Spectral kurtosis, Spectral skewness, Spectral slope, Spectral smoothness, Spectral spread, Spectral standard deviation, Spectral variance, Tristimulus
Waveform	Average deviation, Kurtosis, Mean, NonZero Count, Skewness, Variance

Table 3.1: Summary of all the considered features, grouped by type

are related to statistical properties of the waveform itself in a given time range.

Table 3.1 contains a list of all the features we consider in this thesis, grouped by type. Those face different acoustic characteristics of sound, which proved to have an influence in previous works in MIR in describing acoustic properties like timbre or even high-level concepts like mood [36, 38, 39, 43, 44, 47]. Since music is known to present abrupt change of its properties during the time evolution, feature extraction is performed over short overlapping time frames, which allow to capture rapid variations of the music content. This *windowing* process is to be studied carefully in order not to introduce distortions of the signal properties, as it will be explained below. The actual data used as a feature will then be the mean of the values captured for all of the time windows.

3.2.1 Time-frequency analysis

At a very fundamental level music is made out of a mixture of sounds and noises. The difference among them lies in the fact that sound consists of periodic pressure fluctuations, while noises are irregular in time. People make sense of their auditory environment by identifying periodic sounds with specific frequencies and these sound events can start and stop at different moments in time. Therefore, representations of sound capture time and frequency components separately are commonly used as the first step in audio feature extraction.

Even if it may seem strange to divide specular aspects like time and frequency during the analysis, this is fundamental in order to focus on two different time-related aspects affecting the behaviour of music, which we can refer to as *short-time* and *long-time*. As staten above, in fact, sound is made out of periodic oscillations which can be represented as a continuous signal

over time. In order to represent the continuous process in a limited amount of resources, the signal is sampled at regular periodic intervals. The resulting sequence of samples still has continuous values, thus it is converted to a sequence of discretized samples through the process of quantization. This introduces two variables, the *sampling rate* F_S and the *dynamic range*, the former of which is important in defining the quality of a sound recording: the fundamental theorem by Nyquist and Shannon (1949) states that the sampling rate should be at least twice as big as the maximum oscillating frequency generating the signal in order to allow for perfect reconstruction. Considering that the human perceptual range spans in frequency between 20 and 22000 Hz, this means at first that the minimum sampling frequency should be greater than 44 kHz; then the time period for an oscillation lies between $4 \cdot 10^{-5}$ and $5 \cdot 10^{-2}$ seconds. The variations in the frequency content of a signal, corresponding to the variation of the perceived sound, happen instead within a time window which is way longer than the longest possible period. This makes the motivation for the mentioned procedure of *windowing*, consisting in splitting the time of the sampled signal into shorter windows, any of which defines an acoustic setting to be analysed in its frequency content. The output of this observation is called *spectrum* and it is to be further studied in its variations along different time windows, the interval between which has to be taken large enough in order to monitor the noticeable variations in sound.

The Short-Time Fourier Transform (STFT) is the most common time-frequency representation and has been widely used in many domains in addition to music processing. An important factor in the wide use of the STFT is the high speed with which it can be computed in certain cases when using the Fast Fourier Transform algorithm. The fundamental concept of Fourier transform is to represent the signal of interest as a linear combination of elementary signals forming a complete orthonormal basis of the signals' space. Those generators are actually sinusoids, representing simple, unique-frequency oscillations. The coefficients of this linear combination contains information about how the energy of the signal is distributed in frequency, generating a discrete spectrum as depicted in Figure 3.1.

The STFT is essentially a sequence of Discrete Fourier Transforms (DFTs) applied over subsequent audio segments overlapping in time. Here the term *discrete* defines the possibility of having only a finite amount of frequency bins to analyze, bounded by the length of the evaluated signal [52]. Actually, it is possible to calculate the DFT of an entire audio clip and show how the energy of the signal is distributed among different frequencies. However, such an analysis would provide no information about when these contributions start and stop in time, giving only a partial and static information. The idea behind the STFT is thus to process small segments of the audio clip at a time and the resulting sequence of spectra will contain information about time as well as frequency, as it can be seen in Figure 3.2.

At this stage, the mentioned process of windowing should be better analyzed, since it may damage the signal properties. In particular, it can be viewed as a convolution of the original audio signal with another signal (*time window*) that equals 1 during the time period of interest and 0 outside it. Such a signal in particular is called a rectangular window, but it is not the only possible type of window as long as it is not even the most effective: the signals which are targeted by Fourier analysis are periodical by nature, but, if the analyzed signal has been obtained by rectangular windowing, there will be a large discontinuity where the end of the signal is connected to the start of the signal in the process of periodic repetition. This discontinuity will introduce significant energy in all frequencies and distort the analysis. In order to reduce this effect, called *spectral leakage*, a non-negative smooth bell-shaped window can be used instead of a rectangular one: there are in literature several variants, with slightly different characteristics. The most famous of these are Hanning, Hamming, and Blackman. The convolution window is represented with $w(n)$ in the following formula, which expresses the whole STFT algorithm:

$$X(m,k) = \sum_{n=0}^{L-1} w(n)x(mN_h + n)e^{-2\pi j \frac{k}{L} n} \qquad k = 0, \ldots, L-1. \tag{3.1}$$

Here m represents the index of the current time frame, which addresses the spectrum in the temporal sequence, $\frac{k}{L}$ stands for the normalized frequency component (the actual frequency in Hz is $\frac{f_S}{2}\frac{k}{L}$), n is the sample within the window, L is the length of the window and finally N_h is the hopsize, meaning the number of non-overlapping samples between two consectutive time frames.

It is important to notice that the spectrum is complex, indeed usually it is splitted into its magnitude and phase parts also because of the relationship holding between the energy of a signal and the one of its spectrum. As long as this is just an introduction to the sound processing techniques used in scope of the present work, more details on this matter can be found in [55].

3.3 Feature description

In this section we provide the details about the features chosen for this work, similarly to what was done in [1]. Each paragraph starts with two tags defining the value type (scalar or vector) and the reference bibliography for the feature. The following naming conventions will be used: $x(n)$ is the n-th sample of the audio signal; N is the total number of samples of the windowed signal; a_k is the amplitude of the k-th frequency bin in the spectrum, while $f(k)$ is the frequency corresponding to that bin; K is the total number of bins in a spectrum.

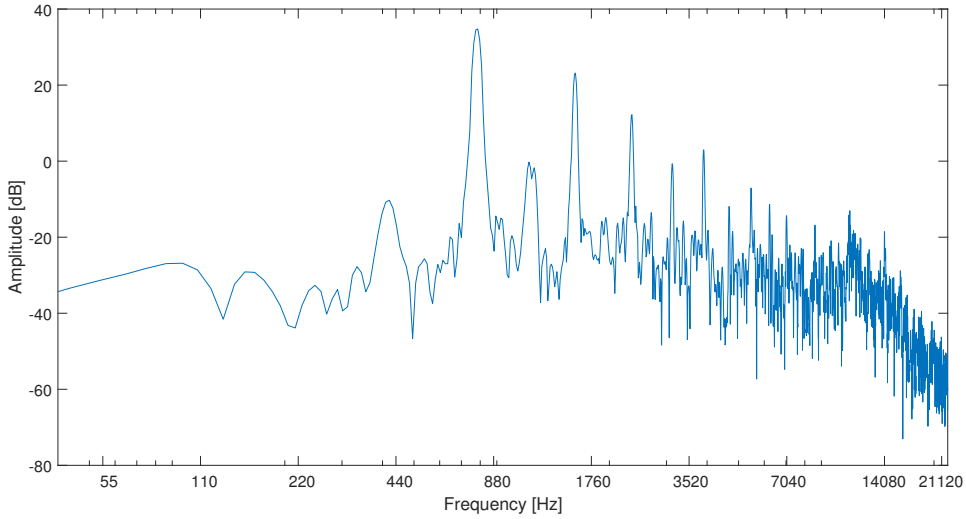

Figure 3.1: ***Discrete Time Fourier Transform*** *– The plot depicts a possible output for a Fourier transform of an audio segment (spectrum). The x-axis contains the reference frequency for the analysis and it appears logarithmically scaled in compliance to human hearing system, which also is. The y-axis instead defines the energy level in dB for each component. The peaks identify the frequency locations of the strongest components forming the analysed sound.*

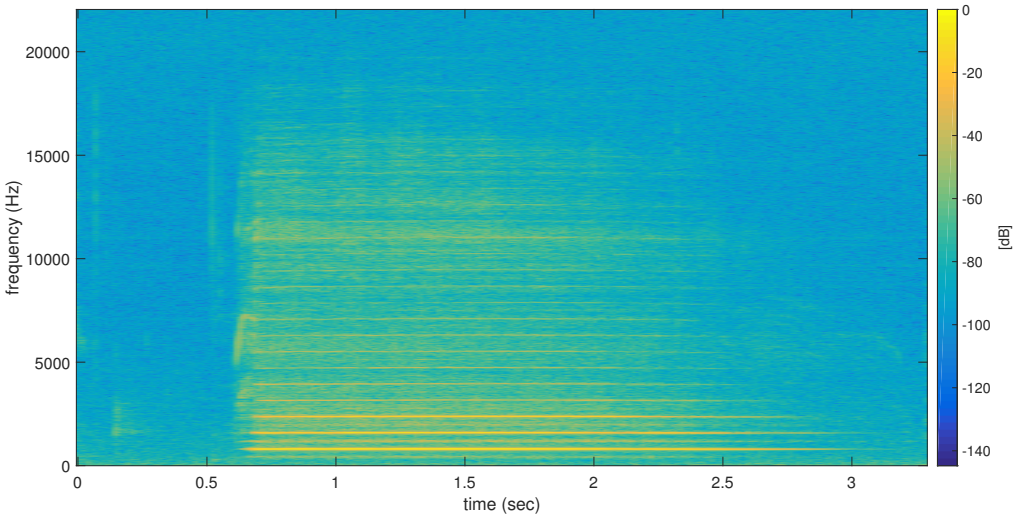

Figure 3.2: ***Short-Time Fourier Transform*** *– The plot consists in a juxtaposition of a sequence of DTFTs along time. The peaks and valleys in the above figure here become colours in the depicted scale, while the abscissa represent the timeline of the audio excerpt. It is possible to identify a first silence phase (blue) and the final fade out of the sound where the yellow lines slowly disappear. The analysed sound corresponds to a single flute note.*

3.3.1 Energy-related features

Energy dip probability

VALUE TYPE: Scalar

REFERENCES: [51, 53, 54]

Some high-level features are affected by the behaviour of a human voice over music, for instance this can be the key point for the identification of the genre *rap*, showing a great presence of speech-like vocals. It is very intuitive to try to discriminate speech and music based on shape of signal's energy envelope: speech signal has characteristic high and low amplitude parts, which represent voiced and unvoiced speech, respectively. On the other hand, the envelope of music signal is more steady. The energy contour is thus capable of separating speech from music, while considering energy minima below some threshold related to peak energy allowed to improve the recognisers' performances. In this context, the dip probability estimate consists in the ratio of frames that have dipped below the dip threshold within the averaging window, where the threshold is a product of a default thresold and the moving average of RMS spanning across sub-frames.

Intensity and Intensity Ratio

VALUE TYPE: Scalar

REFERENCES: [36]

First, the signal is divided into p sub-bands with the following frequency ranges:

$$\left(0, \frac{F_s}{2^p}\right), \left(\frac{F_s}{2^p}, \frac{F_s}{2^{p-1}}\right), \ldots, \left(\frac{F_s}{2^2}, \frac{F_s}{2}\right). \tag{3.2}$$

For each sub-band with a frequency range from L_p to H_p, the intensity ratio is the ratio of that sub-band's intensity to the overall intensity I:

$$Iratio_p = \frac{1}{I} \sum_{k=L_i}^{H_i} a_k. \tag{3.3}$$

The intensity I is computed by summing all the components:

$$I = \sum_{k=0}^{F_s/2} a_k. \tag{3.4}$$

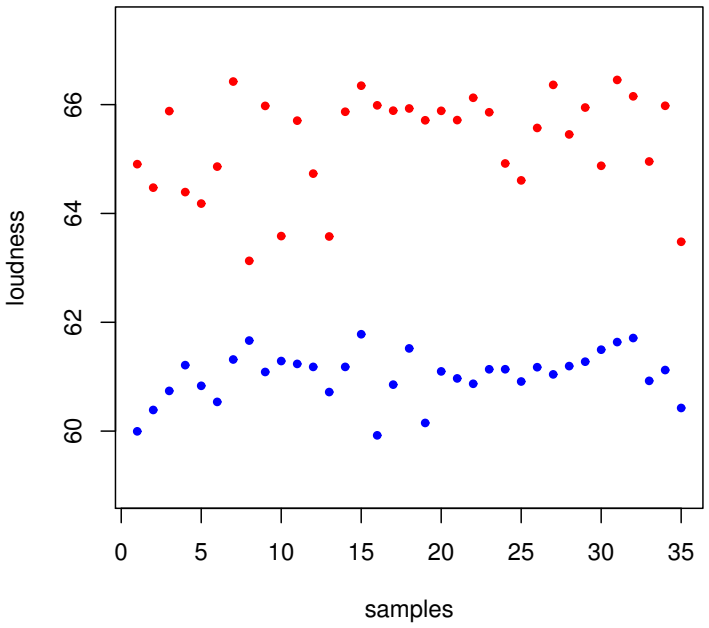

Figure 3.3: ***Loudness*** *– The plot depicts data taken be songs pertaining to different genres: it is immediately noticeable how a "loud" rock song (Queen - We Will Rock You) assumes higher values with respect to a quiet soft-country song (Cowboy Junkies - Postcard Blues)*

Loudness

VALUE TYPE: Scalar

REFERENCES: [37]

Computes the loudness on each frame. The loudness is the characteristic by means of which music can be ordered on a scale extending from quiet to loud. This feature is affected by parameters other than sound pressure, including frequency, bandwidth and duration. Its value is computed with an approach that takes into account these factors and is built on psychoacoustical theories that explain how the human ear perceive sounds. For instance, soft and slow pieces will have lower values than rock and metal highly compressed and distorted songs. In Figure 3.3 this is visualized by choosing a soft, slow country song with respect to a popular, energic rock anthem.

RMS energy

VALUE TYPE: Scalar

REFERENCES: [38]

RMS energy for each frame is computed as

$$RMS = \sqrt{\sum_{n=1}^{N} x(n)^2}, \tag{3.5}$$

where L is the length of the signal $x(n)$.

RMS energy delta

VALUE TYPE: Scalar

REFERENCES: [38]

RMS energy delta represents the difference between the RMS energy among successive time frames:

$$\begin{aligned} \Delta_{RMS}^{(m)} &= RMS^{(m+1)} - RMS^{(m)} \\ &= \sqrt{\sum_{n=1}^{N} \left(x(n)^{(m+1)}\right)^2} - \sqrt{\sum_{n=1}^{N} \left(x(n)^{(m)}\right)^2} \end{aligned} \tag{3.6}$$

3.3.2 Spectral-related features

Chromagram

VALUE TYPE: Vector

REFERENCES: [52]

The western music scale splits the octave in twelve equally spaced intervals (*pitch elements*), commonly known as notes. The chroma feature captures harmonic and melodic characteristics of music. They are robust to changes in timbre and instrumentation, in order to analyze the spectral content of an audio segment and classify the sound components into *pitch classes*. This is based on the fact that humans perceive two musical pitches as similar in color if they differ by an octave: a pitch can thus be separated into two components, which are referred to as *tone height* and *chroma*. The first refers to which octave the sound pertains to, seen as a range of frequencies, while the latter refers to the actual note which is played. Thus, it can assume values represented by the set $\{C, C\sharp, D, D\sharp, E, F, F\sharp, G, G\sharp, A, A\sharp, B\}$. A *pitch class* collects the whole set of pitches that share the same chroma.

Given an audio recording, the main idea of chroma features is to aggregate all the information that relates to a given chroma into a single coefficient for a given local time window. Shifting the time window results in a

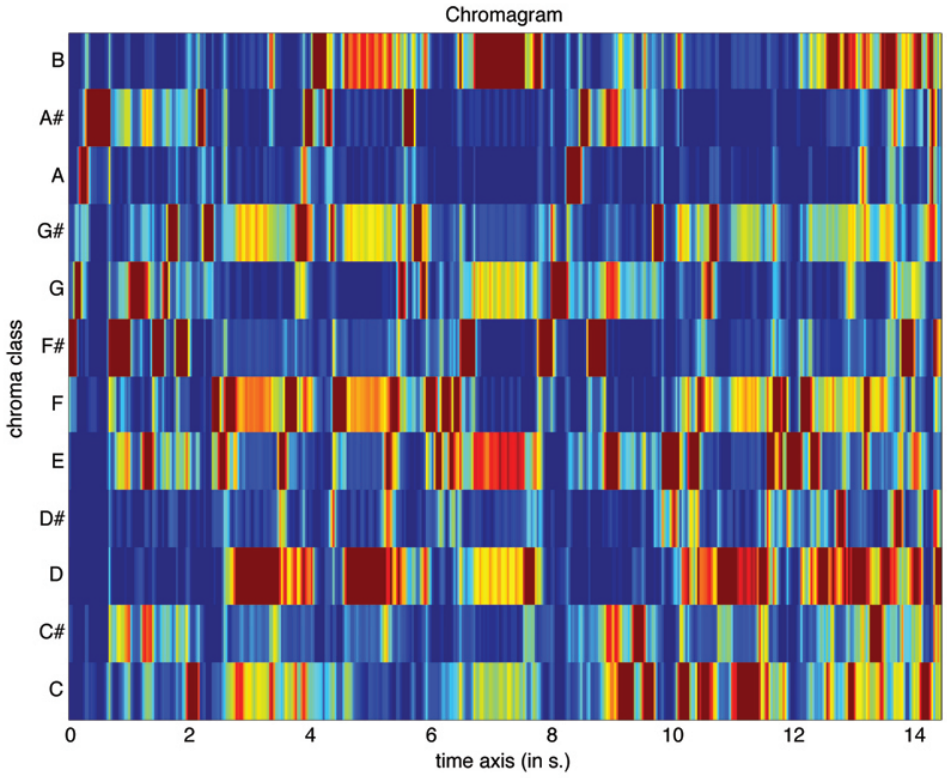

Figure 3.4: ***Chromagram*** *– The plot depicts a 14 seconds long chromagram of a jazz music excerpt. The abscissa contains the time axis, the y-axis instead contains the 12 pitch classes. The intensity of colours, scaled starting from blue up to red, defines how much the spectrum of the analyzed time frame shows peaks which frequency locations are compatible with the given pitch class.*

sequence of chroma features each expressing how the representation's pitch content within the time window is spread over the twelve chroma bands. The resulting time-chroma representation is also referred to as *chromagram*. Figure 3.4 shows a chomagram obtained from the an audio recording excerpt of jazz music excerpt.

Crest

VALUE TYPE: Scalar

REFERENCES: [40]

It is related to the flatness of the frame spectrum, i.e. to the noisiness/harmonicity of the related signal:

$$crest = \frac{\max a_k}{\frac{1}{K}\sum_k a_k} \tag{3.7}$$

where a_k is the amplitude of the k-th frequency bin and K is the number of bins.

Irregularity J

VALUE TYPE: Scalar

REFERENCES: [41]

This feature is related to the variation of the successive harmonic components of the spectrum. It is computed as the square of the difference in amplitude between adjacent partials:

$$irr_j = \frac{\sum_h (a_h - a_{h-1})^2}{\sum_h a_h^2} \tag{3.8}$$

where h is the number of the spectral components corresponding to the multiples of the fundamental frequency.

Irregularity K

VALUE TYPE: Scalar

REFERENCES: [42]

Measures the irregularity of the spectrum harmonics, which is empirically related to perceived timbral characteristics:

$$irr_k = \sum_k \left| a_h - \frac{a_{h-1} + a_h + a_{h+1}}{3} \right| \tag{3.9}$$

where h is the number of the spectral components corresponding to the multiples of the fundamental frequency. It differs from the previous in having the dimension of an amplitude.

MFCC coefficients

VALUE TYPE: Vector

REFERENCES: [43]

Human perception of the frequency content of sounds does not follow a linear scale. This fact has led to the idea of defining subjective pitch of pure tones; thus, for each tone with actual frequency f (measured in Hz), a subjective pitch is measured on a scale called Mel scale. As a reference point, the pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels. Other subjective pitch values are obtained by adjusting the frequency of a tone such that it is half or twice the perceived pitch of a

reference tone. The MFCC (Mel Frequency Cepstral Coefficients) feature extraction procedure is based on a Mel filter bank that shows triangular overlapping windows having center frequencies and bandwidths scaled by subjective measures. MFCCs are commonly derived as follows (see Figure 3.5 for a block diagram of the procedure):

1. Take the Fourier transform of each signal frame;
2. Filter the obtained spectrum, using the Mel filter bank;
3. Take the logs of the powers at each of the Mel frequencies;
4. Compute the discrete cosine transform (DCT) of the list of Mel log powers, as if it were a signal. The MFCCs are the amplitudes of the resulting spectrum.

