VERIFYING TAG ANNOTATION AND PERFORMING GENRE CLASSIFICATION IN MUSIC DATA VIA ASSOCIATION ANALYSIS

TOM ARJANNIKOV
Bachelor of Science, University of Lethbridge, 2012

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfillment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

VERIFYING TAG ANNOTATION AND PERFORMING GENRE CLASSIFICATION IN MUSIC DATA VIA ASSOCIATION ANALYSIS

TOM ARJANNIKOV

Date of Defense: April 30, 2014

Dr. John Zhang Supervisor	Associate Professor	Ph.D.
Dr. Yllias Chali Thesis Examination Committee Member	Professor	Ph.D.
Dr. Gongbing Shan Thesis Examination Committee Member	Professor	Ph.D.
Dr. Howard Cheng Chair, Thesis Examination Committee	Associate Professor	Ph.D.

Dedicated to the warm and loving memory of

Tatiana Arjannikova,

my late mother.

Abstract

Music Information Retrieval aims to automate the access to large-volume music data, including browsing, retrieval, storage, etc. The work presented in this thesis tackles two non-trivial problems in the field.

First problem deals with music tags, which provide descriptive and rich information about a music piece, including its genre, artist, emotion, instrument, etc. At present, tag annotation is largely a manual process, which often results in tags that are subjective, ambiguous, and error-prone. We propose a novel approach to verify the quality of tag annotation in a music dataset through association analysis.

Second, we employ association analysis to predict music genres based on features extracted directly from music. We build an association-based classifier, which finds inherent associations between music features and genres.

We demonstrate the effectiveness of our approaches through a series of simulations and experiments using various benchmark music datasets.

Acknowledgments

I gratefully acknowledge the support and guidance of my thesis advisor Dr. John Zhang. I am very thankful for his patience throughout my studies and the countless hours that he spent reviewing my writing. Many thanks go out to my graduate studies committee members Dr. Yllias Chali and Dr. Gongbing Shan for their thoughtful remarks and encouragement during my studies.

I would like to extend my sincere gratitude to all of the faculty and staff of the Department of Mathematics and Computer Science for their help and encouragement during my undergraduate and graduate studies. If I could single anyone out in addition to the abovementioned, it would be Dr. Akbary-Majdabadno for he is very generous with his time, and I am thankful for his thoughtful assistance on many occasions.

I would also like to extend my thanks to the School of Graduate Studies for giving me this opportunity and for assisting me with the many logistics and formalities throughout my studies.

Last, but not least, I would like to thank all of my friends and family for their continuing support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Music is such a big part of human culture that we listen to it on a daily basis to alter our moods, express emotions, and convey other kinds of information. People around the world create more new music on any given day than a single person can consume in that same day; just think how many free jam sessions might be happening at this very instant around the world. Furthermore, no one can possibly listen to all of the music available today, even if one spends her or his entire lifetime listening to it non-stop.

Consider the following: an average lifespan of a Canadian resident is 80 years [1], which is approximately 42 million minutes; the Gracenote[1] database contains information about more than 130 million tracks. Even if each track was just one minute long, no one on this planet has lived long enough to be able to listen to all of the music available. Albeit Gracenote maintains one of the largest databases that contain information about music, it does not have a record of every single music piece produced.

In addition, vast amounts of music exist only on vinyl, magnetic tapes, scores written on paper, and other analog media, waiting to become digitalized. However, rapid advances in technology, like data storage and the Internet, and more recently, the mobile computing, in many ways help facilitate the process of converting analog data such as books, film, and music into digital format. Moreover, the growing demand for

[1] www.gracenote.com.

diverse and rich listening experiences drives these advances.

It is easy to see how this might happen. For example, many people enjoy listening to music while engaging in various sport activities. This demand for portable access to music encourages some companies to produce tiny devices with large amount of storage. Meanwhile, other companies develop better compression and streaming algorithms, and yet another group of companies take those old favourites from the vinyl and put them into the ears of many eager listeners.