Considering a logarithmic scale in power leads to a better approximation of the auditory system, allowing a better modeling of the human timbre perception. Figure 3.6 shows a comparisons between excerpts of two songs belonging to different genres, respectively a vocal performance and hard rock. Each dot represents an MFCC coefficient, computed on 35 different audio frames for the two tracks. As we can see, there is a substantial difference in the two plots. The vocal performance's MFCCs are on average strongly below zero, while the hard rock song shows values that are slightly higher. As we can see, just considering only the MFCC coefficient allows us to discern songs having strongly different timbres, such as those plotted in the figure.

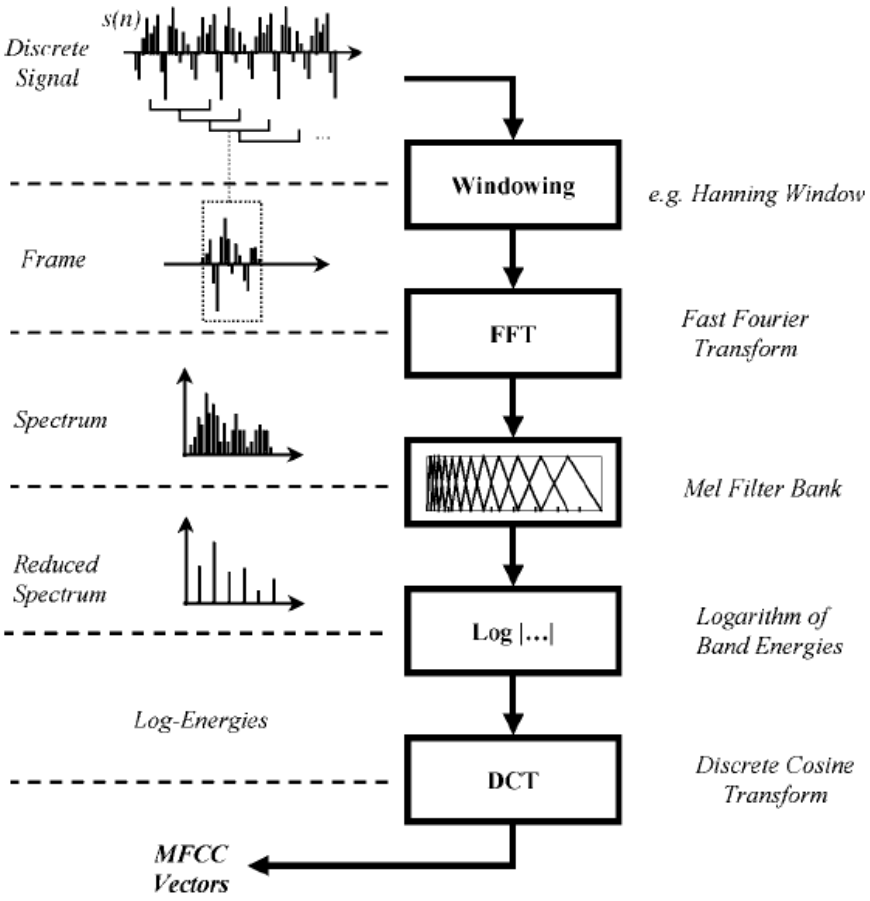

Figure 3.5: ***MFCC*** *– Illustration of the steps performed to compute Mel Frequency Cepstral Coefficients coefficients from a signal frame.*

Odd-even ratio

VALUE TYPE: Scalar

REFERENCES: [39]

It has been shown that some instruments have discernible lacks of energy in even or odd spectral harmonics (components corresponding to frequencies that are multiple of the fundamental frequency). The odd-even ratio is defined as the ratio between odd and even harmonics:

$$\begin{aligned} odd &= \frac{\sum_h a_{2h-1}}{\sum_h a_h}, \\ even &= \frac{\sum_h a_{2h}}{\sum_h a_h}, \end{aligned} \tag{3.10}$$

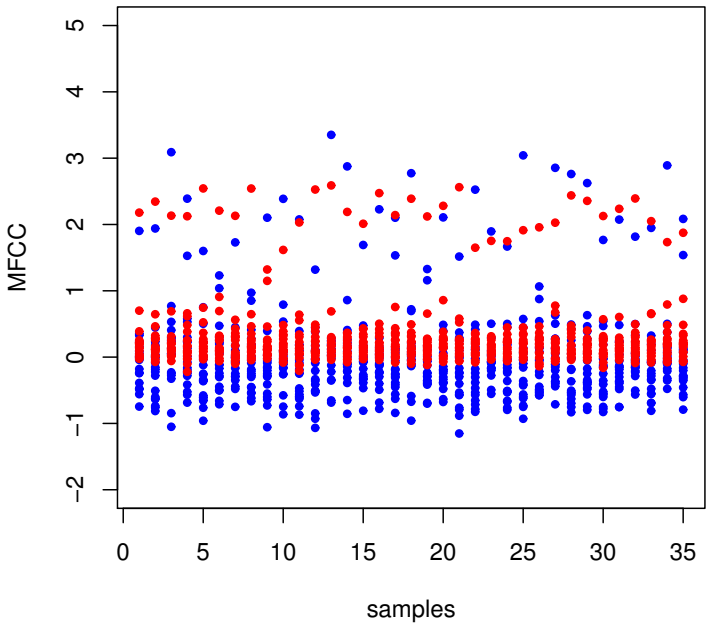

Figure 3.6: ***MFCCs comparison*** *– A plot of the MFCC coefficients extracted from excerpts of a hard rock song (Led Zeppelin - The Immigrant Song) and a vocal performance (Drevo - Our Watcher, Show Us The Way).*

where h is the number of the spectral components corresponding to the multiples of the fundamental frequency. We are not dealing with single instruments here, but this feature can be a useful indicator for spectrum regularity.

Spectral rolloff

VALUE TYPE: Scalar

REFERENCES: [44]

This is the minimum frequency value $f_{K_{roll}}$ such that a given percentage R (usually 95%) of the spectrum energy stays below that frequency:

$$rolloff = \min \left\{ f_{K_{roll}} \left| \sum_{k=1}^{K_{roll}} a_k \geq R \sum_{k=1}^{K} a_k \right. \right\}, \tag{3.11}$$

where K_{roll} is the spectral component corresponding to $f_{K_{roll}}$.
It is a measure of the brightness of the sound: the higher the rolloff, the more

high-frequency components are present in the spectrum, denoting a brighter sound.

Sharpness

VALUE TYPE: Scalar

REFERENCES: [45, 49]

Sharpness is the perceptual equivalent of the spectral centroid and it is based on psycho-acoustical models. It follows from the definition of the Bark psychoacoustical scale [49], which divides the space of frequencies into perceptually equal bands in term of distance. It approximates to a logarithmic scale above 500 Hz, while tends to be linear below that threshold.

$$sharpness = 0.11 \cdot \frac{\sum_{z=1}^{Z} z \cdot g(z) \cdot L'_S(z)}{L_T}, \tag{3.12}$$

where z indicate a Bark band, $L'_S(z)$ is the specific loudness (exhibits the loudness across specific Bark bands), L_T is the total loudness (sum of all the specific loudness values), Z is the number of Bark bands and

$$g(z) = \begin{cases} 1 & \text{if } z < 15 \\ 0.066 \cdot \exp(0.171 \cdot z) & \text{if } z \geq 15 \end{cases}.$$

Spectral inharmonicity

VALUE TYPE: Scalar

REFERENCES: [39]

Measures the divergence of the spectrum components from the multiples of the detected fundamental frequency f_0:

$$inharmonicity = \frac{2}{f_0} \frac{\sum_k |f_k - k f_0| a_k^2}{\sum_k a_k^2}, \tag{3.13}$$

where f_k, a_k and k are as usual. The obtained value is an indicator of how much a sound is harmonic. For instance, the spectrum of a purely harmonic sound would determine an inharmonicity value equal to 0, as the multiples of the fundamental frequency would be null.

Spectral centroid

VALUE TYPE: Scalar

REFERENCES: [39]

Spectral centroid basically represents the center of mass of the spectrum:

$$centroid = \frac{\sum_k a_k \cdot f_k}{\sum_k a_k}. \tag{3.14}$$

It states if the spectrum is mostly composed by low or high frequency components, which in turn it is often related to the perceived brightness of the sound.

Spectral contrast (mean, peak, valleys)

VALUE TYPE: Vector

REFERENCES: [46]

Usually, in a spectrum obtained from harmonic sounds, the strong spectral peaks roughly correspond to harmonic components, while non-harmonic components often appear at spectral valleys. The Spectral Contrast is a measure that reflects the respective distribution of the harmonic and non-harmonic components. First, the track is segmented into overlapping frames; then the spectrum is computed and filtered by an octave-scale filter that divides it into seven sub-bands.

Let us consider the p-th sub-band. We sort it in descending order, such that $a_1^{(p)} > a_2(p) > \cdots > a_k^{(p)}$. In order to increase the steadiness of the features, the peak and valley strengths are found by taking a proportion (defined as α) of FFT bins respectively from the top and bottom of the sorted bins and finding the mean of those.

$$peak_p = \log \frac{1}{\alpha K} \sum_{k=1}^{\alpha K} a_k^{(p)}, \qquad valley_p = \log \frac{1}{\alpha K} \sum_{k=1}^{\alpha K} a_{K-k+1}^{(p)}. \tag{3.15}$$

α is a regularization parameter: here, we keep it equal to the value fixed by the author in the original paper, i.e. 0.02.
Finally, the spectral contrast is defined as:

$$contrast = peak_p - valley_p \tag{3.16}$$

for each sub-band.
In our dataset we do not include it directly, but we consider only separated peak and valley strengths. The means of all the spectral components in each sub-band are also calculated and included in the dataset.

Spectral flatness

VALUE TYPE: Scalar

REFERENCES: [50]

Spectral flatness is a measure used to characterize a spectrum as noisy-like or tone-like. The perceptual difference holds in the fact that the latter kind sounds pitched, so that an actual note could be heard. The usage of this feature may thus resemble the one of spectral variance, but in this context the meaning of tone-like considers much more the amount of resonant structures in the spectrum. A high spectral flatness (approaching 1 for white noise) indicates that the spectrum has a similar amount of power in all spectral bands, sounding similarly to white noise, and the graph of the spectrum would appear relatively flat and smooth. A low spectral flatness (approaching 0 for a pure tone) indicates that the spectral power is concentrated in a relatively small number of bands. This would typically sound like a mixture of sine waves, while the spectrum would appear spiky and regular.

The spectral flatness is calculated by dividing the geometric mean of the power spectrum by its arithmetic mean:

$$\mathit{flatness} = \frac{\sqrt[K]{\prod_{k=0}^{K-1} a_k}}{\frac{\sum_{k=0}^{K-1} a_k}{K}} = \frac{\exp\left(\frac{1}{K}\sum_{k=0}^{K-1} \ln a_k\right)}{\frac{1}{K}\sum_{k=0}^{K-1} a_k}, \tag{3.17}$$

where a_k represents the magnitude of the k-th spectral frequncy bin. Note that a single (or more) empty bin yields a flatness of 0, so this measure is most useful when bins are generally not empty. The ratio produced by this calculation is often converted to a dB scale for reporting.

Spectral flux

Value Type: Scalar

References: [1]

Spectral flux is a measure of the change in energy between various frequency bands in a sequence of spectra measured from the audio data. Spectral flux is calculated in three steps:

$$\mathit{spflux}_m = \left\| a_k^{(m+1)} - a_k^{(m)} \right\|_2 = \sqrt{\sum_k \left| a_k^{(m+1)} - a_k^{(m)} \right|^2}, \tag{3.18}$$

where m stands for the m-th spectrum of the sequence. Usually, one considers only the positive values in the spectral difference:

$$\mathit{spflux}_p = \left\| H^+\left(a_k^{(p+1)} - a_k^{(p)}\right) \right\|_2, \tag{3.19}$$

where $H^+(x) = \frac{x+|x|}{2}$ is the positive half-wave rectifying function which sets negative values to zero and leaves positive values unaltered.

Spectral kurtosis

VALUE TYPE: Scalar

REFERENCES: [39]

Kurtosis is a measure to determine the flatness of a probability distribution around its mean value; when applied to a spectrum becomes an indicator of the noisiness of the signal:

$$kurtosis = \frac{\frac{1}{K}\sum_k (a_k - \bar{a})^4}{\sigma^4}, \tag{3.20}$$

where σ is the standard deviation of the spectral components.

A high-kurtosis spectrum has a sharper peak and fatter tails, while a low-kurtosis spectrum has a more rounded peak and thinner tails. Thus, we expect a high kurtosis from songs where high and low frequencies succeed one another without mixing together. On the contrary, in common songs frequencies are usually mixed together and are quite balanced up to a certain frequency range, thus the kurtosis will be smaller.

Spectral skewness

VALUE TYPE: Scalar

REFERENCES: [39]

Skewness determines the asymmetry of the spectrum around its mean value:

$$skewness = \frac{\frac{1}{K}\sum_k (a_k - \bar{a})^3}{\sigma^3}. \tag{3.21}$$

A positive value means that the spectrum is skewed towards the right, thus showing a long tail on lower frequency components; with negative values, the spectrum is skewed towards the left; for perfect symmetry, skewness has to be 0.

Spectral slope

VALUE TYPE: Scalar

REFERENCES: [39]

Spectrum components usually tend to decrease towards higher frequencies. The spectral slope gives the rate of descent of the spectrum, obtained by computing the linear regression of the spectral amplitude:

$$slope = \frac{1}{\sum_k a_k} \frac{K\sum_k f_k \cdot a_k - \sum_k f_k \cdot \sum_k a_k}{K\sum_k f_k^2 - \left(\sum_k f_k\right)^2}, \tag{3.22}$$

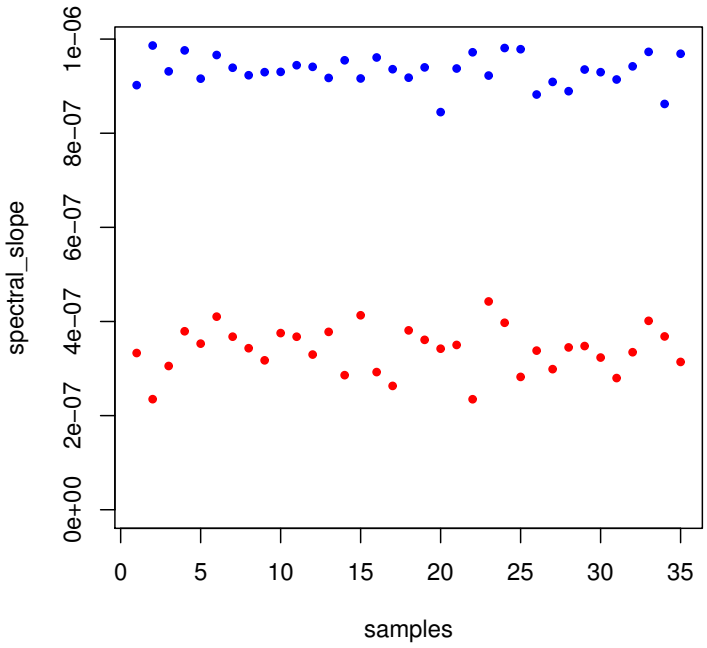

Figure 3.7: ***Spectral Slope*** *– The two plots show the spectral slope values computed on audio frames of an ethnic instrumental song (Yakshi - Chandra) and of a modern pop song (Bobby Brown - My Prerogative).*

where f_k is the frequency corresponding to the k-th spectral component a_k.

Considering the definition of spectral slope, we expect higher values for dark and gloomy songs, and, contrarily, lower values for bright and clear songs.
Figure 3.7 contains a plot of spectral slope values computed on the frames of two songs. The first track, an ethnic instrumental song, clearly shows higher values with respect to the second pop song.

Spectral smoothness

VALUE TYPE: Scalar

REFERENCES: [47]

Related to the differences between adjacent spectral components. It has been empirically found that single instruments can be discerned by means of the spectral smoothness value. For example, a resonant sound like sitar's one shows a fairly smooth spectrum, while a human voice has a more irregular one. This is verified in Figure 3.8, where the data from an indian folk

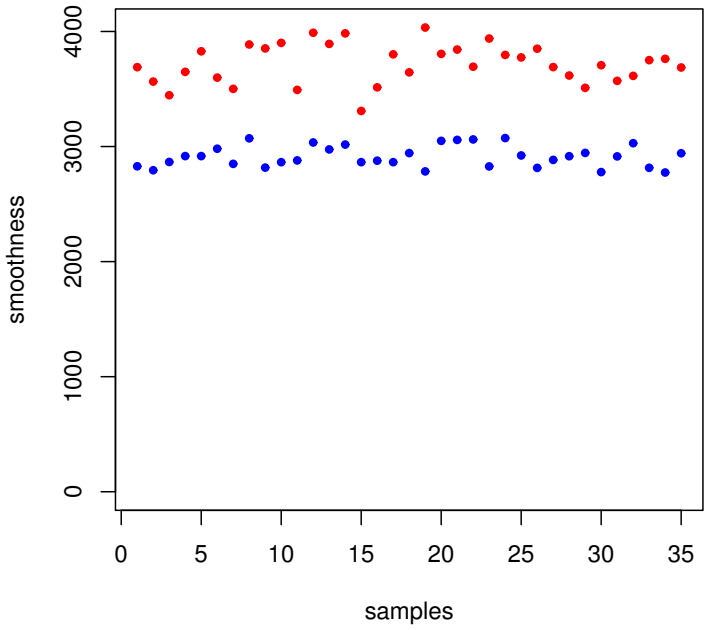

Figure 3.8: ***Spectral Smoothness*** *– Spectral smoothness values extracted from the frames of a strongly resonant song (Kourosh Zolani - Peaceful Planet) and a rap song (Eminem - My Fault).*

song show greater smoothness values with respect to a rap one. Spectral smoothness is calculated by evaluating the log of a component minus the average of the log of the surrounding components:

$$smoothness = 20 \sum_{k} \left| \log(a_k) - \frac{\log(a_{k-1}) + \log(a_k) + \log(a_{k+1})}{3} \right|. \quad (3.23)$$

Spectral spread

VALUE TYPE: Scalar

REFERENCES: [50]

The spectral spread describes the average squared deviation of the spectrum around its centroid, which is commonly associated with the bandwidth of the signal. Noise-like signals have usually a large spectral spread, while individual tonal sounds with isolated peaks will result in a low spectral spread. Similar to the centroid, the spectral spread is normalised by the sum of the

frequency amplitudes, such that the feature value ranges between zero and one:

$$spread = \frac{\sum_k (f_k - centroid)^2 \cdot a_k}{\sum_k a_k}. \tag{3.24}$$

Spectral standard deviation

VALUE TYPE: Scalar

REFERENCES: [39]

Captures the standard deviation of the spectral amplitudes:

$$stdev = \sqrt{\frac{1}{K}\sum_k (a_k - \bar{a})^2}. \tag{3.25}$$

It is an index of the distribution of the spectrum energy, thus of the noisiness of the sound. A small spectral standard deviation means that all of the spectrum energy is concentrated around the same frequency, thus the produced sound cannot be considered noise. Contrarily, a flat and distributed spectrum is typical of noisy sounds.

Spectral variance

VALUE TYPE: Scalar

REFERENCES: [39]

Captures the variance of the spectral amplitudes:

$$variance = \frac{1}{K}\sum_k (a_k - \bar{a})^2. \tag{3.26}$$

The same considerations holding for spectral standard deviation are of course true also for spectral variance, the only difference among which is the fact that the latter in the squared value of the first, thus measuring the same phaenomenon with a different scaling.

Tristimulus (1,2,3)

VALUE TYPE: Vector

REFERENCES: [48]

The three tristimulus values were introduced as acoustic equivalent to the color attributes of the RGB model. These values are defined as energy ratios, and respectively account for the strength of the fundamental, the mid-range, and high-frequency harmonic content.

$$\begin{aligned} tristimulus_1 &= \frac{a_1}{\sum_h a_h}, \\ tristimulus_2 &= \frac{a_2 + a_3 + a_4}{\sum_h a_h}, \\ tristimulus_3 &= \frac{\sum_{h=5}^{H} a_h}{\sum_h a_h}. \end{aligned} \tag{3.27}$$

Here h represents the number of the spectral components corresponding to the multiples of the fundamental frequency. Notice as well that the sum of the three values always equals 1.