Since anyone can consume only a small fraction of all the music in the world, then there needs to be a way to find the right recording for any occasion. Hence, we need to facilitate the effective navigation of our colossal digital collections and speedy retrieval of relevant music. This is just one of many fundamental problems that concerns researchers in the area of *Music Information Retrieval* (MIR) [18].

1.1 Music Information Retrieval

MIR is a fast growing multidisciplinary area that attracts researchers from a wide range of disciplines, such as library science, information retrieval, audio engineering, musicology, cognitive science, psychology, and computer science. The latter drives MIR by advancing technologies that not only increase the speed and capacity of our devices, but also their intelligence.

Because of the improvements in storage and compression, we are able to keep a large amount of music on very small devices, such as the Micro SD cards. Thus, large collections of music become portable. Moreover, advances in multimedia and networking enable us to listen to streaming audio on the go and wirelessly, while advances in signal processing contribute to many tasks, such as source separation, instrument/voice recognition, song identification, and even automatic music genre and mood classification.

The following are some examples of how MIR-related research finds its way into

our daily lives. There are smartphone programs, such as *Shazam*[2] or *SoundHound*[3] that analyse the music being listened to and recognize much of it. Consequently, their users gain access to information about music to which they are listening. Websites like Last.fm[4], Pandora[5], Slacker[6] and Spotify[7] are able to offer customized listening experiences to their users through algorithms that recommend songs and automatically create playlists based on factors such as the listener's mood, genre preference, and listening history.

As outlined by Li *et al.* [29], all of the MIR tasks can be divided into the following eight categories.

Data management is a task of organizing, storing and accessing music. As the amount of music grows daily, so does the need to be able to access it easily. The challenge here is how to apply various data indexing techniques to manage music.

Association mining is the process of detecting correlations between different acoustic features, between music and other documents, and between music features and other aspects of music. In this thesis, we examine the first and the last of the three. In Section 2.3.5, we provide additional information about association mining in MIR.

Sequence mining involves looking at music through time by examining elements like rhythm, chord sequence, and music structure. Compared to other categories, there are fewer works published in sequence mining, mostly focused on music transcription.

Classification, one of the more important tasks in MIR, involves splitting a music collection into various categories. Some of the popular tasks include artist/genre classification, singer/instrument identification and mood/emotion detection. In this thesis, we focus on genre classification.

Clustering is also concerned with splitting a music collection into different groups.

[2]www.shazam.com.
[3]www.soundhound.com.
[4]www.last.fm.
[5]www.pandora.com.
[6]www.slacker.com.
[7]www.spotify.com.

However, it is different from classification because there are no predefined categories or class labels. In music, clustering mainly aims to find groups based on music similarities.

Similarity search is a task that embraces a broader notion than classification or clustering. Here, the researchers examine the music feature space and use it to compute the distance between different music pieces. They focus on searching for music based on an approximate description of one or two other music pieces. Thus, someone browsing a music collection may be able to find pieces similar to the ones that he or she heard before.

Music summarization is similar to text summarization [29], it focuses on generating the most informative and compact description of music. Until now, most music summarization was generated manually. However, with the rapid increase in volume of many digital collections, automated approaches are currently in high demand.

Data visualization consists of two sub-tasks: visualizing individual music pieces and visualizing entire music collections. The first one is concerned with visualizing metadata and audio content of a single music document. This is useful in cases where a consumer attempts to gauge the content of a music piece, similar to skimming a book before buying it. The second visualisation task is also oriented around a consumer, where one might want to discover new music by navigating through an entire repository in some intuitive way.

1.2 Contribution

All of the terms and concepts mentioned in this section are further explained in the following chapters.

Although much effort goes towards automating music tag annotation and propagation, our work is among the initial efforts to verify the quality of these. We propose using association analysis to verify the quality of tags within a given music reposi-

tory [5]. Not only can we test whether the new tags entering an existing repository are consistent with the ones already in that repository, we can also use our approach to verify the quality of the tag annotation process by which music tags are obtained.