3.3.3 Waveform-related features

Mean

VALUE TYPE: Scalar

REFERENCES: [50]

It corresponds to the mean value of the waveform samples in the frame and it is fundamental for the computation of the other Waveform features:

$$\bar{x} = \frac{\sum_{n=1}^{N} x(n)}{N}. \tag{3.28}$$

Average deviation

VALUE TYPE: Scalar

REFERENCES: [39]

Computes the average deviation of the frame signal, i.e. the mean of the absolute deviations of each sample from the samples mean:

$$avgdev = \frac{\sum_{n=1}^{N} |x(n) - \bar{x}|}{N}, \tag{3.29}$$

where N is the number of samples in the frame, $x(n)$ is the n-th sample and $\bar{x}$ as above.

Kurtosis

VALUE TYPE: Scalar

REFERENCES: [39]

It has the same definition of spectral kurtosis in Section 3.3.2, but it is applied to the samples of each frame.

Skewness

VALUE TYPE: Scalar

REFERENCES: [39]

It has the same definition of spectral skewness in Section 3.3.2, but it is applied to the samples of each frame.

Variance

VALUE TYPE: Scalar

REFERENCES: [39]

It has the same definition of spectral variance in Section 3.3.2, but it is applied to the samples of each frame.

3.4 Statistical tools

This section will be devoted to the description of the mathematical and statistical foundations of this work. It starts with a brief description about data mining, explaining its role of primary importance in being the field of research which collects the statistical experience and transforms it in knowledge acquisition mechanisms. The second part of this section is devoted to the concept of distance, according to what was stated in Chapter 2, and it includes formal definitions as well as a collection of technical approaches and tools descriptions. The last part of the present is devoted to delineate some techniques that are used for the management of data, with particular care given to the concepts coming from data exploration, data standardization and prediction.

3.4.1 Data mining and machine learning: investigating data

Data mining is a recent subfield of informatics which goal is to transform raw data into understandable structures. It involves pre-processing steps as well as modelization and data visualization. Hence the contribution of statistics and machine learning techniques is strong, since this fields of research provide the knowledge and the applied techniques. The core of data mining is made of concepts like *inference*, *classification* and *model learning*. In particular, this work will use the first to extend the knowledge collected from a sample of users and songs to the whole population; in Chapter 4 we will see that classification will be adopted for integrating the respective inferred characteristics. Finally, model learning will be used in order to define high-level structures where to incapsulate the information and to correlate it throughout the populations.

The meanings of *understandable structures* of data, towards the above ideas, are:

- **Data visualization** is an effective approach for displaying data information in graphic, tabular or other visual formats. The goal of visualization is to allow data to be immediately recognisable in their visual patterns. Thus, they will easily be interpretated in order to synthesize the information they contain both as inner correlations or as evolution tendencies. Successful visualization requires the data to be formatted in such a way that those relationships can be analyzed and reported.

- **Association mining** is the task of discovering interesting relations between variables in large databases: it is intended to identify strong regularity rules using some measures of interestingness. It differs from sequence mining because of the lack of an order in data presentation.

- **Sequence mining** is the task of recognising patterns that are visible in a series of data, if those are presented as a ordered sequence (*time series*). The patterns consist in subsequences of the original data showing the same characteristic behaviour, keeping the same samples order in any of the pattern instances.

- **Inference** is the process by which a population is characterized in its properties by induction out of a partial amount of data. Usually, an observation allows to provide an hypothesis on the distribution of data: inference is accomplished by either the explicitation or the refusal of a cause-effect relationship in the shape of a statistical test on that hypothesis. It basically deals with the comparison of the statistical distributions of data. It is worth to mention also that inference allows two different approaches which divide the whole statistic theory, *frequentist* and *bayesian*. While the former works only on data in order to make its considerations, the latter permits the formulation of an *a-priori* conjecture on the distribution to be verified after data realizations are known.

- **Clustering and classification** techniques both consist in rationally grouping objects. The former start from the data to derive the best way of aggregating or disgregating clusters of objects, on the basis of some optimality criteria, possibly without information on the number of classes. This encompasses an implicit similarity rule, since the points belonging to the same class are defined as similar, while the points belonging to different classes will be dissimilar.
  Classification techniques identify to which of a set of sub-populations (called *categories*) a new observation belongs, on the basis of a training set of data containing observations which category membership is

known. The focus is on finding the best relationship between an attribute set and the class label of the input data. The model generated by a learning algorithm should then both fit the input data and predict the class labels of records it has never seen. Therefore, a key objective for classification is to build models with good generalization capability.

- **Model learning** consists in deriving an hypothesis on the model which originated the data. This entails considering a given *dependent* variable, possibly the belonging class in a classification problem, and explaining it as a function of the other *independent* variables: e.g., the features collected plus some kind of noise, which is always present when measuring and modeling. The most trivial example of model to be learned consists in linear regression, but general models may present nonlinear structures, as we will see later.

- **Similarity search** is a task which objective is to capture the similarity of complex domain-specific objects: data are encapsulated in a multi-dimensional vector space as features of objects but the distance function in this feature space is not known. That is to be obtained by adapting some usual metric through data transofrmation according to a known similarity concept which behaviour to be emulated: the similarity search is naturally translated into a neighborhood query in the feature space.

The machine is thus required to *learn* properties from the data in a way that resembles the human attitude: it builds a knowledge by grasping informations from the world and deriving a general rule, which can possibly be used later for prediction purposes on the behaviour of new data.

3.4.2 The concept of distance

As stated in Chapter 2, *distance* between objects, along with its opposite, *similarity*, is one of the most important concepts in recommending. Nevertheless, its description is a difficult task: a real world distance can in fact be o7fickle and unclear, far from the usual euclidean way of describing equidistant places from an origin point as a circle centered in it. Here will be shown some of the theories and applications which had been basis for the investigations of the present work.

Distances in mathematics

A *distance* is a numerical description of how far apart objects are. In mathematics, a distance function or metric is a generalization of the concept of physical distance: a metric is a function that behaves according to a specific set of rules and a way of describing what it means for elements of some space to be "close to" or "far away from" each other.

Definition 3.1. Let X be a nonempty set. A function $\rho : X \times X \to \mathbb{R}$ is called a *metric* provided, for all $x, y, z \in X$,

1. $\rho(x, y) \geq 0$ (non-negativity);
2. $\rho(x, y) = 0 \iff x = y$;
3. $\rho(x, y) = \rho(y, x)$ (simmetry);
4. $\rho(x, y) \leq \rho(x, z) + \rho(z, y)$ (triangle inequality).

A pair (X, ρ) is called a *metric space*. If X is a linear vector space, a nonnegative real-valued function $\|\cdot\| : X \to \mathbb{R}$ is called a *norm* provided, for each $u, v \in X, \alpha \in \mathbb{R}$,

1. $\|u\| = 0 \iff u = 0$;
2. $\|\alpha u\| = \alpha \|u\|$;
3. $\|u + v\| \leq \|u\| + \|v\|$.

The pair $(X, \|\cdot\|)$ is called a *normed linear space*, and any norm on a linear space induces a metric ρ on the same space by defining

$$\rho(x, y) = \|x - y\| \ \ \forall x, y \in X.$$

In the previous definition, some condition may be relaxed in order to obtain spaces which are characterized in a different way. An useful example is the *pseudometric space*, which holds when allowing the possibility that $\rho(x, y) = 0$ even if $x \neq y$. On such a space, it is possible to define an equivalence relation, namely $x \cong y$ provided $\rho(x, y) = 0$, and the set X can be partitioned into a collection of disjoint equivalence classes $X/\cong$. It comes apparent how the pseudometric ρ defines a metric $\hat{\rho}$ on this quotient set. This solution finds application when dealing with different points in a space owing equivalent properties, as in the case of silent song slices belonging to songs owing different properties.

Once given a metric space (X, ρ), it is possible to define the concept of *open ball* in order to characterize better the metric in a geometric way:

Definition 3.2. Let (X, ρ) be a metric space. Given a point $x \in X$ and $r > 0$, the set

$$\mathcal{B}(x, r) \coloneqq \left\{ x' \in X \middle| \rho(x', x) < r \right\} \tag{3.30}$$

is called the *open ball* of radius r, centered in x. A *neighborhood* of x is a subset of X that contains at least an open ball of some radius r, centered in x.

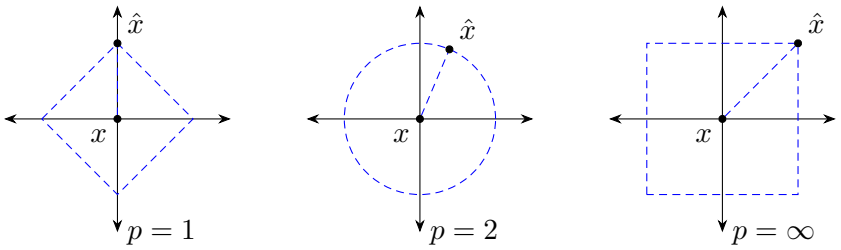

*Figure 3.9: ℓ^p-**norms** – Showing the different shapes of unitary balls in $\mathbb{R}^2$ (with $x = (0, 0)$, $r = 1$). This extends naturally for a higher number of dimensions.*

The definition 3.2 makes clear the notion of closeness, since it explains how a point becomes a neighbor for another. It is important to notice that, anytime a radius is fixed, the open ball becomes a defined subset of X, thus it acquires a shape in its representation. This shape does not depend strictly on the points in X, but rather on the metric ρ which is the only responsible of what points are close or not to each other inside X.

Example 1. The most common example of normed linear spaces is the *Euclidean* space, defined by the pair $(\mathbb{R}^n, \|\cdot\|_2)$, where $\|x\|_2 \coloneqq \sqrt{\sum_{i=1}^n x_i^2}$ for all $x \in \mathbb{R}^n$. This space owes a corresponding metric space, within which a unitary radius open ball owes the shape of a n-dimensional sphere, as it appears from the formula defining the norm. Even if it is easy to generalize this result in an algebraic way, this is not true in a geometrical perspective. The most natural extension of the Euclidean metric space consist in the so-called *ℓ^p-norms*, which are generated by substitution of the index 2 with a generic $p \in [1, \infty)$:

$$\begin{aligned} \ell^1(x, y) &= \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i| \,; \\ \ell^p(x, y) &= \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \qquad \text{for } p \in (1, \infty); \\ \ell^\infty(x, y) &= \max_{i=1,\dots,n} |x_i - y_i|. \end{aligned} \tag{3.31}$$

The consequence of the chosen metric on the shape of the unitary open balls is clear, as it can be seen in Figure 3.9. However, it is important to notice that the one showed is a simple extension of the notion of Euclidean distance which does not affect relevant properties like, for instance, the invariance on translations over the space X. Although other concepts and approaches had already been studied, it is easy to figure out that, without considering nonlinear behaviours, the building of a metric in order to shape real distance concepts may lead to models which are too simplistic even if easier in construction and computation.

Definition 3.3. Suppose X is a linear space, let $\|\cdot\|$ and $\|\cdot\|_*$ be two different norms on X. They are said to be *equivalent* if there exist $0 \leq m \leq M \in \mathbb{R}$ such that

$$m \|x\|_* \leq \|x\| \leq M \|x\|_* \quad \forall x \in X.$$

Proposition 1. *Any norm on $\mathbb{R}^N$ is equivalent.*

Proposition 2. *Any normed linear space X with finite dimension equal to N is isomorphic to $\mathbb{R}^N$.*

Proof. It is sufficient to exhibit an isomorphism $I : X \to \mathbb{R}^N$: consider a basis of X given by the set of vectors $\{x_1, \ldots, x_N\}$. Then it is sufficient to define the mapping $I : x_i \mapsto e_i$ where e_i is the i-th basis vector of $\mathbb{R}^N$. □

When working with data, the usual approach considers sampled observations of continuous measurements as points in a vector space, where each variable corresponds to a dimension taking values in $\mathbb{R}$. This leads to the conclusion that any distance defined in the variable space is equivalent, according to the previous Propositions. The usual theory which is applied is thus the one of numeric Hilbert spaces, because it naturally generalizes the notion of Euclidean space:

Definition 3.4. A Hilbert space $\mathcal{H}$ is a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product. To say that $\mathcal{H}$ is a complex inner product space means that $\mathcal{H}$ is a complex vector space with an inner product $\langle x, y \rangle$ associating a complex number to each pair of elements $x, y \in H$ that satisfies the following properties:

1. The inner product of a pair of elements is equal to the complex conjugate of the inner product of the swapped elements: $\langle y, x \rangle = \overline{\langle x, y \rangle}$;

2. The inner product is linear in its first argument: $\langle ax_1 + bx_2, y \rangle = a\langle x_1, y \rangle + b\langle x_2, y \rangle \ \forall a, b \in \mathbb{C}$;

3. The inner product of an element with itself is positive definite: $\langle x, x \rangle \geq 0$, where the case of equality holds precisely when $x = 0$.

A real inner product space is defined in the same way, except that $\mathcal{H}$ is a real vector space and the inner product takes real values. Such an inner product will be bilinear, meaning linear in each argument. It can be noticed that the inner product of any element with itself, namely $\langle x, x \rangle$ for $x \in H$, defines a norm, and this justifies the first asserption about $\mathcal{H}$ being a metric space.

In the following part, H will be a Hilbert space, V a subset of H.

Definition 3.5. The *orthogonal complement* of V is defined as

$$V^{\perp} \coloneqq \{x \in H | \langle x, v\rangle = 0,\ \forall v \in V\}$$

and it is a subspace of H.

Proposition 3. *Let V be a closed subspace of H, $x \in H$. Then it exists a unique $v^* \in V$ such that*

$$dist(x, V) = \inf_{v \in V} \|x - v\| = \|x - v^*\|.$$

Proposition 4. *Under the same conditions above,*

$$H = V \oplus V^{\perp} \coloneqq \{v + w | v \in V, w \in W\},$$

where the $\oplus$ operator is called orthogonal sum.

Proof. Let $x \in H$. From the previous Proposition, it exists a unique $v \in V$ such that $\text{dist}(x, V) = \|x - v\|$. Define then $w = x - v$ and consider $u \in V$ and $\lambda \neq 0$ such that $\lambda\langle w, u\rangle \geq 0$. It results

$$\|w\|^2 \leq \|x - v - \lambda u\|^2 = \|w - \lambda u\|^2 = \|w\|^2 + |\lambda|^2 \|u\|^2 - 2\lambda\langle w, u\rangle.$$

When dividing everything by $|\lambda|$ and then applying the limit as $|\lambda| \to 0$,

$$|\langle w, u\rangle| \leq \frac{|\lambda|}{2} \|u\|^2 \implies \langle w, u\rangle = 0. \tag{3.32}$$

This proves that $w \in V^{\perp}$ because of the generality in the choice of $u \in V$. □

Definition 3.6. The vector v in the previous proof is called the *orthogonal projection* of x onto V and the linear map $P_V : H \to V,\ x \mapsto v$ is called *projection map*. Its output is the point in V which realizes the minimum distance of x from V. In particular, the projection map equals the identity map if restricted to the subspace V and, if the codomain of P_V is extended to H itself, this means the map to be trivially idempotent, since $P_V^2 = P_V(P_V(x)) = P_V(v) = v$. The *range* of P_V equals to V itself; it corresponds to the *kernel* of its complementary map $P_{V^{\perp}} = I_H - P_V$ and vice versa.

For every $x, y \in H$, this means that $\langle Px, (y - Py)\rangle = \langle (x - Px), Py\rangle = 0$. Equivalently: $\langle x, Py\rangle = \langle Px, Py\rangle = \langle Px, y\rangle$.

Proposition 5. *Given $\{v_1, \ldots, v_K\}$ an orthonormal basis of the subspace V and let A denote the n-by-k matrix which columns are $\{v_1, \ldots, v_K\}$. The projection matrix P is given by*

$$P = AA^T = \sum_{i=1}^{K} \langle v_i, \cdot\rangle v_i. \tag{3.33}$$

The matrix A^T is the partial isometry that vanishes on the orthogonal complement of V and A is the isometry that embeds V into the underlying vector space.

Proof. Consider an arbitrary $x \in H$ as the sum of the two orthogonal components $x_{\parallel} = P_V(x)$ and $x_{\perp} = (I - P_V)(x)$. Applying the defined matrix transformation,

$$Px = P(x_{\parallel} + x_{\perp}) = AA^T x_{\parallel} + AA^T x_{\perp} = x_{\parallel} + 0 = x_{\parallel} = P_V(x).$$

□

Proposition 6. *Given $U \neq V$ subspaces of H, $A = \{a_1, \ldots, a_{N_U}\}$, $B = \{b_1, \ldots, b_{N_V}\}$ the matrices defining an orthonormal basis for U, V respectively, the projection operator $P_{U,V} : U \to V$ is represented by the matrix $P_{UV} = BB^T A$.*

Proof. Consider $u \in U$; $\hat{u} = (A^T A)^{-1} A^T u$ is the vector representing u within the coordinates system defined by A. Then, according to the previous notation, $P_V(u) = Pu = PA\hat{u}$. □

Since the geometric properties of this map can be studied by looking to either the singular values of the matrix P_{UV} or the eigenvalues of the symmetric version $P_{UV}^T P_{UV}$, it is important as well to state some results about the related estimates. The basic notion about this is the Gershgorin theorem:

Theorem 1. *Let $A = (a_{ij}) \in \mathbb{C}^{n \times n}$. Then each eigenvalue of A lies in one of the disks in the complex plane*

$$D_i \coloneqq \left\{ \lambda \;\middle|\; |\lambda - a_{ii}| \leq r_i \coloneqq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \right\}, \qquad i = 1, \ldots, n. \tag{3.34}$$

Furthermore, if k disks constitute a connected region but are disconnected from the other $n - k$ disks, then exactly k eigenvalues lie in this region.

The usage of the eigenvalues of $P_{UV}^T P_{UV}$ nevertheless takes a disadvantage in the fact that the smallest singluar value will be very badly conditioned in such a way that it will not be possible to give a nonzero lower bound for it. Thus another result is presented:

Proposition 7. *Suppose $A = (a_{ij}) \in \mathbb{C}^{m \times n}$. Provided*

$$r_i \coloneqq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|, \qquad c_i \coloneqq \sum_{\substack{j=1 \\ j \neq i}}^{m} |a_{ji}|, \qquad s_i \coloneqq \max(r_i, c_i), \qquad a_i \coloneqq |a_{ii}|$$

for $i = 1, \ldots, \min(m, n)$; for $m \neq n$ define

$$s \coloneqq \begin{cases} \max\limits_{n+1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}| & \text{for } m > n \\ \max\limits_{m+1 \leq i \leq n} \sum_{j=1}^{m} |a_{ji}| & \text{for } m < n \end{cases} \tag{3.35}$$

Then each singular value of A lies in one of the real intervals

$$\begin{aligned} B_i &= [(a_i - s_i)_+, a_i + s_i]\,, \qquad i = 1, \dots, n, \\ B_{n+1} &= [0, s]\,. \end{aligned} \tag{3.36}$$

If $m = n$ or $m > n$ and $a_i \geq s_i + s$ for $i = 1, \dots, n$, then B_{n+1} above is not needed. Furthermore, every component interval of the union of B_i contains exactly k singular values if it contains k intervals of $B_1, \dots, B_n$.

A sharper estimate is still possible:

Proposition 8. *The previous Proposition still holds true if B_i for $i = 1, \dots, n$ is replaced with $G_i = [(l_i)_+, u_i]$, where*

$$\begin{aligned} l_i &= \min\left(\sqrt{a_i^2 - a_i r_i + \frac{c_i^2}{4}} - \frac{c_i}{2}, \sqrt{a_i^2 - a_i c_i + \frac{r_i^2}{4}} - \frac{r_i}{2}\right), \\ u_i &= \max\left(\sqrt{a_i^2 + a_i r_i + \frac{c_i^2}{4}} + \frac{c_i}{2}, \sqrt{a_i^2 + a_i c_i + \frac{r_i^2}{4}} + \frac{r_i}{2}\right), \end{aligned} \tag{3.37}$$

and the non-real numbers in the previous formula can be omitted.

Proofs for the previous results, which are out of the scope of this work, along with deeper details and references can be found in [56].

Distances in statistics

Most multivariate analysis techniques are based upon the simple concept of distance. Straight-line Euclidean distance is unsatisfactory for most statistical purposes, because each coordinate contributes equally to the calculation of this metric. When the coordinates represent measurements that are subject to random fluctuations or differing magnitudes, it is often desirable to weight coordinates subject to a great deal of variability less heavily than those that are not highly variable: this suggests to adopt a different measure for distance, which accounts for differences in variations and to the presence of correlations.