We also propose to use association analysis for music genre classification [6]. It is worth noting that we are not the first to classify music via association rules. Neubrath *et al.* [37] apply association mining to discover relation between two folk music ontologies: genre and region. We discuss it in further detail in Section 2.3.5. Our approach is different, because we employ a data mining technique called association rule mining [21]. It is a supervised learning approach that consists of two components. First, we build a statistical model of the given data, and then we apply this model to classify new data.

The work outlined in Chapter 4 explores the feasibility of applying association rule mining to music genre classification. During our work, we find that there are some inherent associations between audio characteristics and human assigned music genre labels. Thus, association analysis is applicable in MIR for classification purposes. Furthermore, it is comparable to existing classification methods. However, we believe there are many ways to improve our proposed method, which we leave to future work, as discussed in Section 5.2.

Our work contributes specifically to the following MIR tasks: data management, association mining, classification, and similarity search. More generally, any individual user or a corporation maintaining a database of descriptive tags about music could indirectly benefit from the findings in our work. Additional beneficiaries are the consumers of products and services surrounding such repositories; for example, anyone who uses online radio and recommendation services, such as Last.fm and Pandora.

1.3 Outline

This thesis is organized as following. First, we unfold all of the underlying concepts and ideas in Chapter 2. Starting with a brief overview of music data in Section 2.1.1, following by an in-depth look at *tags* and the *tag annotation* process in Section 2.2. Then, in Section 2.3, we examine in detail the central idea of this thesis, *association analysis*. This is followed by an overview of classification in Section 2.4 with the focus on music genres. In this chapter, we also expose previous works.

In Chapter 3, we propose an approach to use association analysis towards verifying music tag annotation. We first outline it in Section 3.2 and present the issues in Section 3.3 that prevent us from testing the approach directly. To overcome these issues, we set up a series of simulations and discuss their outcomes in Section 3.5.

In Chapter 4 we present a classifier that is based on rules, which we mine using association analysis. We start by explaining the how we set up the experiments in Section 4.2, which includes the particulars of the classifier and the goals that we wish to achieve. Then we show the results of our experiments in Section 4.3 and follow up with brief discussions in Section 4.4.

Finally, in Chapter 5 we draw some concluding remarks and expose future directions on using association analysis for both verification of music tag annotation and classification of music into genres.

Chapter 2

Background

In this chapter, we start with an overview of music data and the associated challenges in Section 2.1.1. We leave tag annotation-specific challenges for Section 2.2, where we examine the sources of music tags and current approaches to verifying them.

In Section 2.3, we describe the principal component of this thesis, association analysis. After introducing this data mining technique, we formulate the problem and illustrate it with an MIR related example. In the same section, we also present some previous works in MIR that use association analysis.

Then, in Section 2.4, we present the problem of classification in MIR. From all classification tasks in MIR, we focus on music genre classification. Thus, in this section, we outline the kinds of features that are used in predicting music genres. We also give a brief outline of discretization, a data mining technique, which we employ in order to make the features suitable for our approach. Then we present several classifiers that were successfully applied to music genre classification. We also outline the established approaches to evaluating classifier performance that we use in our work.

2.1 Music Data

2.1.1 Types of Music Data

There is a natural distinction between two types of music data: the actual sound and its *metadata*. Of course, the actual sound is not data; it is a unique event