The one of Mahalanobis is maybe the most important statistical distance, widespread in any field of analysis and easy to implement and understand. It is due to Prasanta Chandra Mahalanobis (1893-1972) and dates back to 1936. Given a dataset, represented as a point cloud in some vector space, the basic idea is to compute the covariance of each pair of variables and use it as a weight. The Mahalanobis distance can be seen as distorting the space of features in different ways towards different directions. It becomes thus interesting to investigate how this warping happens.

The Mahalanobis distance of an observation $\vec{x} = (x_1, \dots, x_N)^T$ from a set of observations with mean $\vec{\mu} = (\mu_1, , \dots, \mu_N)^T$ and covariance matrix S

is defined as:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}. \tag{3.38}$$

This can also be defined as a dissimilarity measure between two random vectors $\vec{x}$ and $\vec{y}$ of the same population owing the covariance matrix S:

$$d(\vec{x}, \vec{y}) = \sqrt{(\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y})}. \tag{3.39}$$

If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance; if the covariance matrix is diagonal, then the resulting distance measure is called a weighted Euclidean distance; if two observations in the point cloud are far along a direction which has low variability, their distance will be considered greater with respect to another pair showing the same separation along an axis which is covered by a wider distribution. Geometrically speaking, this means that the locus of equidistant points from a fixed origin becomes an ellipse. Consider Figure 3.10 for a better graphical explanation.

A final consideration regards normality: if data are normally distributed in any number of dimensions, the probability density value in correspondance of an observation is uniquely determined by the Mahalanobis distance d. In particular, the distance is proportional to the square root of the negative log likelihood. In general, given a Gaussian random variable X with variance $s = 1$ and mean $\mu = 0$, any other normal random variable R with mean μ_1 and variance S_1 can be defined in terms of X by the equation $R = \mu_1 + \sqrt{S_1}$. Conversely, to recover a normalized random variable from any normal random variable, one can typically solve for $X = (R - \mu_1)/\sqrt{S_1}$. If both sides are squared and the square-root is taken, this will result in an equation for a metric that looks similar the Mahalanobis distance:

$$D = \sqrt{X^2} = \sqrt{(R - \mu_1)^2 / S_1} = \sqrt{(R - \mu_1) S_1^{-1} (R - \mu_1)}. \tag{3.40}$$

Principal components analysis (PCA) is a statistical procedure that uses an orthogonal transformation of the covariance matrix to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called *principal components*. The number of principal components is less than or equal to the smaller of the number of original variables or the number of observations. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible). Each succeeding component has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set of the data space. It is important to notice that PCA is sensitive to the relative scaling of the original variables.

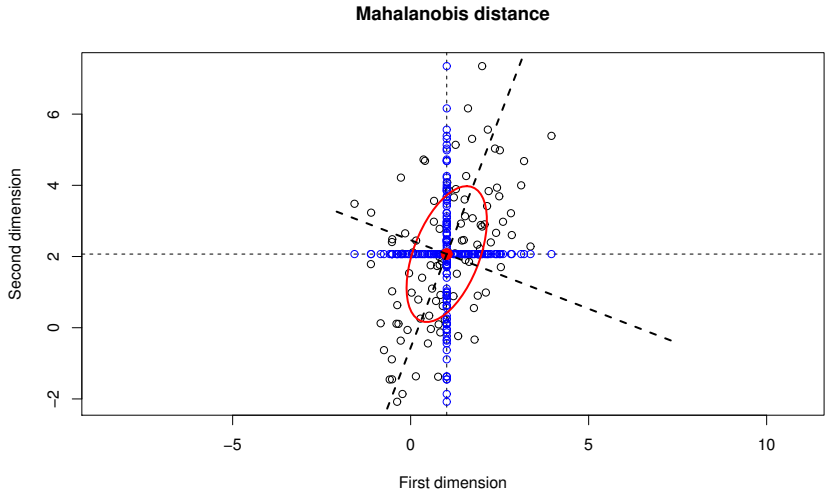

Figure 3.10: ***An example of Mahalanobis distance*** *– The red ellipse, which is centered on the mean of some randomly generated data, represents the locus with distance equal to one from it. The blue points on the horizontal and vertical axes are the projections of the point cloud on the two directions respectively. This makes clear that the variance along the two dimensions is different. The black lines instead represent the two principal components. Code for this plot can be found in Appendix A.*

In order to implement PCA, it is possible to use the eigendecomposition of the covariance matrix and find a pattern of orthonormal axes, representing a set of *fictional variables*, along which the distortion is stronger, meaning the variance is higher. These axes correspond to the first k eigenvectors of the matrix, which are related to the k eigenvalues which are greater in absolute value. The directions correspond to the principal components, while the reciprocals to the eigenvalues represent the scaling factor of axis warping with respect to those directions. This makes clear what the relation is between Mahalanobis' distance and the PCA.

The principal components analysis is really useful whenever the dimension of the data space is too great to be easily managed: the design parameter k can in fact be lower with respect to that dimension, thus this procedure can help in saving memory whenever the contribution in variance of the last directions is negligible, meaning they are poor of informations. Of course Mahalanobis distance along with PCA is not the only possible approach, in particular it does not allow a nonlinear generalization. A possible solution consists in the substitution of the usual scalar product in the data space with a nonlinear, kernelized scalar product. This, according to the theory in the previous paragraph, transforms the data space in a different Hilbert space, which metric is transformed consequently. The implementation consists in

a data trasformation step which maps the old feature space onto a new one, which can be linearly analyzed. This introduces an important advantage, consisting in the possibility of performing algorithms in a different space without having to modify them, thus allowing a deeper investigation (e.g. nonlinear regression) with essentially no computational effort even in presence of complex, high-dimensional feature spaces. For instance, according to [1], the upgrade for a linear support vector regression algorithm becomes no more complex than a value substitution, whereas a complete optimization algorithm on a deeply nonlinear structure would become an hard task. Nevertheless, many attempts can be done towards finding the best possible nonlinear transformation of data but no warranty is given about the compliance of the chosen kernel to the structure of data themselves; in particular, the choice of a kernel does not resolve the issue of dealing with anisotropic metrics.

The Gaussianization technique

Density estimation is a fundamental problem in statistics. In literature, the univariate problem is well-understood and well-studied [58, 59, 60]. Techniques such as variable kernel methods, Gaussian Mixture Models etc. can be applied successfully to obtain univariate density estimates. However, the high dimensional problem is very challenging, mainly due to dimensionality: data samples are often sparsely distributed, it requires very large neighborhoods to achieve sufficient counts and the number of samples has to grow exponentially according to the dimensions in order to achieve sufficient coverage of the sampling space. This however can be overcome by exploiting independent structures in data, by reducing the problem to a multiplicity of univariate problems along each dimension. [57] describes the gaussianization techniques for high dimensional estimation, but it is important to notice, according to what stated before, that the same methods allow for keeping a concept of neighborhood which is inherent to data.

Definition 3.7. For a random variable $X \in \mathbb{R}^N$, the *Gaussianization transform* is an invertible and differential transform $\mathcal{T} : \mathbb{R}^N \to \mathbb{R}^N$ such that the transformed variable $\mathcal{T}(X) \sim \mathcal{N}(\vec{0}, I_N)$.

Now, it is important to understand how to build such a function. For the scope of this work, only univariate gaussianization is used, but theory for the multivariate technique is presented in [57] and it could be interesting for immediate further developments in the method. In order to do so, some notation is required. $\phi(\cdot)$ will denote the probability density function of a standard multivariate normal $\mathcal{N}(\vec{0}, I_N)$, while $\phi(\cdot, \vec{\mu}, \Sigma)$ will describe the same for $\mathcal{N}(\vec{\mu}, \Sigma)$ with $\Sigma \in M^{N\times N}(\mathbb{R})$; then, $\Phi(\cdot)$ will denote the cumulative distribution function of the standard gaussian.

Definition 3.8. Let $X \in \mathbb{R}$ be a univariate random variable and let assume its density function to be strictly positive and differentiable; define with $F(\cdot)$ its cumulative distribution function. T is a Gaussianization transform if and only if it satisfies the following partial differential equation:

$$p(x) = \phi(T(x)) \left| \frac{\delta T}{\delta x} \right|. \tag{3.41}$$

This equation has a unique solution, except for the sign:

$$T(X) = \pm \Phi^{-1}(F(X)) \sim \mathcal{N}(0, 1). \tag{3.42}$$

In practice, $F(\cdot)$ is not available and it has to be estimated from the training data. Possible ways to proceed include a raw estimate by exploiting directly the quantiles of the sample cumulative distribution and Gaussian mixture models:

$$\begin{aligned} \hat{F}_q(x) &= \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{(-\infty, x]}(x_n), \\ \hat{F}_{GMM}(x) &= \sum_{i=1}^{I} \pi_i \Phi \left(\frac{x - \mu_i}{\sigma_i} \right); \end{aligned} \tag{3.43}$$

where N is the number of realizations of the variable, x_n represents each of the sample values, I is the number of gaussians in the mixtures and π_i, μ_i, σ_i are parameters that can be estimated via maximum likelihood using the standard expectation-maximization algorithm.

Chapter 4

Method

4.1 Motivation

This work has the dual objective of model individual music semantic based on user-provided song classification and define a personalized similarity function in the context of music recommendation. The algorithm will allow the comparison between class models belonging to either the same or different users, as well as the creation of a similarity measure regarding users themselves. The capability to compare users depending on their tastes in particular will exploit the generated class models in order to allow for collaborative filtering in a future recommending application. Moreover, it will be possible to exploit the same models in order to provide personalized content recommendation in the shape of automatic classification of songs along with rated label predictions.

The motivation for this work lies in the two-faced need for personalization in similarity modeling, as stated in Chapter 2. Indeed, researchers found a limit in approaches for music similarity which do not involve the study of the individual user in their analysis. At the same time, business requirements in the field of music recommendation constantly look for some improvement towards identifying the needs of each user. This personalization is more and more consideredamong the commercial services as it is proven to be a key point for their market success. It is clear, indeed, how different people owe different perceptions and tastes, especially in music consumption and listening experience. Nevertheless, users of a music database often show the same consumption habits, thus the difference among them lies in the subjective interpretation of what they listen to. Several factors contribute to form personal listening experience: demographic aspects like country or age as well as cultural level, musicological background and music attitude, just to cite some of them. The same factors also influence how people refer to music. Each user might provide personal meaning to labels and concepts used to describe music as well as the way similarity between songs is perceived:

for instance, a label like *chillout music* will rely on the individual idea of chilling out. This gives the reason for having this work scoped to individual semantic modeling, with the attempt to characterize how and what acoustic features are significant according to the meaning a user gives to music.

Apart from the business and musicological interest towards this subject, this work is intended also to investigate towards a novel and efficient method for music data preprocessing. Indeed, procedures for the management of nonlinearities achieved a strong relevance recently as a fundamental step for an accurate analysis, though some mathematical issues are still open or lie hidden in usual implementations. In this sense, the present follows a previous work: [1] indeed already assumes, mostly for the sake of simplicity, that there exists a linear relation between features and human perception, thus using linear functions to model the similarity among songs. However, this relation might be nonlinear; in particular here is shown that the same feature space does not owe a structure allowing to safely assume its linearity. A solution is thus purposed, aiming at overcoming these limitations: differently from previous works, this work will not introduce any arbitrary nonlinear kernel in the analysis, but exploits the inherent metric structure provided by data distributions. This reveals to be more flexible with respect to users' classification of music.

4.2 The method and its formalization

A full methodological description of the work will be provided in this section, along with the mathematical formalization of all the necessary steps. A high-level introduction to each component of the model will be interposed to the detailed analysis in order to understand what the main operations and the expected output are. On a very general perspective, as shown in figure 4.1, the main scope of the model is to collect information from users in the shape of *personal musical tags* and elaborate those in order to establish the hidden semantic logic for the data-label association: users are asked to associate personal tags to a predefined set of songs and the model will capture the different users' classification methods. This will happen by building a relation between the labels and the acoustic features of the songs. This relation may be exploited lately for classification purpose, with the prediction of new songs' individual tags, as well as concept comparison, meaning the identification of the similarity relation between labels derived from the similarity of the respective classification functions. The workflow diagram is summarized in Figure 4.2. The adoption of the mentioned approach entails that exploiting music tags solely is sufficient to describe the similarity concept for any user, as well as it looks forward to overcome the different ways of expressing music concepts. This can be due to either the users' cultural background or the purposes in listening music; the observation of individual

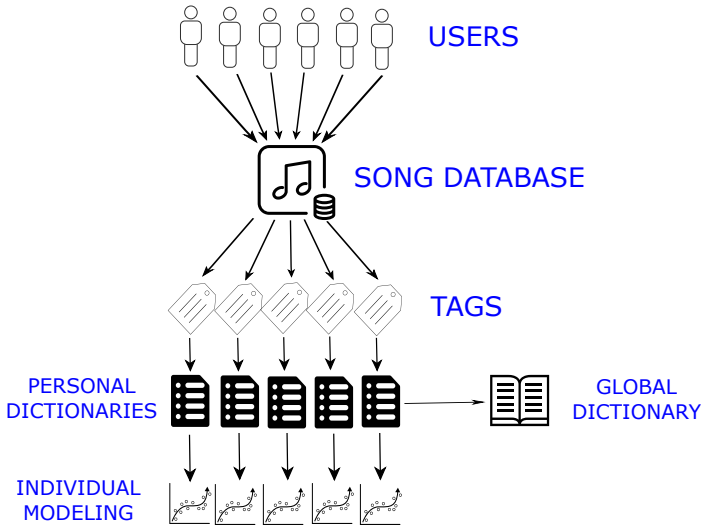

Figure 4.1: ***The information workflow*** *– The users are required to label a provided song dataset with personalized labels. This informations are then linked to acoustic data, obtained by songs through feature extraction, in order to provide individual semantical models.*

music tagging process lead to interesting observations which are collected in Section 4.4.

In order to build the model, it is at first needed a set of users, a set of songs and the possibility of collecting individual tags. Consider a triplet $(\mathcal{U}, \mathcal{S}, \mathcal{D})$. $\mathcal{U}$ is the set of *users*, while $\mathcal{S}$ is the set of all the songs in our dataset and $\mathcal{D}$ is the *dictionary*, the set of all the words that can be used to describe music. For each user $u \in \mathcal{U}$ it exists a *personal dictionary* $\mathcal{D} \supset \mathcal{D}_u = \{d_{u,1}, \dots, d_{u,N_u}\}$ made by the words that u uses to describe the songs in $\mathcal{S}$.

Once the setting is clear, it is necessary to define what the *personalized labeling* operation is and to understand how to relate individual labels to songs. The meaning of *individual* here lies in the fact that different users are allowed to autonomously choose the words they want to use for describing music, which are not necessarily shared among them:

Conjecture 1. Individual semantic assumption – In general $\mathcal{D}_u \neq \mathcal{D}_v$ for $u, v \in \mathcal{U}, u \neq v$.

This means that each user is free to associate any song in the dataset to whatever label in his own personal dictionary:

Definition 4.1. We can define the *personal labeling* as a relation

$$\mathcal{L} : \mathcal{U} \times \mathcal{S} \to \mathcal{P}(\mathcal{D}),\ (u, s) \mapsto \mathcal{D}_{u,s} \subset \mathcal{D}_u. \tag{4.1}$$

A remark here is necessary:

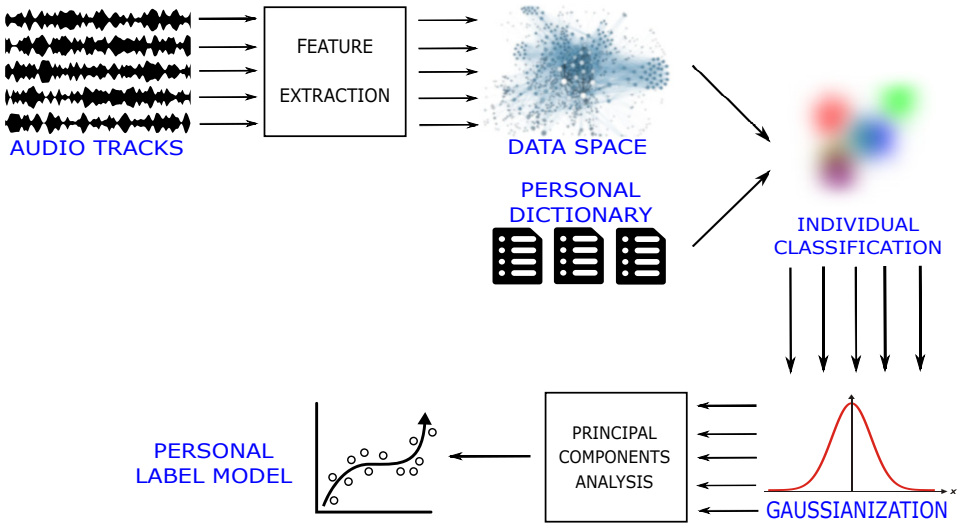

Figure 4.2: ***The block diagram*** *– This figure shows the detail of data processing: songs are first acquired and feature acquisition is performed. This generates a data space, which points are classified once for each user, according to his personal dictionary (cf. Fig. 4.1). Data are then classwise remapped through Gaussianization and a final linear model is generated for further uses.*

Remark. (Multi-labeling) In general $\mathcal{L}$ is **not** a function, since it is possible for the same user to describe the same song with different words belonging to $\mathcal{D}_u$; thus its output is a set and not a single point.

This assumption has not been considered during the development of this work for the sake of simplicity in data acquisition. However in principle the model is flexible to manage multi-labeled songs, this in order to be more precise in the analysis and allow *fuzzy* labeling. This is necessary since people may apply to songs different concepts or genres.

As far as data modeling is concerned, the objective is to relate the collected labels to the acoustic features of songs. This introduces a new data collection and analysis phase, oriented to the acquisition and structuring of feature data from music. In order to perform this step, it comes necessary to define an algebraic structure which is suitable for the operations described in Section 4.3: consider $\mathcal{S}'$ as a N_f-dimensional linear space, the dimensions of which consist in the acoustic features measurements. For each song $s \in \mathcal{S}$, a set of K music excerpts is extracted and $\vec{s_j} \in \mathcal{S}'$, $j = 1, \dots, K$ represents a complete set of features extracted for each song segment. Consequently, for each song $s \in \mathcal{S}$ it exists a set $\mathcal{S}' \supset \bar{S} = \{\vec{s_j}\}$.

Definition 4.2. The *feature acquisition* map is a function

$$\mathrm{FA} : \mathcal{S} \to \mathcal{P}(\mathcal{S}'), \;\; s \mapsto \bar{S} \tag{4.2}$$

representing the composition of the segmentation and actual feature extraction.

It may be interesting to investigate the characteristics of the FA map, making it compliant with the practical process in 4.3. In particular, the injectivity of the sole extraction phase would be a desirable property in order to avoid local overlapping between songs representatives in the linear space. It is also rather reasonable to assume it, since completely identical acquired features should correspond to perfectly equal acoustic properties, meaning that the musical excerpts analyzed are composed out of the same sounds. Thus the following

Conjecture 2. Unicity assumption – The map $\mathrm{FA}(\cdot)$ defines a partition of $\mathcal{S}'$ such that any $s_j \in \mathcal{S}'$ corresponds at most to only an $s \in \mathcal{S}$.

Unfortunately, in practice it is not always possible neither to assume nor to check directly that the same measurement is not repeated in the database, as it will be shown later. Thus, a lighter formulation of this approach is needed. As long as the work is not focused to individual songs but rather to model global sound properties, it is sufficient to require that, if two identical samples are present, the songs they come from must be identified by the same label. Of course, for generality purposes this should hold for all of the users, namely

Conjecture 3. Relaxed unicity assumption – If $\exists\ s_j \in \mathcal{S}'$, $s, t \in \mathcal{S}$ such that $s_j \in \mathrm{FA}(s) \cap \mathrm{FA}(t)$ then, $\forall u \in \mathcal{U},\ \mathcal{L}(u, s) \equiv \mathcal{L}(u, t)$.

The following step will be the connection between user- and content-originated data: given the user u, it is necessary to associate to each of his N_u individual labels $d_{u,i}$ the corresponding data in $\mathcal{S}'$. This will of course be done by exploiting the personalized labeling relation; in particular, it will be necessary to apply the feature acquisition to the songs which the user labeled with each of the considered tags. We give thus two definitions for the dataset, a first formal and a second operative.

Definition 4.3. Provided a user $u \in \mathcal{U}$ and one of his labels $d_{u,i},\ i \in \{1, \dots, N_u\}$, the *labeled data* consists in the feature acquisition applied to the preimage of $d_{u,i}$ through the restriction to user u of the personalized labeling relation:

$$\mathbf{d}_{u,i} \coloneqq \mathrm{FA}(\mathcal{L}^{-1}(u, \cdot)[d_{u,i}]). \tag{4.3}$$

An equivalent implementative definition is

$$\mathbf{d}_{u,i} = \{\vec{s_j} \in \mathcal{S}' \mid \exists\ s \in \mathcal{S},\ d_{u,i} \in \mathcal{L}(u, s),\ \vec{s_j} \in \mathrm{FA}(s)\}. \tag{4.4}$$

$\mathbf{d}_{u,i}$ is a multivariate dataset with $N_{u,i}$ samples and N_f features, each variable of which owing a different metric. It is not possible to make any prior assumption neither over its (univariate or joint) distributions nor on the metric to be adopted. Moreover, the data space is not even a linear space, because some of the variables cannot take negative values. This creates the necessity for the implementation of a model which should reconstruct the metric over the space of observation. The metric should depend on the observations in order to account for the different data structures, according to the classification the users apply to data:

Conjecture 4. Modeling assumption – $\forall u \in \mathcal{U},\ i \in \{1, \dots, N_u\}$, we can model $\mathbf{d}_{u,i}$ and define a distance on $\mathcal{S}'$, $\mathbf{D}_{u,i} : \mathcal{S}' \times \mathcal{S}' \to \mathbb{R}^+$ which depends on the chosen $(user, label)$ pair.

In order to build different metrics for different datasets, the only available information to rely on is the distribution of each over the space $\mathcal{S}'$, with no prior assumption on the shape data would take. Thus the following

Conjecture 5. Data-related metric assumption – For each $(user, label)$ pair (u, i), $\mathbf{d}_{u,i}$ owes an inherent metric structure which can be collected out of the probability density functions $f_{u,i,v}(\cdot),\ v \in \{1, \dots, N_f\}$ on the current univariate spaces.

In statistics, the *standard score* is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured. Observed values above the mean have positive standard scores, while values below the mean have negative standard scores. The standard score is a dimensionless quantity obtained by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. This conversion process is called *standardization* or *normalization*. This method sets the basis for the execution of hypothesis tests, since standard scores are most frequently used to compare an observation to a standard normal deviate and they can be defined without assumptions of normality. The process of course does not change the structure of data in terms of the probability density function. The further information provided by the density itself may be of particular importance, since it conveys information that are not fully captured by the simple covariance parameter. This can be seen in Figure 4.3, where two different distributions are compared: they owe the very same covariance matrix, equal to the idenitity, but do not show the same plot and the data structure is actually different. Usually, the metric considered for data processing is the Mahalanobis' one described in Section 3.4.2, which only relies on the covariance structure of data. As stated there, this is actually effective in terms of univokely modeling data when the variables are normally distributed, since this is the only case in which a direct relationship holds between distance and probability distribution. This makes the

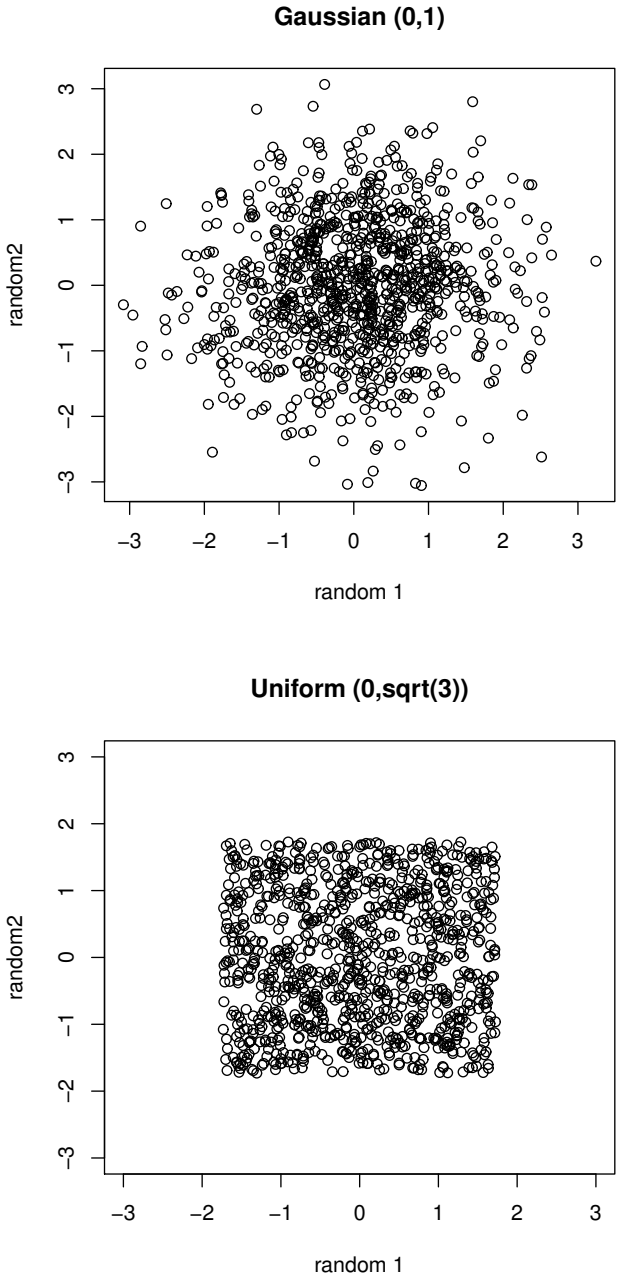

Figure 4.3: ***Different distributions, same covariance*** *– These plots show two bivariate datasets: the first is obtained out of two uncorrelated gaussians random variables with null mean and variance equal to* 1*, the second with the same procedure, but the RVs are uniform, null mean and range* $\sqrt{3}$*, which still corresponds to variance equal to* 1*. Their covariance matrices both correspond to the identity, thus the PC decomposition is the same.*

reason for change the usual data preprocessing step from standardization to Gaussianization:

Conjecture 6. Metric building assumption – It is possible to redefine the metric structure of the data space to be Euclidean by univariate Gaussianization of the features.

Via univariate Gaussianization we obtain a N_f-dimensional Hilbert space, in which the usual inner product represents a user-label-related covariance kernel. Data are transformed according to this, leading to an Euclidean space within which usual covariance analysis is possible.

Definition 4.4. For $v \in \{1, \dots, N_f\}$ and u, i as usual, the *univariate Gaussianization transform* is $T_{u,i,v} \coloneqq \Phi^{-1}(F_{u,i,v}(\cdot))$, where Φ represents the cumulative distribution function of a standard Gaussian random variable and $F_{u,i,v}$ is the empirical cumulative distribution function obtained out of the v-th variable of $\mathbf{d}_{u,i}$.

Definition 4.5. The *user-label covariance kernel* transformation is the composition of the independent univariate transforms:

$$T_{u,i} : \mathcal{S}' \to \mathbb{R}^{N_f},\ T_{u,i}(\vec{x}) \coloneqq [T_{u,i,1}(x_1), \dots, T_{u,i,N_f}(x_{N_f})]. \tag{4.5}$$

Remark. The user-label covariance kernel always exists and it is unique, provided the cumulative distribution function of data. Moreover, the Gaussian cumulative distribution is an injective function along with its inverse, thus, the only way to have $T_{u,i} = T_{v,j}$ for $u \neq v \in \mathcal{U}$ is to have $d_{u,i} = d_{v,j}$.

The metric in the transformed features space is well known and clear, then it is interesting to understand the kind of metric transformation induced on $\mathcal{S}'$. In particular, it is possible to exploit the usual kernel approach in order to describe the user-label metric on the old feature space, $\mathbf{D}_{u,i}$, by the shape of its unitary open ball centered in s:

$$\mathcal{B}(s, 1) = \left\{ s' \in \mathcal{S}' \middle| \left\| T_{u,i}(s) - T_{u,i}(s') \right\|_2 = 1 \right\}. \tag{4.6}$$

More in general, this procedure allowed to finally define

$$\mathbf{D}_{u,i}(s, t) = \left\| T_{u,i}(s) - T_{u,i}(t) \right\|_2. \tag{4.7}$$

Remark. Due to the high nonlinearity of map $T_{u,i}$, $\mathbf{D}_{u,i}$ is not isotropic, thus it is not certain that triangle inequality holds. For sure instead positivity and symmetry do, so in general we can only state that $\mathbf{D}_{u,i}$ is a *semimetric*.

Unfortunately, it is pretty difficult to show the shape of a unitary ball in the new metric with respect to the old one (Euclidean on $\mathcal{S}'$) due to the fact that it is only possible to know samples of $F_{u,i,v}(\cdot)$, but not the whole function, if not approximately. What is it actually feasible to compute with

precision is the opposite procedure, as to say the transformation of a unit ball in the old feature space to the new one: it is immediately noticeable in Figure 4.6 the high level of nonlinearity that this method is able to capture. Moreover, the shape may vary dependent of the input data distributions. Now the data have been preprocessed in a convenient way to manage their nonlinearity and this new space can be exploited for the practical scope of the work: the creation of a linear model for the characterization of the label $d_{u,i}$, which allows easy comparison in the definition of a semantic similarity measure.

For each pair $(user, label) = (u, i)$ a new sampling $\mathcal{S}_{u,i} = T_{u,i}(\mathbf{d}_{u,i})$ of $\mathbb{R}^{N_f}$ is defined. It is now possible to perform dimensionality reduction through principal components analysis and obtain a final model $\hat{\mathcal{S}}_{u,i}$ of $d_{u,i}$. This consists in the orthonormal basis of the transformed feature subspace which is maximally relevant in terms of explained data variance, with a chosen explained variance threshold of 90%. The matrix M_{ui} will collect the basis vectors, and it is noticeable that the column space dimension of $\hat{\mathcal{S}}_{u,i}$ is variable depending on how many *pseudovariables* are actually relevant for explaining the dataset variance.

In order to compare labels, we can make a projection of a label subspace onto another one and find the multidimensional angle between them. The bigger it is, the less the labels depend on one another, because they approach orthogonality. The arcsine of that angle will provide a measure for label similarity:

Definition 4.6. For each pair of possible label models $\hat{\mathcal{S}}_{u,i}, \hat{\mathcal{S}}_{v,j}$, their *label-to-label semantic similarity* λ_{ij} is defined as

$$\lambda_{ij} \coloneqq 1 - \frac{2}{\pi} \arcsin(\min(1, \max_{n}(\sigma_n))). \tag{4.8}$$

Here, σ_n represents the singular values of the matrix

$$P_{u,i,v,j} \coloneqq M_{ui} - M_{vj} M_{vj}^T M_{ui}, \tag{4.9}$$

constituting the projection of model $\hat{\mathcal{S}}_{v,j}$ onto $\hat{\mathcal{S}}_{u,i}$.

The item similarity identification task is concluded with this definition, because it is now possible to take any song, then transforming its acoustic features according to any possible user identified music group and state if it is compliant or not, in order for instance to provide for personalized music genre prediction. The compliance can be determined by measuring how the data metric defined by the song itself is similar to the label model, as well as this method can be used more in general to have a similarity index for two general songs. The approach is similar to traditional cosine similarity, but it is supposed to work better in determining the relevant correlations between variables.

An important task in music recommending, other from item similarity, is user similarity identification. Considering now that the knowledge of the user is based on the provided labels along with their models, it is possible to suggest a method based on the previous procedure in order to provide also this information:

Definition 4.7. Consider two users $u \neq v \in \mathcal{U}$ along with their label models $\hat{\mathcal{S}}_{u,1}, \dots, \hat{\mathcal{S}}_{u,N_u}$ and $\hat{\mathcal{S}}_{v,1}, \dots, \hat{\mathcal{S}}_{v,N_v}$. We define the *inter-label similarity matrix* $\mathrm{USM}_{u,v} = [\lambda_{ij}]$ as the matrix containing the pairwise label-to-label semantic similarities for $i \in \{1, \dots, N_u\}$, $j \in \{1, \dots, N_v\}$.

Now it is necessary to define how to extract a general users' behavior information out of label-to-label similarities. The principle is to exploit again the power of linear algebra with the following:

Conjecture 7. User comparison assumption – The more two users' labels are similar to each other, the more the same users will be similar as well and vice versa. This argument is included in the SVD analysis of their inter-label similarity matrix.

Remark. Of course, the similarity between a user and himself should be maximal, namely equal to 1.

This leads to

Definition 4.8. The overall *user similarity function* is defined as

$$\mathrm{usim} : \mathcal{U} \times \mathcal{U} \to [0,1],\ (u,v) \mapsto \frac{\sum_{i=1}^{\min(N_u,N_v)} \sigma_i}{C}, \tag{4.10}$$

where σ_i are the singular values of $\mathrm{USM}_{u,v}$ and C is a normalization constant depending of some Gershgorin-like estimate chosen under the constraint $\mathrm{usim}(u,u) = 1\ \forall u \in \mathcal{U}$.

A basical requirement for checking if the method is working in modeling users lies in is its capability to understand the labeling rule. Another procedure has been identified under the following hypothesis:

Conjecture 8. User acknowledgement assumption – The label modeling separates real user-crafted labels from randomly machine generated ones.

In order to check for this, there must be a higher-level user structure which identifies the human behaviour not to be casual. This should be related to an overall acoustic feature map, which transcends single label-to-label relationships towards being more explicative of the individual music perception. This work aims at capturing subjectivity by exploiting the structure of the labeled data: this will be the starting point for further analysis. The formal translation of this reasoning is the concept of *user-characteristic feature space.*

Definition 4.9. Given a user $u \in \mathcal{U}$ and his label models $\hat{\mathcal{S}}_{u,1}, \ldots, \hat{\mathcal{S}}_{u,N_u}$, we define the *characteristic space* $\mathbf{\Gamma}_u \coloneqq \text{rank}\left\{\hat{\mathcal{S}}_{u,1} + \cdots + \hat{\mathcal{S}}_{u,N_u}\right\}$.

As long as we are working on transformed features space with the same structure, a way to operate for comparison of user-characteristic spaces follows: it will be sufficient to proceed with the same algorithm used for inter-label similarity determination in order to understand if users are somehow related or orthogonal to each other in their determination:

Definition 4.10. For each pair of possible user characteristic spaces $\mathbf{\Gamma}_u$ and $\mathbf{\Gamma}_v$, their *characteristic space similarity* $\gamma_{u,v}$ is defined as

$$\gamma_{u,v} \coloneqq 1 - \frac{2}{\pi}\arcsin(\min(1, \max_n(\sigma_n))). \tag{4.11}$$

Here, σ_n represents the singular values of the matrix

$$P_{u,i,v,j} \coloneqq M_{\Gamma_u} - M_{\Gamma_v} M_{\Gamma_v}^T M_{\Gamma_u}, \tag{4.12}$$

constituting the projection of model $\mathbf{\Gamma}_v$ onto $\mathbf{\Gamma}_u$.

Remark. As long as the user-characteristic space represents a linear transform to $\mathbb{R}^{N_f}$, an equivalent definition could exploit the ker of the same subspace of $\mathbb{R}^{N_f}$.

The next Sections will focus on the data processing steps, along with the description of the issues met in applying this whole procedure to actual training data for modeling and prediction. The accurate description of the tests performed in order to grant the quality of the model and the respective results are instead given in Chapter 5. Instead, the R code developed for generating the figures used in the present Section is provided in the Appendix.

4.3 Content data collection and management

The dataset which was used for the development of this work is a MIR standard called Computer Audition Lab 500-song (CAL500). It has been developed within the research in music semantic description (see [20] for details) and it consists of 500 popular western music songs collected as .mp3 files in order to address the shortcomings of noisy semantic data mined from text-documents. The songs in the dataset span across the last 50 years of music production, corresponding to an heterogeneous set of tracks covering different genres and sonorities, thus allowing a widespread evaluation. Some issues emerged during the song processing, it is known for instance that some of the present audio files are partial or corrupted[1] and this had been addressed during a first data preprocessing stage.

[1] http://media.aau.dk/null_space_pursuits/2013/03/using-the-cal500-dataset.html contains some examples

The first step of content data analysis consists in a process called *segmentation*. From everyone's music experience, it is clear that songs are characterized by different events in their temporal evolution: these could correspond to changes in the instruments played during its different parts, in its tempo or dynamic properties and, more in general, in its sonority. Noticeably, this happens especially for some kind of music genres, like jazz for instance. Nevertheless, a common audio file contains several information which makes it too large for the complete evaluation of its properties all along the duration. In order to address this processing issue, all of the audio files in the dataset had been segmented by extracting $K = 35$ audio excerpts spanning 3 seconds of the song each. The starting time instants of the segments have been choosen by sampling a uniform random distribution on the whole duration of the songs, as it can be seen in the code included in Appendix A.

In order to perform feature extraction, a python script provided in [2] has been used. This exploits the Librosa python library for sound processing, along with a series of VAMP plugins[2] elaborated by different institutions (e.g., the Queen Mary University of London and BBC R&D Dept.). These perform actual song annotations of the provided audio tracks and the result are output as .csv files. Table 4.1 shows the plugin bundles where the respective algorithms for feature extraction can be found. As described in Chapter 3, each song in the database is represented by features extracted from audio tracks and combined into vectors. According to the feature's value type, this operation may or may not require averaging on the audio frames, and may return a single representative value or a vector of values.

Referring to what previously in this Chapter and due to the dataset issues, some preprocessing has been necessary. In particular, some data samples revealed to be corrupted, as well as some of the song segments showed unacceptable acoustic characteristics. For instance, many songs, due to *fade-in* or *fade-out* sound effects, revealed in a prolonged recording of silence, which is useless in data analysis or rather harmful; indeed, feature extraction out of almost silent excerpts is reflected into a massive presence of either null or *not applicable* (NA) values. This phaenomenon helps to justify the Relaxed Unicity Assumption (Conjecture 3); moreover, it may be impossible to check whether the songs owing a silent segment are actually labeled in the same way by all of the users: the set $\mathcal{U}$ may in fact be too large to perform this. A solution to fulfil the assumption is to discard any measurement violating it. This is not a big deal, in fact the event may appear within songs showing in general very different properties. In the specific case, an almost silent segment collected from a rock song could exhibit the same features as another extracted from a classical music piece. Thus it is not the labeling procedure to be uneffective, but the measurement itself is

[2] http://www.vamp-plugins.org/ is the website of VAMP project

Plugin bundle	Features
Libxtract[1]	Average deviation, Crest, Irregularity J, Irregularity K, Kurtosis, Loudness, Mean, NonZero Count, Odd-even ratio, Rolloff, Sharpness, Skewness, Spectral centroid, Spectral flatness, Spectral inharmonicity, Spectral kurtosis, Spectral skewness, Spectral slope, Spectral smoothness, Spectral spread, Spectral standard deviation, Spectral variance, Tristimulus, Variance
Queen Mary[2]	Chromagram, MFCC coefficients
BBC[3]	Energy dip probability, Intensity, Intensity Ratio, RMS energy, RMS energy delta, Spectral contrast, Spectral flux

[1] https://code.soundsoftware.ac.uk/projects/vamp-libxtract-plugins
[2] http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html
[3] https://github.com/bbc/bbc-vamp-plugins

Table 4.1: Summary of the VAMP plugins used for extracting features from audio-tracks.

untrustable and could be deleted.

A last technical aspect to take care of is the one regarding the Gaussianization process. When applying the raw estimation technique for the empirical cumulative density function of data, indeed, the output (estimated ECDF) will consist in a stepwise function assuming value equal to 0 before the minimum sample value and 1 after the maximum, as shown by Equation 3.43. The successive application of the Gaussian ECDF $\Phi(\cdot)$ will map the corrisponding data points respectively to $-\infty$ and $+\infty$, which represent the quantiles of the Gaussian random distribution itself. This of course is not acceptable, since it means to generate a pair of infinite values everytime a variable is analyzed, and this unbounded mapping does not reflect the behaviour of data. The solution consists in adding to the measurements a pair of values $-\infty, +\infty$ as placeholders for correctly mapping the intermediate data; this will as well remap the quantiles of the estimated ECDF, that needs to be corrected accordingly. Doing so, the new raw estimate for the ECDF

of the measurements becomes

$$\hat{F}_q(x) = \frac{1}{N+1}\left(\frac{N+2}{N}\sum_{n=1}^{N}\mathbb{1}_{(-\infty,x]}(x_n) - 1\right). \tag{4.13}$$

Of course, the artifact consisting in the infinite values previously added has to be removed prior to the further processing.

4.4 User data collection and overview

The user data collection was performed manually, by filling a table containing user labels, which were provided for a subset of the song database during a perceptual survey. The individual execution of the survey did not require predefined dictionaries or topics as basis for the users to discern songs, but allowed them maximum freedom in supplying the musical descriptive tags they perceived as the most appropriate. In particular, music genre description constituted the main focus in the evaluation, even if it was not explicitly asked.