occurring in time and space. Hence, when we talk about actual sound data, we refer to whatever is stored on various media such as vinyl records, magnetic tapes and now in digital formats. Metadata, then, is the information about this event that we store and use. There are many types of metadata, ranging from descriptions of sound to instructions on how to produce sound. For example, Johann Sebastian Bach would write a set of instructions on how to reproduce a certain sequence of sounds using a particular instrument. However, the sound that he produced from his own sheet music would be different depending on the instrument, the room and even his mood at the time. *Musical Instrument Digital Interface* (MIDI) devices use similar type of instructional metadata. MIDI carries event messages, which contain information about the sound such as pitch, tempo, instrument, effects, and clock signals. This information is very compact and easy to manipulate. It is stored in digital format and is used to reproduce the sound very accurately. However, it is limited in such a way that it cannot reproduce any sound, but only the sound generated by MIDI devices in the first place. If prescriptive metadata contains instructions on how to produce sound, then the descriptive type of metadata simply describes it. For example, we often categorize music into genres, such as *rock*, *blues* and *country*. We also often name songs and remember their original authors. This kind of information does nothing for reproduction of the sound, but it describes the different attributes of that sound. In music repositories, this type of metadata is commonly referred to as *music annotations*. These annotations are the focus of this thesis.

Most digital music libraries nowadays store musical annotations along with the audio and use it to help organize their collections. They often describe artists and music using terms like *upbeat*, *jazzy*, *happy*, and *danceable*. For example, the database maintained by Gracenote, mentioned in Chapter 1, has information about the content of audio compact discs. Gracenote provides its data to companies like Nullsoft[8] and

[8] www.nullsoft.com.

Apple[9] for use with their popular music players Winamp and iTunes, respectively. These databases include information such as music genre, artist, and release year, which is useful for organizing and browsing of music. Music annotations are stored in the format of textual tags, like "rock" and "jazz". These tags make it easier for a user to browse and even search the contents of a large music repository without spending time to listen to each song individually.

2.1.2 Challenges

Most of actual music data is copyrighted. This means that researchers and developers must each purchase a copy of every song that they want to work with. Furthermore, if they are to collaborate or compare their approaches, they must ensure that the music collections used in their experiments are identical between them. This is very unsustainable, and only slows the research and development of anything related to actual music data.

In addition to financial burdens, there are also computational challenges in MIR, such as dealing with the sequential nature of music. Music happens in time, or through time, and we cannot fully describe it without using temporal information. There are various techniques, with their own challenges, which deal with time. A standard approach in MIR is to break the music piece into short timeframes or windows and look for patterns across them; these windows can be overlapping, adjacent or sparse. For example, the Latin Music Dataset described in Section 4.2.2 includes features extracted from first, middle and last 30 seconds of each song; Silla [23] shows that using all three together achieves higher classification accuracy than using only features from any individual timeframe. The challenges of looking for patterns across time include, but are not limited to, determining the number of frames and determining the location of said frames. Furthermore, they include the development of algorithms, which model patterns and structures across time, which is an area in machine learning

[9] www.apple.com.

with many open questions [41].

Another challenge is the fact that music is a cultural phenomenon, which changes over time. Consider, for example, the case of building a schema for music genre classification, where each genre known to us is considered and placed in some hierarchy of genres. Over time, new genres will be introduced into the music world, which the established schema is obviously not able to handle. Thus, a desirable approach is one that is flexible enough to handle such changes.

The lack of data, which is of good quality and large size, poses another serious challenge to researchers in MIR. Nonetheless, the situation is constantly improving, especially with the recent release of the Million Song Dataset (*MSD*) in February 2011 [8]. Shortly after, in 2012, the Million Song Dataset Challenge [34] attracted further attention to the MSD. It was set up to help bridge the gap between real life data in the industry and individual researchers. The MSD Challenge also encouraged reproducible, open evaluation of algorithms and approaches in MIR. In addition, in 2012, Schindler *et al.* [44] released an addition to MSD called the Million Song Dataset Benchmarks (*MSDB*). It expands the number of content-based features and provides genre and style tags for supervised learning tasks. Despite their seemingly large size, both MSD and, by extension, MSDB datasets are still quite small when compared in size to the industry repositories with billions of tracks.

We discuss the issues specific to tag annotation in Section 2.2.1, where we also consider the sources of metadata, because these issues are closely related to their respective sources.

2.2 Tag Annotation

As mentioned in Section 2.1.1, music tags can help with organizing and browsing a music collection. This is mainly because they are very precise and compact. Because of how they are generated, tags carry the exact information needed for a given task.

Furthermore, they are very popular because they are not restricted by copyrights. Even though in some cases, such as Pandora's situation, access to metadata is very restricted, most music metadata is freely available for use. Moreover, many companies create *Advanced Programming Interfaces* to enable outside developers to use their metadata. Usually, this results in the appreciation of said metadata's value and popularity, as outside developers enable and enhance the said data and its use.

2.2.1 Music Tags and Related Issues

Sources of Music Tags

In his work, Pachet identifies three types of metadata creation: manual, cultural and acoustic [39].

The first type involves people ranging from experts to amateurs performing annotation manually by listening to individual tracks and tagging them accordingly. This process introduces subjectivity, ambiguity and errors into the metadata created, and thus, makes it difficult to maintain the metadata in large databases [38]. Web sites like Pandora pride themselves on the accuracy of their annotations because they employ expert musicologists to generate their annotation tags manually. Albeit this approach is proven to be financially expensive, experts rarely disagree and the resulting annotations are of the highest quality possible.

The second type of metadata, cultural, comes mostly from the application of various crowdsourcing methods or from the analysis information about users' behaviours. For example, the recommender system in use by iTunes[10] and Last.fm websites employs one of the prominent approaches in this category called *collaborative filtering*. This method involves examining which songs the users listen to and then, based on that information, predicting what preferences or tastes an individual user might have. One of the main issues with this and other crowdsourcing methods is the *cold start problem*. Simply put, if there is no user data to begin with, then no recommendations

[10] www.itunes.com.

can be made. This motivates the research of content-based approaches, which brings up the third type of metadata creation described by Pachet [39].

Acoustic is perhaps the most objective of the three types, as it mainly involves signal processing and statistical approaches. This process produces metadata information like the average number of beats per minute or a presence of a certain instrument in a given track. However, there is still a large gap between human perception of sound and that, which we can represent with computers. Hence, some acoustic metadata, such as instrument recognition, is still generated manually. This is because automatic methods are less precise than human efforts in many annotation tasks, particularly in genre recognition [32, 19] and instrument [17] recognition.

Larger music repositories with numerous contributors are especially sensitive to such issues as described above.

Types of Music Tags

During the annotation process, we can distinguish between two types of tags that an annotator may use.

One type of tags comes from a discrete set, which contains all possible situations for the corresponding characteristic it describes. For instance, a tag may describe whether an instrument is present or not, then each instrument would have its own location in the characteristics space and the corresponding tag would come from the set $\{present, not\_present\}$

The second type comes from a set only one element, that tag. Because it is hard to determine all possible tags for a given characteristic, especially when new tags are introduced over time. In the case of genres, for example, many researchers are moving away from a single-label genre assignment, since the styles of many compositions tend borrow from multiple genres at once. In this case, when an annotator assigns a tag $\{rock\}$ to some music, the same music may also display styles of $\{country\}$ or $\{blues\}$.

Hence, the only way to fit all possible genre assignments into one complete set of tags is to use a different tag for each possible combination of all genres. Not a desirable strategy, especially when new genre distinctions are regularly introduced into the music culture. Therefore, most practitioners choose to assign each genre tag from a set containing one element $\{present\}$. In this way, each tag comes from its own dimension in the characteristics space and does not interfere with others. In addition to allowing for more flexibility, this also creates difficulties. Some researchers study this as a multi-label problem, further discussed in Section 2.4.4.

Although association analysis, described in Section 2.3, naturally deals with multiple tags assigned to a single piece of music, our genre classification experiments described in Chapter 4 use single-labeled data, so exactly one genre tag per music piece. This is because the genre information in the data we use is represented by the first of the two types of tags presented in this section.