Users' interaction with the dataset happened in a neutral context, in order to be as compliant as possible to the usual listening approach of the subjects: they were asked to listen to a song until they were able to provide a label for that, unless they desired to explore it better or asked to skip it since no evaluation was possible. In the former case, the listening restarted by skipping to another random point within the song, otherwise the provided label would be recorded as *non applicable* (NA). Users were not allowed to know any metainformation on the song, like for instance the artist, the title or the release year, previously to providing their evaluation; this in order not to influence their decision with other than the acoustic properties of the songs. In particular, from user feedback it was acknowledged that cultural metainformation like the song year proved to be relevant for their classification process.

Some observations on the data allow to better understand the reason for collecting information in this way. A first is that users' perception of their classification criteria improved along the analysis: the first provided cathegories always tend towards generality and are poorly personalized, oriented to identify a common-sense definition of music genre in a wide shape. This results in popular labels like *pop* or *rock* which appear constantly across the different people involved. While going further, the labels specialize instead into personal semantic related details, with deep characterization of the emotional impression coming from the songs. For instance, it happened often to have the *pop* concept to be divided into more cathegories, like *melodic* or *easy listening*, while also the *rock* label spreads across individual subgenres like *alternative*, *hard* or *progressive*. Others provide a classification based on a *sound-likes-those-times* basis. This of course generates a wide

range of possibilities, as expected, and corresponds to a fragmentation in a relevant number of tags which may differ a lot among users, namely from 10 to more than 40 different labels. Many of these, due to the nature and the size of the dataset, are scarcely represented and this could generate an issue in modeling due to few data samples. In order to prevent the issue, a final step in the survey involved the collection of the poorly characterized labels for users' review: whenever applicable, people was asked to make label association in order to join concepts that were too similar to be analyzed separately. Songs out of this logic were simply marked as *non applicable* and kept for further label prediction. An important consideration in this sense, coming from users' feedback, consists in the the dataset being felt as poorly exploratory: even not experienced users perceived the dataset, made only of western popular music, as if songs were globally similar, meaning that their concept analysis had to go deep into music structure in order to determine an effective difference.

Data coming from surveys had been processed as mentioned: the result consists in the *non applicable* songs to be classified, according to the described projection method, and paired to all of the individual user's possible labels along with a *similarity score*. This has been done for all of the users on a second *test* dataset of 20 songs, which was built in order to allow users to provide their feedback on the overall classification result; moreover, in order to automatically test the algorithm, the whole dataset was divided multiple times into a *training* and a *testing* part in the proportions of $80\% - 20\%$ of the songs. The objective is to check the performance of the method, by training the model with the first part of the dataset and to provide predictions for the second: the difference between the real classified data (test dataset) and the extimates made by the algorithm will provide a measure for the performance of the classifier. In Chapter 5 we will have a description for this method, in order to verify that the method is effective and is able to model the users' perception of music similarity.

4.5 Issues

A relevant part of the work has been oriented to provide a sufficient amount of information inside the dataset, in terms of samples of music excerpts pertaining to each of the individual labels. This indeed revelead to be an important parameter in order to perform a correct analysis: the dimension N_f of the data space conditions the number of points that should sample it, each corresponding to a song segment to be analyzed. The necessity of having a minimum number of the latter forced the choice to collect at least 3 songs per label and to extract $K = 35$ segments per song, in order to grant the points to sufficiently explore the space. Moreover, this has an influence also on the global number of songs to be listened to, with respect to the

number of labels the user provides: songs should populate adequately every new definition. A poor characterization for the experienced users, whose evaluation is more punctual and discriminative, follows from this reasoning.

If the previous can be seen as an under-sized version of the mentioned *cold start* problem, it is important to notice that the system deals with the *long tail* issue. In a first attempt to implement the prediction step, indeed, a different model was provided with respect to the one described above; this working through the analysis of the variation induced in a label model by a new song which is added to it. The method was ill-posed with respect to the sample dimensionality, because the abundance of data characterizing a given, popular label reflects into the little impact of new data on the overall distribution. In practice, this results in a predictive bias towards popular labels, which keep being similar to themselves more than the poor ones whenever a new song is assigned to them. This *rich-gets-richer* effect resembles the usual mentioned for standard collaborative filtering algorithms.

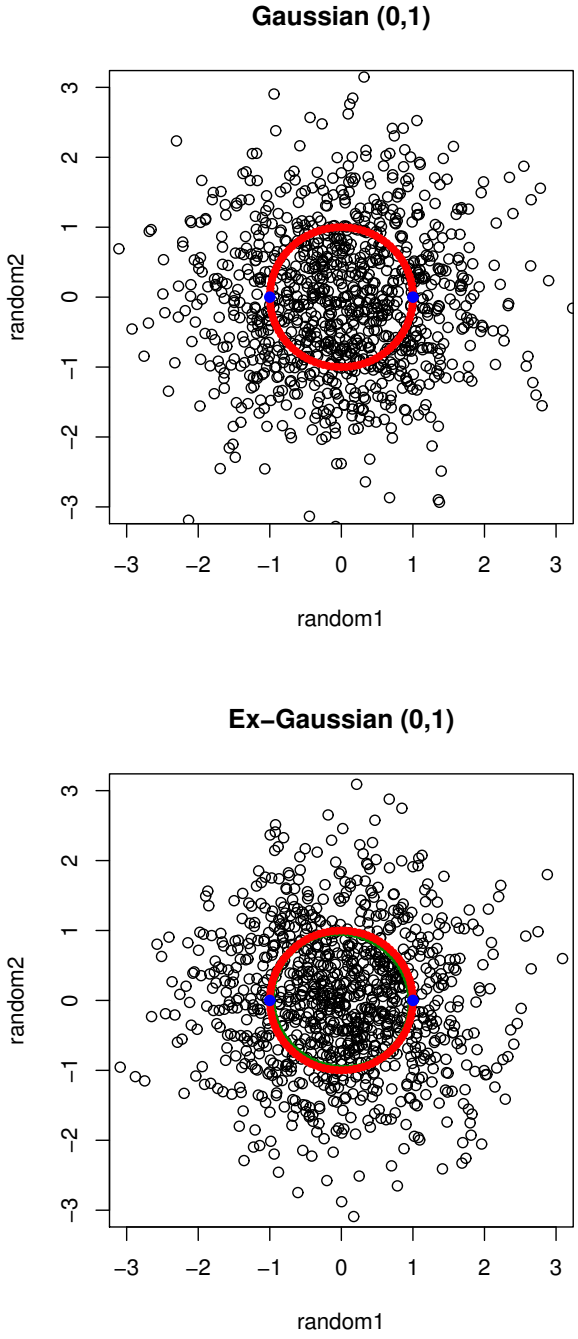

Figure 4.4: ***Metrics comparison (1)*** *– This plot shows the bivariate independent gaussian dataset in Figure 4.3 and its transform: the red circle on top represents a unit ball in the old metric, which comes nonlinearly reshaped in the figure at the bottom. In the case of independent RVs, the nonlinear map is actually the identity.*

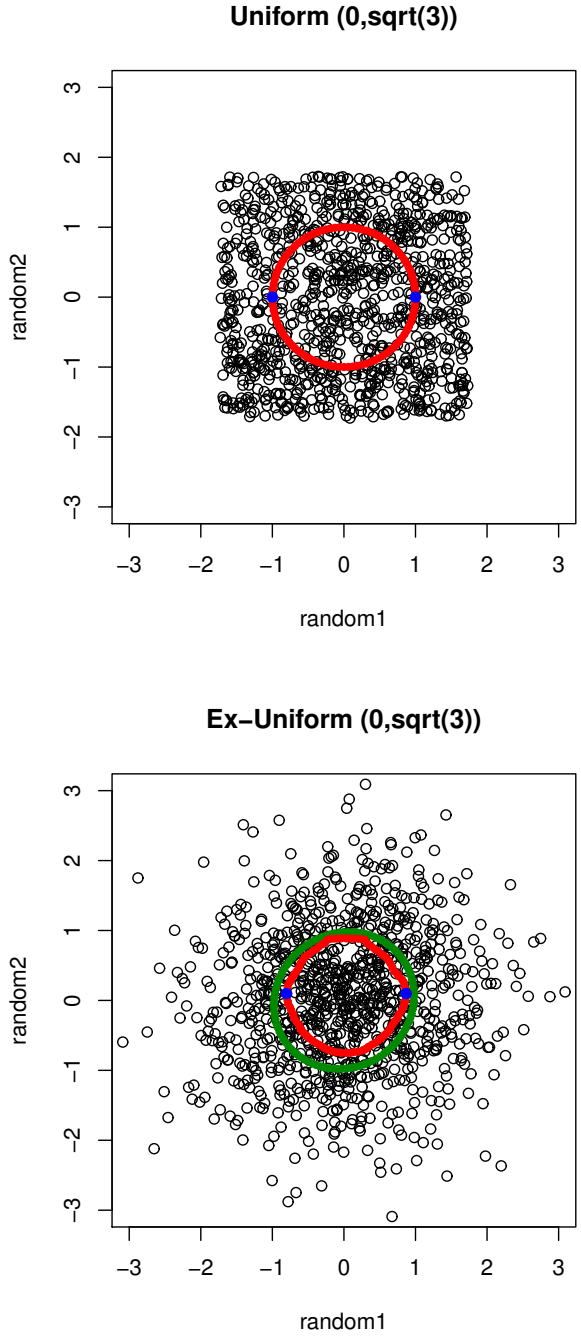

Figure 4.5: ***Metrics comparison(2)*** *– This plot shows the bivariate independent uniform dataset in Figure 4.3 and its transform: the red circle on top represents a unit ball in the old metric, which comes nonlinearly reshaped in the figure at the bottom. In the case of independent RVs, the circle keeps it shape, but the radius is reduced.*

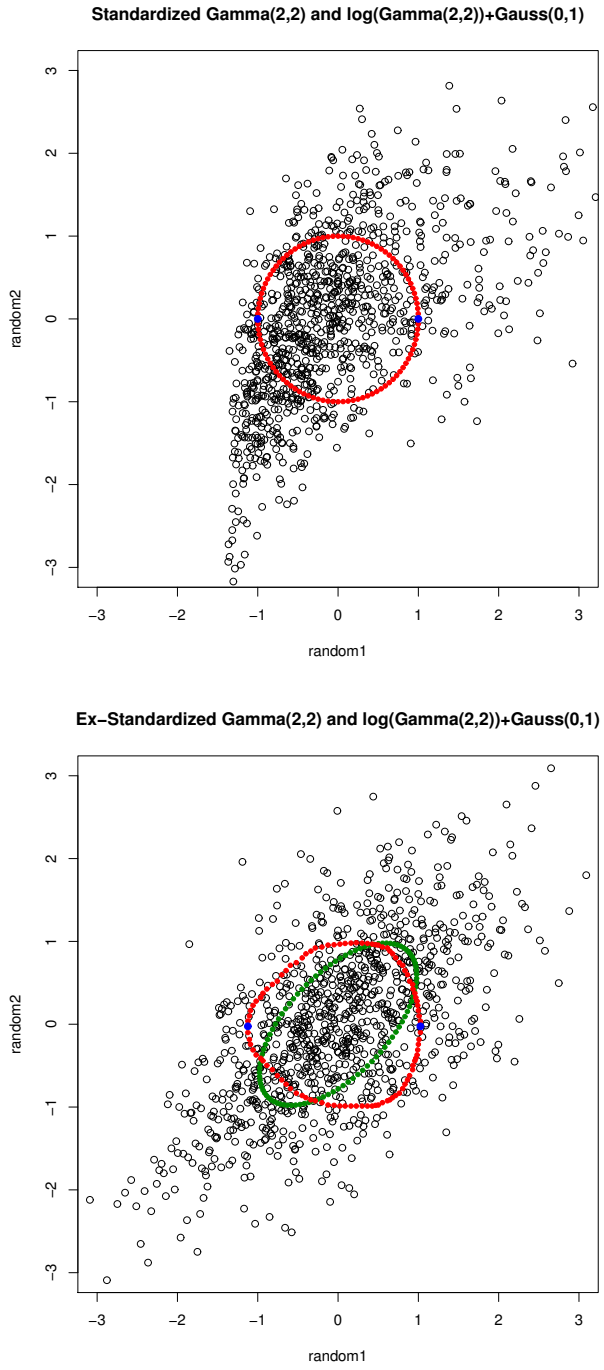

Figure 4.6: ***Metrics comparison(3)*** *– This plots shows a correlated dataset made out of* $X \sim \Gamma(2,2)$ *and* $Y = \log(X) + Z$, $Z \sim \mathcal{N}(0,1)$, *along with its transform. The red circle on top represent a unit ball in the old metric, which comes nonlinearly reshaped in the figure at the bottom. Notice that, in the case of dependent RVs, circles do not keep their shape. In particular, the green ellipse shows the shape of a confidence region.*

Chapter 5

Experimental results

The previous chapter described the model in deep, along with the characterization and management of the collected user and content data which were needed to train it. The result of this elaboration is showed hereafter, where a thorough evaluation of the involved algorithms is described along with its conclusions on the effectiveness of the provided method.

Four different tests were conducted in order to assess the precision of the method in describing the semantic music similarity for the different users: the first two involve directly the comparison of individuals, by presenting two standalone analysis on the discriminative capability of the model in separating real users from randomly simulated fake individuals; a third instead assesses the correctness in prediction by exploiting the usual training-testing division of the dataset. The three described tests provide an *objective* evaluation of the algorithm. Nevertheless, because this work aims at the modelling of the single user, a *subjective* evaluation is also needed in order to understand if this method correctly deals with the personalization issue. Thus, a further test was performed on the basis of new data.

5.1 Objective evaluation

The objective evaluation of the model consists in all of the procedures applied in order to allow for automatic self-evaluation of the performance based on the collected data. This kind of assessment can be done in different ways, by using either the dataset as a whole or by splitting it into two parts. The first option is to be chosen while evaluating the data processing procedures; the second instead is preferrable in the case the test deals with the measurement of the prediction performances of the algorithm: the first part of the dataset will be used to train the model, the second instead will be used for predicting the values of a variable according to the model previously trained. The real values taken by the variable will provide a ground truth to be compared with the predicted values in order to check for their compliance.

A first test that was performed used the whole dataset as training samples. This aimed to understand the correctness of the model in individuating a real user behaviour by testing the similarity of the users' characteristic spaces: for each user, a *phantom user* has been generated by applying his same tags to random songs in the dataset. The phantom user results then in a fake user profile, which labels are the same of the correspondant real user but with a complete rearrangement of the music categories in the feature space. In order to understand if the model is able to distinguish the real user from his phantoms, the characteristic space from Conjecture 4.9 was used. The test consists indeed in measuring the characteristic space similarity γ_{u_r,u_p} between a real user model $\mathbf{\Gamma}_{u_r}$ and the correspodant phantom models $\mathbf{\Gamma}_{u_p},\ p \in \{1,\dots,P\}$. This first test did not give positive results in discriminating real users from phantoms: a deeper analysis on the testing hypothesis shows that the characteristic spaces originate from a basis of the sum of other label-specific linear subspaces. Those are the images of a series of different non-linear mappings of the same original space, thus they can neither be compared with each other nor be summed: the only condition for having them compareable would be to have the transformed feature mapping to be the same. This means that the songs would define the same feature distribution on the space once provided the label. Thus, the choice of the label for each user would be independent of the feature distribution, which is against the preliminary hypotesis of this work. This proves the hypothesis expressed in Conjectures 4.9, 4.10 to be wrong along with the similarity computation method. Another procedure to identify user similarity and allow comparison is consequently to be found and analysed: this this consists in the projection method, verified by means of the second test.

This forward step involves the evaluation of the label similarity procedure and, therefore, the svd-based algorithm for user similarity identification. Given the user model in shape of a label-indexed sequence of transformed feature subspaces, label-to-label similarity is performed by projecting one subspace on another at a time, regardless to which user the label pertains to. This leads to the generation of an inter-label similarity matrix $\mathrm{USM}_{u,v}$ for $u, v \in \mathcal{U}$. From this set of matrices it is immediate to elaborate the *user similarity matrix* containing the values of the user-to-user similarity computed according to the user similarity function defined in Conjecture 4.8: $\mathrm{US} \coloneqq [\mathrm{usim}(u,v)],\ \forall u, v \in \mathcal{U}$. A test like the previous, involving the generation of a number P of phantom users for each of the real ones, is necessary to understand if this model is able to capture the human perception of the acoustic features. In order to perform this, $P = 100$ was chosen to be the number of running simulations: in this way, for each pair u, v of real users, including the case $u = v$, we will have a computed similarity value $\mathrm{US}_{u,v} = \mathrm{usim}(u,v)$ and P phantom similarity values $\mathrm{usim}(u, v_p)$. We assume the model to be able to distinguish between real users and phantoms if the simulation is biased, meaning that the computed similarity value is an outlier

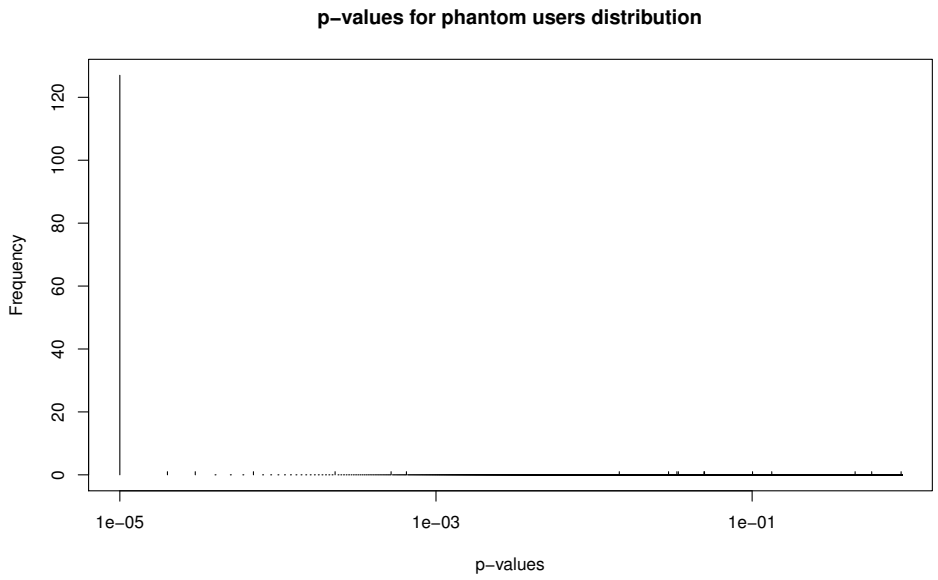

Figure 5.1: ***Hypothesis testing*** *– For each pair of users, a t-test is performed. This tries to reject the null hypothesis of having the mean of the phantom user similarities equal to the real user similarity. The histogram shows the frequency of p-values for 144 possible pairs of users: the hypothesis is almost always rejected also with significance* $\alpha = 0.01$*, thus we can conclude that the phantom users are far from the real ones according to the algorithm.*

with respect to the distribution of the phantom similarities. This was performed by means of a t-test, checking the null hypothesis that the phantom similarity distribution mean corresponds to the computed similarity values. This can be assessed only after checking for normality of the simulated data, assumption which is satisfied according to the results of the Shapiro [61] test that was performed. As it can be seen in Figure 5.1, the test rejects the null hypothesis in most of the cases, thus proving that the computed user similarity can never be assumed to be the average of the phantom similarities. The model manages to distinguish a real human behaviour in labeling from randomly generated tags, because the distribution of the phantom data is never compatible with the algorithmic result. It is noticeable that now the testing method is valid: even if different feature spaces are involved, they are different monotonic mappings of the same variables, thus keeping the orientation of relevant correlations in the space.

A last objective test has been performed by exploiting directly the provided user classification: this is important in order to assess the predictive capability of the model. The evaluation consists in the application of a method called *cross-validation*. Cross-validation [62, 63, 64] is a model validation technique for assessing how the results of a statistical analysis will

generalize to an independent data set. It is mainly used in settings where the goal is prediction in order to understand how the model will perform in practice. In a prediction problem, a model is given a dataset of known data on which training is run, called *training set*, and a dataset of unknown data against which the model is tested, called *validation or testing set*. One round of cross-validation involves partitioning the whole available dataset into complementary subsets, performing the analysis on the training set and validating the analysis on the testing set. In order to reduce variability, multiple rounds of cross-validation are performed using different partitions, as described in Section 4.4. In our case, the outcome of this process consists in the test set of songs to be labeled in an individual way, according to the different users' classes. The output will be a descending ranking for the most relevant classes the song could belong to. This is computed by maximizing an index which represents the similarity between the transformed song data model and any label model. In particular, data points belonging to a song are transformed according to the user-label covariance kernel map; then, the projection method is used to assess the actual similarity, supposing the provided label actually to be the correct one. This was performed for all of the users and labels in order to rank the possible label choices prior to the assignment to the song. As mentioned in Section 4.2, it should possible for a user to assign more than a single personal label to any song, both because musical genres might easily overlap and different kind of classifications are possible within one's own consumer habits. The conclusion is to deliver a recommendation which is not choosing solely the first element in the ranking of labels, rather it selects a set of possible labels which exceed a dynamic threshold in their similarity with the song itself. Furthermore, the threshold has been chosen to be the 10% of the maximum similarity index in the ranking, this in order to make it compelling with the self-awareness of the algorithm confidence on the results, thus testing also the latter.
A first assessment of the predictive capability of the method is shown in fig. 5.2. Here, it is possible to see, for each user, the amount x of labels which should be predicted in a top-x recommendation ranking in order to reach a certain accuracy. This plot considers 100 predictive simulations; the results of those are presented in the shape of ECDFs and a further information about the average matching label position is given. This assessment was fundamental for the decision of considering label prediction in a broader sense, but, as long as all of the labels are taken into consideration, this does not face the description of the capability of the algorithm to recognise a trustable predictive result. The definition of *trustable* lies in the 10% reference threshold, since it helps in creaming off the labels for which the prediction seems fuzzy. Within this interval, at least the top-ranking label is always present, while the dimension of the chosen label set underlines the safety or dubiousness of the prediction. Figure 5.3 shows in a boxplot the accuracy in prediction for each user, providing a result that spans across the different simulations.

It is possible to identify at first sight which users are correctly modeled and which model instead need a deeper training. Indeed, a relevant part of the labels were modeled on the basis of a meager quantity of songs. This of course impoverish the precision of the analysis and makes the predictions to be fickle. Moreover, the average index of the correct label in the ranking is shown in red above every user's results. If compared with what can be seen in fig. 5.2, it comes easily apparent how the introduction of the threshold allows to improve the predictions without affecting the accuracy. Figure 5.4 goes forward in this analysis, showing how many of the labels are needed proportionally to the amount the user provided (in red). The main evidence from this consists, of course, in the significant negative correlation between the overall number of available labels and the selected ones. This means that a fixed amount of labels are sufficient to predict the correct one with accuracy. We can thus infere from the figures that a top-3 recommendation ranking would be sufficient in most of the cases to hit the correct label.

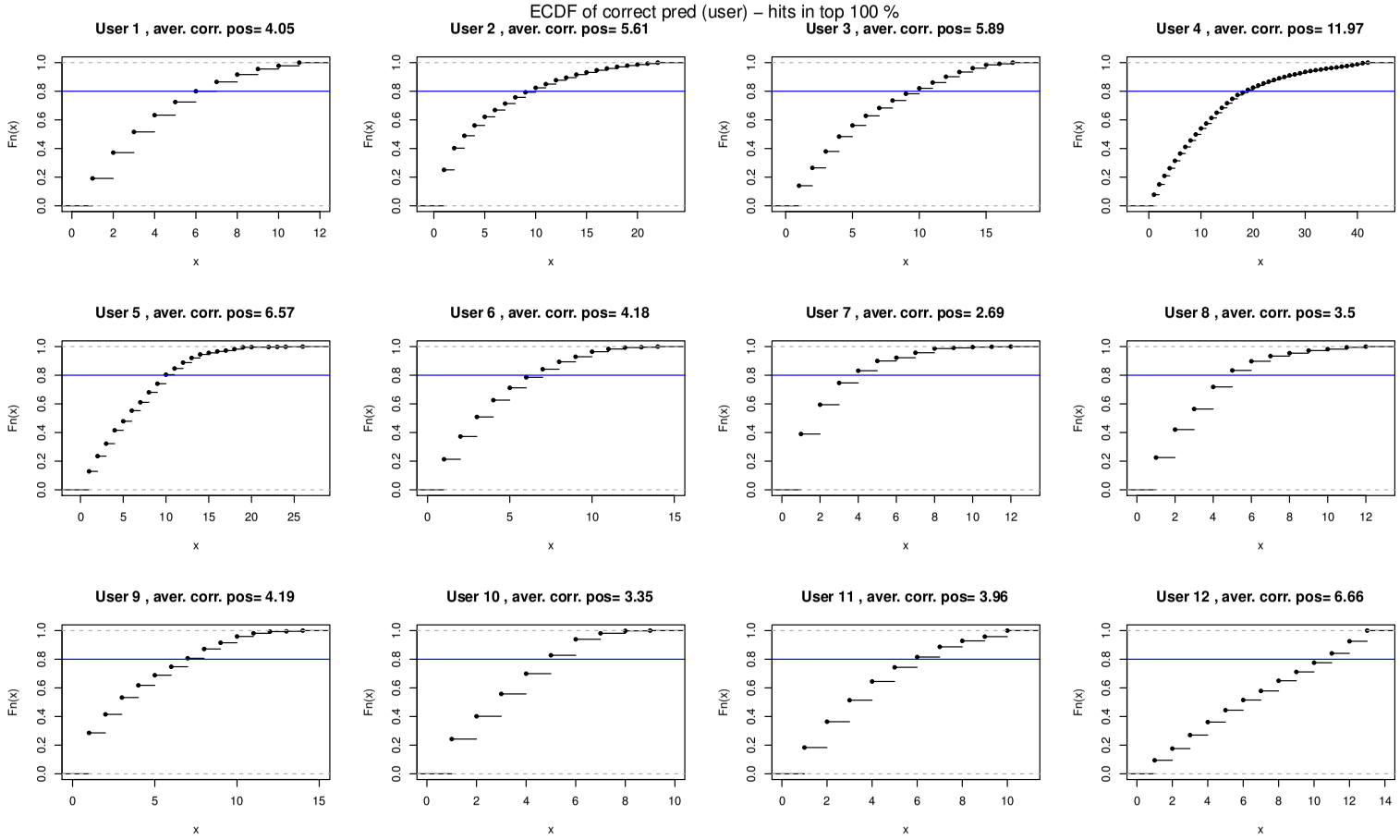

Figure 5.2: ***Cross-validation testing*** *– For each user, 100 different simulations of the model are runned by splitting the songs into training and testing sets in different ways. The training test is chosen randomly to represent 80% of the labeled data in each case, the remaining testing data are used for prediction. Each plot shows, for a user, the cumulative distribution of the correct prediction label in the whole ranking.*

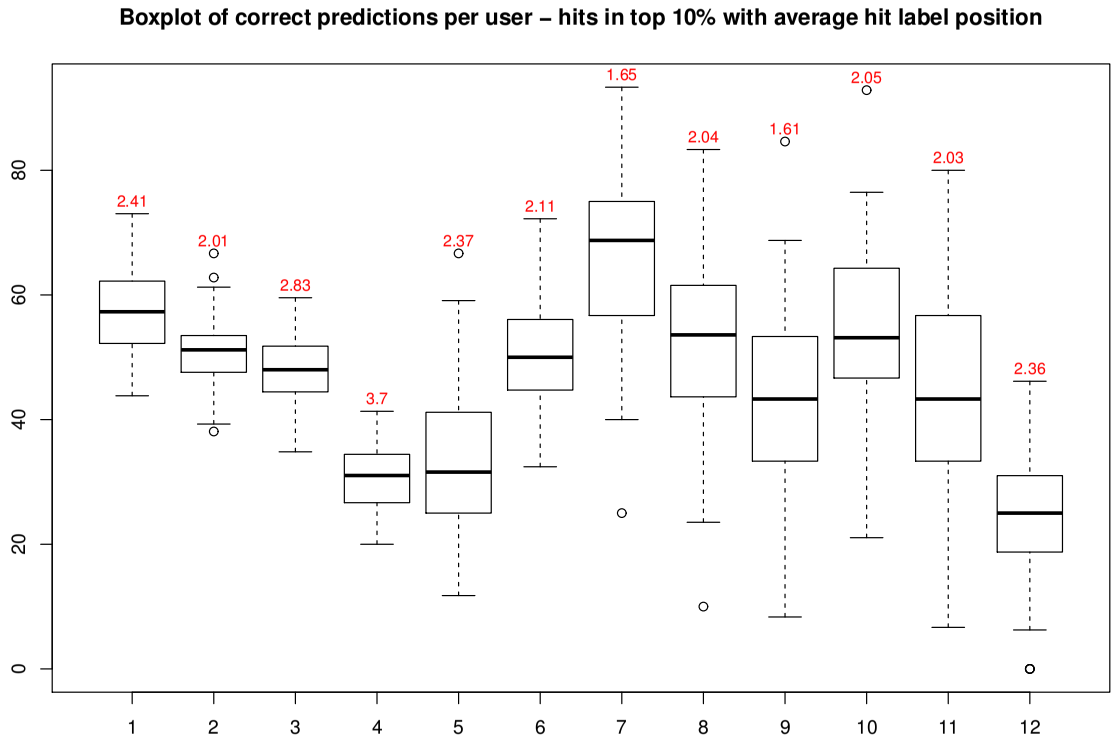

Figure 5.3: ***Cross-validation testing (2)*** *– Given the same simulation as in the previous figure, the boxplot shows, for each user, the correct predictions rate in the testing set when the hit-or-miss threshold is set to* 10% *of the maximum label score. The red number represents instead the average ranking position for a correctly predicted label. A significant decrease in this with respect to the previous figure can be noticed.*

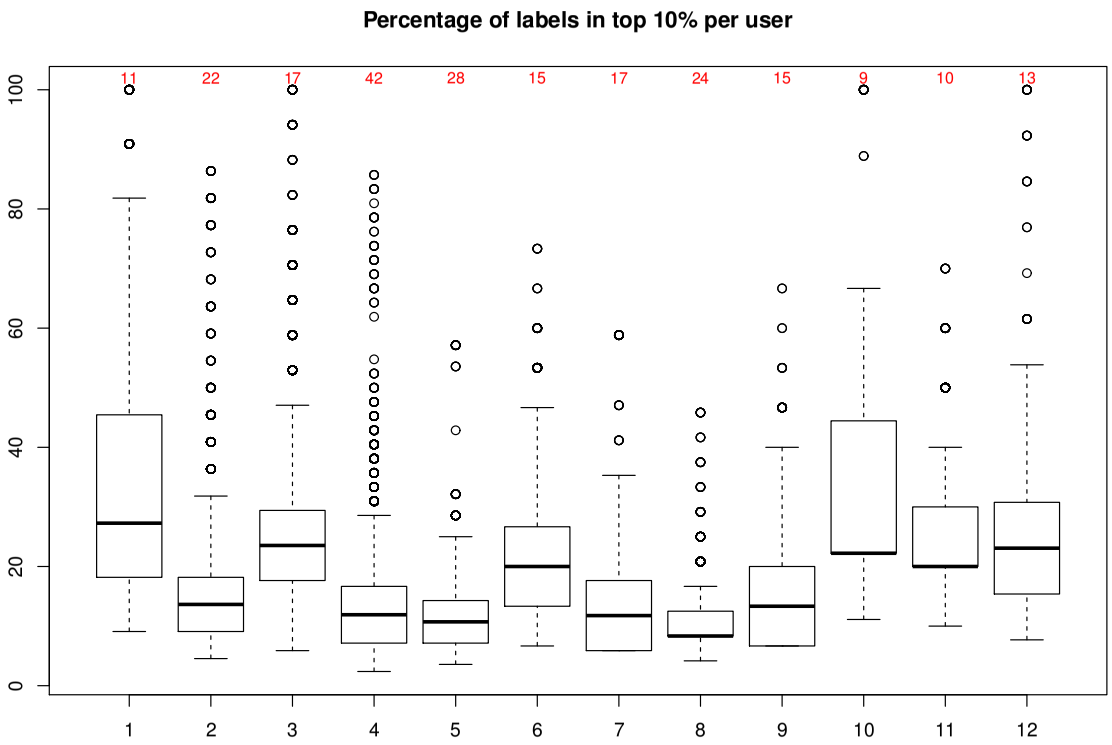

Figure 5.4: ***Cross-validation testing (3)*** *– For each of the above simulations, a different number of labels is contained in the interval of* 10% *of the maximum value. This boxplot collects the amount of kept labels for each user in each simulation. In order to allow for comparison between users, not to be affected by the different amount of total individual labels, the values are expressed in percentage with respect to the previous value (in red).*

5.2 Subjective evaluation

The subjective evaluation represents the most important feedback for the algorithm, since it allows performance measuring based on actual judgements by users. These must inherently be the same people providing the data for training, since the algorithm is aimed to personalization. In particular, the test consists in the same operations performed in the objective evaluation of predictions, but considering a new set of songs, which the users evaluated separately. Results of this first stage, fully compliant with the objective evaluations, are shown in figs. 5.5 and 5.6. Like the previous step, for each user the predicted labels are compared to the ones the user provided, but in this case people are required to provide also a value of agreement with the automatic annotation of each song. This value is chosen accordingly to a Likert scale, which ranges from 1 (complete disagreement with the assigned labels) to 7 (complete agreement). A further step in the evaluation consists in the analysis of the predictive error: if the prediction is wrong, the correct label is not among the ones owing maximum scores. The relative difference among the maximum score and the one of the exact label can be used not only as an overall measure of correctness, but also to check for correlation with the level of user agreement, measured with the Likert score. This is supposed to be present and negative, since the bigger the error, the worst the prediction will be. Furthermore, another metric which is investigated is related to indecision situations, meaning when the predicted scores in a top-x labels recommendation ranking are low and considerably similar to one another: the decision becomes an harder task, because many labels in this case are almost equally assignable to the query song. In order to deal with this, the correlation between the same Likert and the maximum predicted score is considered as a measure for indecision: this will allow to understand if some link is present among the indecision metric and the overall predictive capability.

The results of the described analysis follow: the correlation between the relative prediction error score and the Likert subjective evaluation is relevant and negative as expected, meaning that the prediction comes worst when the prediction error is bigger, thus the incertainty in the predicted scores already allows the system to understand that the following prediction can be wrong. The same holds for the description of the user agreement with the prediction, which is in positive correlation with the absolute value of the label similarity score, proving that also that value could a-priori identify the credibility of the predicted label. The last considered measures concern these observations in a more systematic way: two tests were run in order to check if the populations of the correctly identified labels and the wrong ones are different in terms of either the prediction scores or the minimum predictive delta. These analysis do not show significance to state they have different means, probably due to the small dimension of the song sample.

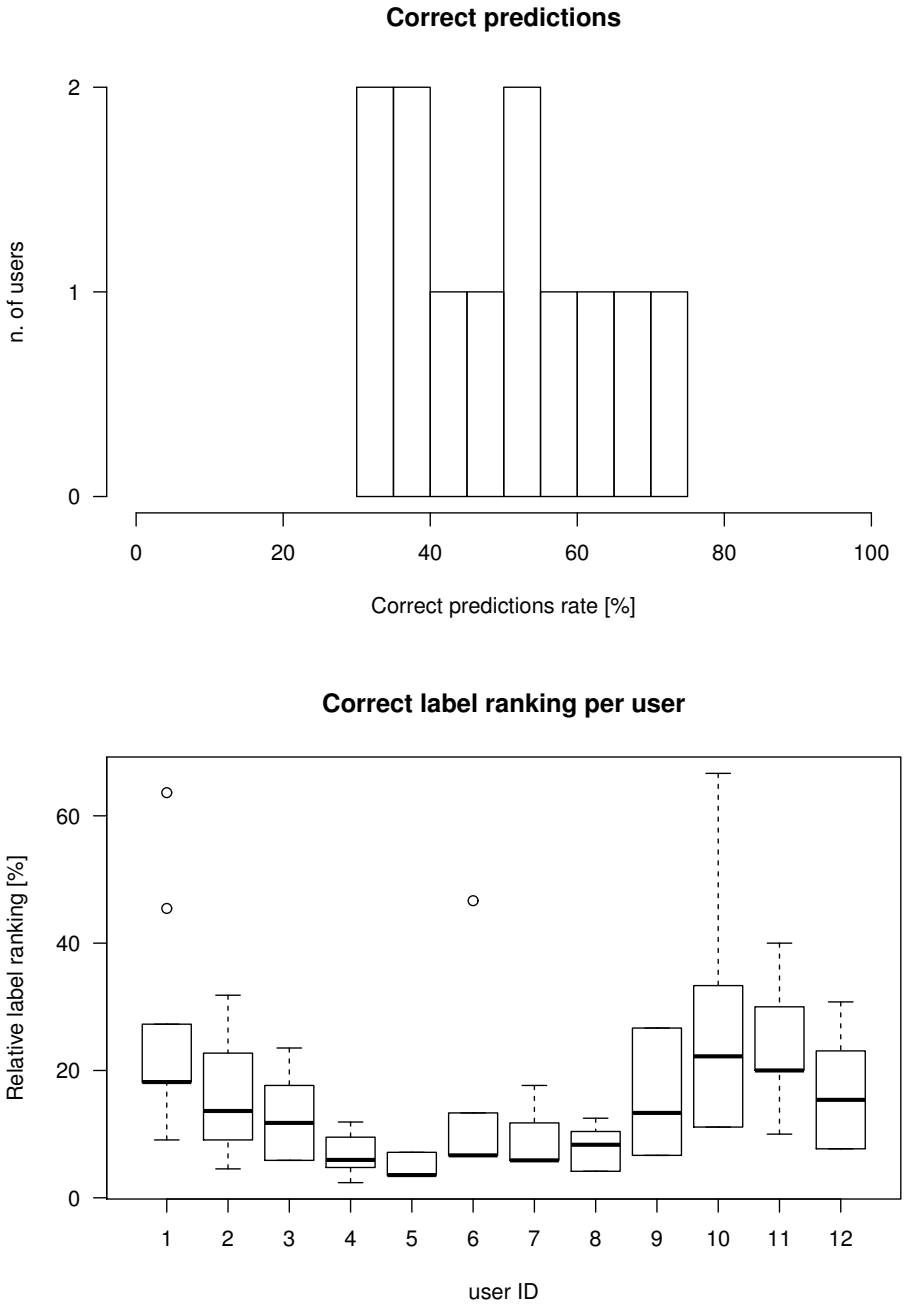

Figure 5.5: ***Correct predictions rates distribution and density of top-score labels*** *– The figure shows an histogram and a boxplot which have the same function of the ones shown for the objective evaluation of the algorithm. It is possible to observe that the values of the histogram are compliant with the mean values of the boxplot in figure 5.3, and the same holds for the boxplot related to the one in figure 5.4.*

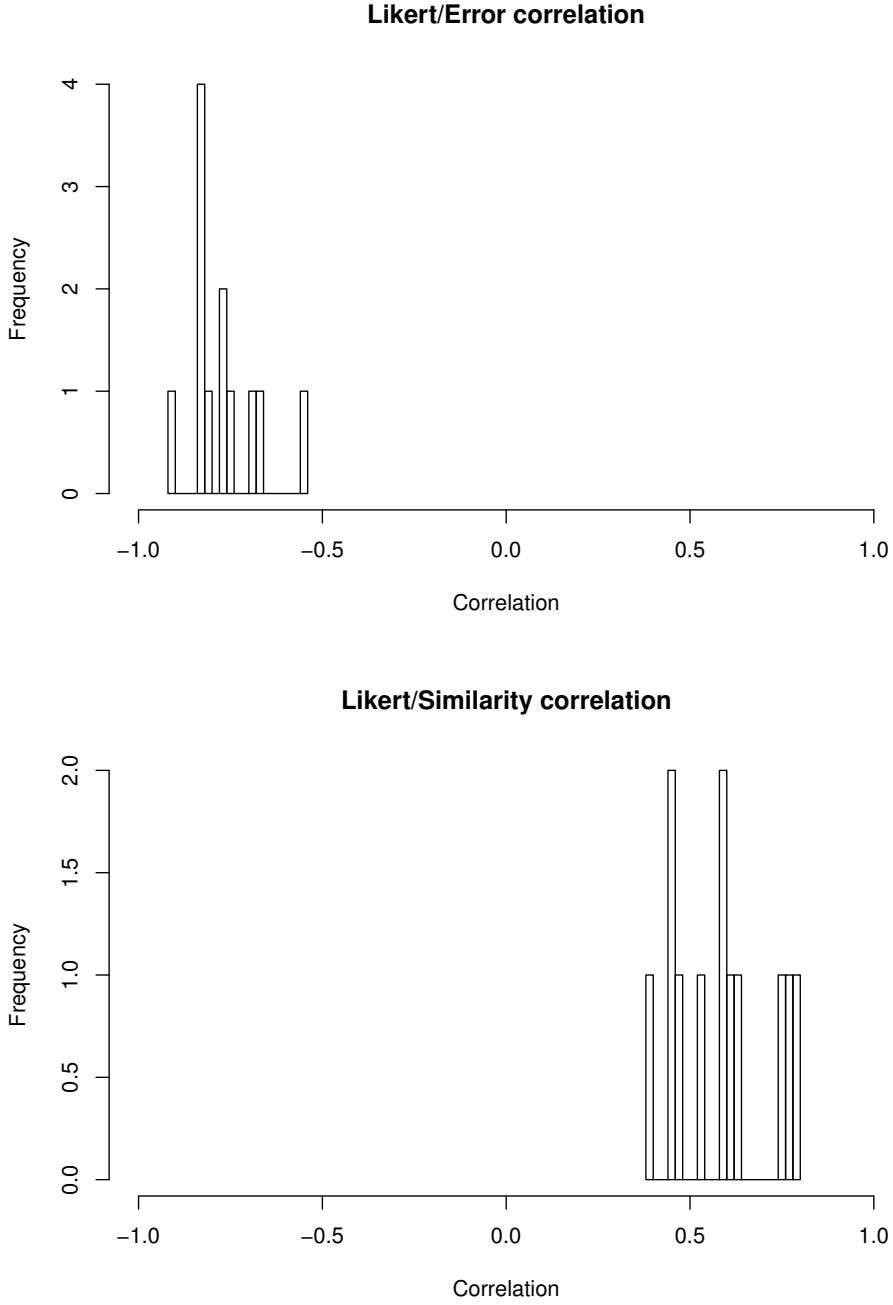

Figure 5.6: ***Correlation with subjective evaluation*** *– This histograms show the distribution of the correlations between the Likert-scaled indvidual evaluation of the predictions and two different parameters for performance self-evaluation: the first shows a significant negative link with the relative prediction error, meaning that correctness in predictions actually follows the users' agreement; the second instead positively relates the users' feedback with the score of the correct label, showing that, if the system assigns a greater score to a label, this will result in a more satisfactory prediction.*

Chapter 6

Conclusions and next steps

This thesis provided a new personalized approach to music similarity based on individual semantic. The work started with extracting suitable features from a database of audio tracks which was previously labeled by users. The next step consisted in modeling each individual label by considering it as a class of audio events, designed according to the acoustic features of its songs. The method had the advantage of taking into account the subjectivities of each user while assessing the similarity between songs thus defining a personalized classifier, which could be applied for recommending purposes either in a content- or collaborative-based environment.

The model was based on exploiting the probability distribution of acoustic data within each of the labels: since most algorithms for data analysis are based on Euclidean spaces, which is not always the case within musical features data, a first feature space mapping was necessary in order to make uniform the linear space of observations. This entailed a metric learning procedure, aimed to represent the actual structure of data in a more suitable way for processing. The procedure consisted in exploiting the cumulative probability density function of each acoustic feature within a label. The shape of a class in the data space was thus modeled, while the data were trasformed to univariate gaussians in order to explorate correlations in the Euclidean space. Those data were analyzed with the application of principal components analysis, which allowed for data reduction, and the label models obtained in the shape of low-dimensional vector spaces were compared by projection. The numeric results of projections were used as similarity metrics when comparing labels first and users in a successive step.
Of course, any song new to the model could be modeled as well as a set of acoustic events with given feature data. In this way, it could be joint with any of the labels with a similarity index. This score was maximized in order to provide for a classification: this has been the entry point for the recommendation step.

The obtained metric was inherently user- and label- specific, thus it could

be exploited to provide for similarity measurements between either individual labels or users, as well as it allowed to provide an indication about the pertinence of any song to the label. This was necessary in order to have a practical application for the model, which was identified in providing a new statistical engine for the outbreaking field of recommender systems for digital marketing.

The evaluation of this work included two different aspects, linked to both the objective predictive capability of the algorithm and the subjective satisfaction of the final user when the automatic classification is performed. In particular, we allowed each song to be labelled by more tags because of the fuzziness of the classification process. A suitable amount of labels was quantified in the top-scoring 10% of the user's labels, varying from 2 to 4 for the subjects of our tests. The performance of this operation was comparable with the state-of-the-art for non-personalized recommenders. Regarding the subjective tests, the users were provided with new songs along with their user-specific predicted classification and asked to express their satisfaction within a Likert scale. The result was compliant in efficiency with the previous objective evaluation. Moreover the users' feedback showed to be significantly correlated with some self evaluation of the confidence in prediction.

This proves the effectiveness of the algorithm, which it is worth to model music similarity according to the single user. In particular the model is able to provide good predictions even with a small amount of labeled songs. This can be a hint for possible applications towards reducing the impact of two well-known issues in recommending, known as *cold-start* and *long-tail* problem. Moreover, we proved the use of non-linear models to be necessary to successfully model the acoustic measurements: there is not a trivial linear relation among the features we chose to model songs and the user perception of music similarity.

6.1 Future works

The assessments on the proposed method have been performed by using a well known academic database of songs, called CAL500. This audio track collection was designed to contain different songs in terms of genre, age and popularity. Nevertheless, we proved users to have deeply subjective opinions about similarity among heterogeneous songs, as a common feedback on tests was the distribution of songs being non homogeneous towards genre. It would thus be interesting to consider the behavior of the algorithm with other user-specific songs collections (playlists), which are supposed to show this coherence towards more personalized characteristics. To this end, we would repeat our tests using user-generated listening databases. We could determine up to what extent our method is able to catch the subjectivities of the users on very specific music contexts. This could moreover improve the

analysis in solving the issue of having too few song representatives within the user-specific classes.

The previous assessment could be the starting point for the implementation of an effective recommender system based on user-generated playlists. Another possible development in this direction is the integration of user-related metadata for linking user similarities with their geographical or cultural data, in particular focusing on factors, such as music attitude and musicological background, that could influence the similarity perception of people. This could be useful in identifying correlations between the tastes of a group of users and their characterization: the generation of a *listening persona* based on its cultural or demographic metadata could importantly contribute to further smoothing the mentioned cold-start problem. The conclusion of this evaluation is the necessity of a broader and less characterized user database.

Some development could finally involve the modeling techniques: a deeper analysis may be necessary towards improving some methodological choices. In particular, in the gaussianization process, the empirical cumulative density of data could be changed by considering a smoother estimate (like expectation-maximization of gaussian mixtures). The same gaussian target distribution may be discussed with the objective of finding another probability kernel, whereas an analysis has already been performed on the similarity normalization constants: studies like [56] suggest different estimates for singular values which could be implemented while looking for improvements. Nevertheless, the present work is meant in its originality to set up an explorative study in statistics for personalized music recommending, introducing the importance of unpredictable nonlinear variations in the algebraic steps towards building a working implementation.

Appendices

Appendix A

Codes

Mahalanobis R code for Figure 3.10

```
### Principal Component Analysis

# data generation
library(mvtnorm)
mu  <-  c(1,2)
sig <-  cbind(c(1,1), c(1,4))
n   <-  100

X <- rmvnorm(n, mu, sig)

# data plotting
plot(X, asp=1,main = "Mahalanobis distance", xlab="First dimension",
     ylab = "Second dimension")

# plotting the average
points(colMeans(X)[1], colMeans(X)[2], col='red', pch=16)

# plotting the projections over the axis and computing variance
abline(h=colMeans(X)[2], lty=2)
points(X[,1], rep(colMeans(X)[2], n), col='blue')
var(X[,1])

abline(v=colMeans(X)[1], lty=2)
points(rep(colMeans(X)[1], n), X[,2], col='blue')
var(X[,2])

# plotting the ellipse
library(car)
M <- colMeans(X)
S <- cov(X)
ellipse(M, S, 1, add=T)

# computing eigenvalues and eigenvectors
eigen(S)
```

```
x <- seq(min(X), max(X), length = 100)
lines(x, M[2]+eigen(S)$vectors[2,1]/eigen(S)$vectors[1,1]*(x-M[1]),
      col='black', lty=2, lwd = 2)
lines(x, M[2]+eigen(S)$vectors[2,2]/eigen(S)$vectors[1,2]*(x-M[1]),
      col='black', lty=2, lwd = 2)
```

Segmentation Matlab code for *segmentation* process 4.3

```
tic
orig = 'C:\Users\pansi\Desktop\TESI\ongoing\ISPG20';
dest = 'C:\Users\pansi\Desktop\TESI\Script Python\data_segmented2';
mkdir(dest)
ext_out = '.wav';
num_seg = input('Please insert the number of the desired ...
    ... segments per song:');
seg_len = input('Please insert the duration of a segment [s]:');
filetype = {'/*.wav'; '/*.mp3'};
for type = 1:2
    ext_inp = filetype{type};
    filelist = dir([orig,ext_inp]);
    names = [];
    foldlist = dir(dest);
    foldernum_0 = str2num(foldlist(end).name);
    if isempty(foldernum_0) foldernum_0=0; end
    for i=1:length(filelist)
        foldernum = i+foldernum_0;
        if foldernum<10
            foldername = ['00' num2str(foldernum)];
        elseif foldernum<100
            foldername = [num2str(0) num2str(foldernum)];
        else foldername = num2str(foldernum);
        end
        mkdir(dest,foldername);
        filepath = [orig '\' filelist(i).name];
        filename = filelist(i).name;
        [X,fs] = audioread(filepath);
        seg_qnt = fs*seg_len;
        for j=1:num_seg
            init_pos = randi([1 length(X)-seg_qnt-1]);
            segment = X(init_pos:init_pos+seg_qnt,:);
            numlab = num2str(j-1);
            if j<11 numlab=['0' numlab]; end
            outname = [dest '\' foldername '\'
                filelist(i).name(1:end-4) '_' numlab ext_out];
            audiowrite(outname,segment,fs);
        end
        disp(['Elaborated file ' num2str(i) ' of '
            num2str(length(filelist)) ' after ' num2str(toc)])
    end
end
```

Bibliography

[1] Luca Poddigue (supervised by A. Sarti and M. Zanoni), *A personalized content-based music similarity function*, master graduation thesis at Politecnico di Milano, Milan, 2014.

[2] Maria Stella Tavella (supervised by A. Sarti and M. Zanoni), *Audio features compensation based on coding bitrate*, master graduation thesis at Politecnico di Milano, Milan, 2017.

[3] Michele Buccoli, Alessandro Gallo, Massimiliano Zanoni, Augusto Sarti, Stefano Tubaro, *A Dimensional Contextual Semantic Model for music description and retrieval*, in Proceedings of ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 673-677.

[4] Bruno Di Giorgi, Massimiliano Zanoni, Augusto Sarti, Stefano Tubaro, *Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony*, in NDS '13 - Proceedings of the 8th International Workshop on Multidimensional Systems (2013), pp. 1-6.

[5] Isinkayea F.O., Folajimib Y.O., Ojokoh B.A., Recommendation systems: Principles, methods and evaluation, in Egyptian Informatics Journal, 16:3 (2015), 261-273.

[6] Friedman N., Geiger D., Goldszmidt M., *Bayesian network classifiers*, Machine Learning, 29 (2–3) (1997), pp. 131-163.

[7] Duda R.O., Hart P.E., Stork D.G., *Pattern classification*, John Wiley & Sons (2012)

[8] Bishop C.M., *Pattern recognition and machine learning*, Vol. 4, no. 4. Springer, New York; 2006.

[9] Bobadilla J., Ortega F., Hernando A., Gutiérrez A., *Recommender systems survey*, in Knowledge-Based Systems, 46 (2013), pp. 109-132.

[10] Melville P, Mooney-Raymond J, Nagarajan R., *Content-boosted collaborative filtering for improved recommendation*, in Proceedings of the eigh-

teenth national conference on artificial intelligence (AAAI), Edmonton, 2002. p. 187–192.

[11] Adomavicius G., Tuzhilin A., *Towards the next generation of recommender system. A survey of the state-of-the-art and possible extensions*, in IEEE Transactions on Knowledge Data Engineering, 17 (6) (2005), pp. 734-749.

[12] John Blacking, *How musical is man?*, University of Washington Press, Washington, 1974.

[13] Eric J. Humphrey, Juan P. Bello, Yann LeCun, *Moving beyond feature design: Deep architectures and automatic feature learning in music informatics*, 13th ISMIR Conference, 2012.

[14] Corey Kereliuk, Bob L. Sturm, Jan Larsen, *Deep Learning and Music Adversaries*, in IEEE Transactions on Multimedia, 17:11 (2015), pp. 2059-2071

[15] Pengjing Zhang, Xiaoqing Zheng, Wenqiang Zhang, Siyan Li, Sheng Qian, Wenqi He, Shangtong Zhang, Ziyuan Wang, *A Deep Neural Network for Modeling Music*, in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (2015), pp. 379-386

[16] Jean-Julien Aucouturier, Emmanuel Bigand, *Mel Cepstrum & Ann Ova: the difficult dialog between MIR and music cognition*, 13th ISMIR Conference, 2012.

[17] Marcin Maleszka, Bernadetta Mianowska, Ngoc Thanh Nguyen, *A method for collaborative recommendation using knowledge integration tools and hierarchical structure of user profiles*, in Knowledge-Based Systems, 47 (2013), pp. 1-13.

[18] Xiangyu Zhao, Zhendong Niu, Wei Chen, *Interest before liking: Two-step recommendation approaches*, in Knowledge-Based Systems, 48 (2013), pp. 46-56.

[19] Daniel Wolff, Tillman Weyde, *Adapting metrics for music similarity using comparative ratings*, 12th ISMIR Conference, 2011.

[20] Douglas Turnbull, Luke Barrington, David Torres, Gert Lanckriet, *Towards musical query-by-semantic description using the CAL500 data set*, SIGIR '07, 2007.

[21] Brian McFee, Luke Barrington, Gert Lanckriet, *Learning similarity from collaborative filters*, 11th ISMIR Conference, 2010.

[22] Markus Schedl, Arthur Flexer, *Putting the user in the center of music information retrieval*, 13th ISMIR Conference, 2012.

[23] Arthur Flexer, *On inter-rater agreement in audio music similarity*, 15th ISMIR Conference, 2014.

[24] Arthur Flexer, Thomas Grill, *The problem of limited inter-rater agreement in modelling music similarity*, in Journal of New Music Research, 45:3 (2016), pp. 239-251.

[25] Markus Schedl, Arthur Flexer, Julian Urbano, *The neglected user in music information retrieval research*, in Journal of Intelligent Information Systems, 41:3 (2013), pp. 523-529.

[26] Dmitry Bogdanov, Perfecto Herrera, *How much metadata do we need in music recommendation? A subjective evaluation using preference sets*, 12th ISMIR Conference, 2011.

[27] Dmitry Bogdanov, *From music similarity to music recommendation: Computational approaches based on audio features and metadata*, PhD thesis, Universitat Pompeu Fabra, Barcelona, 2013.

[28] Nicola Orio, Roberto Piva, *Combining timbric and rhythmic features for semantic music tagging*, 13th ISMIR Conference, 2012.

[29] Gabriel Vegliensoni, Ichiro Fujinaga, *Automatic music recommendation systems: do demographic, profiling and contextual features improve their performance?*, 17th ISMIR Conference, 2016.

[30] Rion Snow, Brendan O'Connor, Daniel Jurafsky, Andrew Y. Ng, *Cheap and fast — but is it good? Evaluating non-expert annotations for natural language tasks*, in Proc. Empirical Methods in NLP (2008), pp. 254–263.

[31] Audrey Laplante, *Improving music recommender systems: what can we learn from research on music tastes?*, 15th ISMIR Conference, 2014.

[32] Chris Sanden, Chad R. Befus, John Z. Zhang, *A perceptual study on music segmentation and genre classification*, in Journal of New Music Research, 41:3 (2012), pp. 277-293.

[33] Jean-Julien Aucouturier, François Pachet, *Representing Musical Genre: A State of the Art*, in Journal of New Music Research, 32:1 (2003), pp. 83-93.

[34] Francesco Ricci, Lior Rokach, Bracha Shapira, *Recommender Systems Handbook*, Springer Science+Business Media, New York, 2011.

[35] Y-Lan Boureau, Francis Bach, Yann LeCun, Jean Ponce, *Learning Mid-Level Features For Recognition*, in Proceedings of the International Conference on Computer Vision and Pattern Recognition (2010), pp. 2559–2566.

[36] Lie Lu, Dan Liu, Hong-Jiang Zhang. *Automatic mood detection and tracking of music audio signals*, in IEEE Transactions on audio, speech and language processing, 14(1):5–18, 2006.

[37] Brian CJ Moore, Brian R Glasberg, *Modeling binaural loudness*, in The Journal of the Acoustical Society of America, 121(3):1604–1612, 2007.

[38] Tae Hong Park, *Salient feature extraction of musical instrument signals*, PhD thesis, Dartmouth College Hanover, New Hampshire, 2000.

[39] Geoffroy Peeters, *A large set of audio features for sound description*, in IRCAM, 2003.

[40] Chris Duxbury, Juan Pablo Bello, Mike Davies, Mark Sandler et al., *Complex domain onset detection for musical signals*, in Proc. Digital Audio Effects Workshop (DAFx), 1, pp. 6–9, 2003.

[41] Kristoffer Jensen, *Timbre models of musical sounds*, PhD thesis, Department of Computer Science, University of Copenhagen, 1999.

[42] Jochen Krimphoff, Stephen McAdams, Suzanne Winsberg, *Caractérisation du timbre des sons complexes - II - analyses acoustiques et quantification psychophysique*, in Le Journal de Physique IV, 4(C5):C5–625, 1994.

[43] Lawrence R Rabiner and Biing-Hwang Juang, *Fundamentals of speech recognition*, Vol. 14. PTR Prentice Hall Englewood Cliffs, 1993.

[44] Bee Suan Ong *Towards automatic music structural analysis: identifying characteristic within-song excerpts in popular music*, PhD thesis, Citeseer, 2005.

[45] Brian CJ Moore, Brian R Glasberg, Thomas Baer, *A model for the prediction of thresholds, loudness, and partial loudness*, in Journal of the Audio Engineering Society, 45(4):224–240, 1997.

[46] Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, Lian-Hong Cai, *Music type classification by spectral contrast feature*, in Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on, volume 1, pages 113–116. IEEE, 2002.

[47] Stephen Mcadams, *Perspectives on the contribution of timbre to musical structure*, in Computer Music Journal, 23(3):85–102, 1999.

[48] HF Pollard, EV Jansson, *A tristimulus method for the specification of musical timbre*, in Acta Acustica united with Acustica, 51(3):162–171, 1982.

[49] Eberhard Zwicker, *Subdivision of the audible frequency range into critical bands*, in The Journal of the Acoustical Society of America, 33(2), pp. 248-248, 1961.

[50] http://jamiebullock.github.io/LibXtract/documentation/

[51] https://github.com/bbc/bbc-vamp-plugins/blob/master/src/

[52] Meinard Müller, *Fundamental of Music Processing*, Springer, 2015

[53] John Saunders, *Real-time discrimination of broadcast speech/music*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, pp. 993-996 (1996)

[54] Eric Scheirer, Malcolm Slaney, *Construction and evaluation of a robust multifeature speech/music discriminator*, in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2, pp. 1331-1334 (1997)

[55] Tao Li, Mitsunori Ogihara, George Tzanetakis, *Music data mining*, Chapman & Hall/CRC, Boca Raton, 2012.

[56] Liqun Qi, *Some simple estimates for singular values of a matrix*, in Linear Algebra and its Applications, 56 (1984), pp. 105-119.

[57] Scott Shaobing Chen, Ramesh A. Gopinath, *Gaussianization*, in Advances in neural information processing systems, pp. 423-429, (2001).

[58] Murray Rosenblatt, *Remarks on Some Nonparametric Estimates of a Density Function*, in The Annals of Mathematical Statistics, 27 (3), pp. 832-837. (1956)

[59] Emanuel Parzen, *On Estimation of a Probability Density Function and Mode*, in The Annals of Mathematical Statistics, 33 (3), pp. 1065-1076 (1962).

[60] Wand, M.P, Jones, M.C., *Kernel Smoothing*, Chapman&Hall/CRC, London, 1995.

[61] Sam S. Shapiro, Martin Bradbury Wilk, *An analysis of variance test for normality (complete samples)*, Biometrika, 52, (3:4), pp. 591-611 (1965).

[62] Seymour Geisser, *Predictive inference*, Chapman&Hall/CRC, New York, 1993.

[63] Ron Kohavi, *A study of cross-validation and bootstrap for accuracy estimation and model selection*, in Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 2:12 (1995), pp. 1137-1143.

[64] Pierre A. Devijver, Josef Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall, London, 1982

Acknowledgements

Colgo l'occasione per ringraziare tutti coloro che mi sono stati vicini e hanno creduto nel mio percorso e in questo lavoro. In primis la mia famiglia, senza la quale, semplicemente, non sarei. Ringrazio inoltre chi mi ha formato prima e durante i miei anni universitari, come studente e come persona. A questi si aggiungono e di questi fanno parte alcune persone speciali, il cui supporto è stato fondamentale pure in questo lavoro: Paolo, Julien, Ahmed, Giulio e Valeria; tutti a loro modo compagni di fatiche, ma soprattutto di vita.