Missing Information

Annotators often do not use a predetermined set of tags during the annotation process. This creates a situation of uncontrolled vocabulary, which means that anyone can use any word to describe any piece of music. Because of this, the tags in a single music repository become diluted, since many tags share the same meaning. Uncontrolled vocabulary also leads to multi-label situations. More importantly, a problem of missing information creeps in.

Suppose that one annotator listens to a song and determines that it should have a tag "happy" and moves onto the next song. This does not mean that there is only one word in existence, which describes that song. Usually others come along and add more tags. However, if the song has no violins in it, rarely would someone tag it "no violins". This is the case with Last.fm data, where the annotators will tag a presence of a certain characteristic, but not necessarily its absence.

Hence, this type of data becomes unusable for conventional machine learning techniques, and requires different approaches. This problem of missing information affects many MIR tasks, especially classification. To deal with it, many approaches are proposed, such as automatic tag prediction and propagation; a common one is to verify the tag annotations manually.

2.2.2 Verifying Tag Annotation

Having bad initial tags is detrimental to most tasks in MIR; hence, their verification is very important. Whether it is the annotation process or the tags themselves that are verified in a given repository, many efforts are focused towards ensuring good quality of metadata.

One of the most conventional methods is to ensure that during the manual annotation process, more than one person annotates each music piece and a certain level of agreement is reached before the tag is applied. For example, Kim *et al.* [24] propose MoodSwings, a game that collects music mood labels from players; it enforces some agreement between annotators through game design. The game pits two players against each other, and their goal is to agree on the mood of several music clips with very limited feedback from each other. Gathered in this way, the mood annotations with high agreement are considered to be of high quality and become useful for various tasks, such as automatic playlist generation and automatic prediction and propagation of mood tags. This game also deals with another big issue outlined in Section 2.2.1, the cost of hiring people to annotate music manually. By making it fun, Kim *et al.* encourage people to volunteer their time to tagging music [24].

Laurier *et al.* [26] use a different approach, as they study the agreement between a large community and experts in mood classification task. They create a semantic mood space from tag annotations found on last.fm website, which serves a community of over 30 million users who are very active at annotating music with mood tags.

Then, they compare the mood represented in this space with existing representations from psychological studies, and show that there is agreement between experts and the large community. Their work was inspired by Sordo *et al.* [48], who study the agreement between genre annotation created by the last.fm community and experts from MP3.com and find that there is some agreement, but not always, as in the case of the *rock* genre. This stresses the importance of verifying annotations, especially when large communities annotate a music repository.

Good quality tags are important in all areas of MIR, particularly classification, where researchers are working to bridge the gap between music signal and human perception of it. Since the main way of gaining insight into human perception and understanding is from testimonials and music annotations created by people serve as such, the presence of erroneous or ambiguous tags is contrary to these efforts. Obtaining good quality tags is detrimental to both parts of this thesis - association analysis and genre classification. Furthermore, the problem of missing information also causes issues in music genre classification. Association analysis, however, is not impeded very much by this problem; instead, it reflects it, which makes it suitable to verify tag annotations in a music repository, and even the tag annotation process itself.

2.3 Association Analysis

Association analysis [2] attempts to discover the inherent relationships among data objects in an application domain. Such relationships are represented as association rules. An example of such application domain is the shopping basket analysis in supermarkets, where one tries to discover the relationships among the items purchased by customers. For instance, the association rule $\{milk, eggs\} \rightarrow \{bread\}$ implies that, if *milk* and *eggs* are bought together by a customer, then *bread* is likely to be bought as well, i.e., they have some inherent statistical relationships [21].

We can use association analysis in MIR to examine music tags in a repository. Each music piece or track in the repository can have multiple tags associated with it. Each tag represents a specific characteristic or feature that either objectively or subjectively describes the track. Although many music tags are subjective, when grouped together, they contain some patterns. We can look for these patterns and examine them. The patterns that occur frequently can help reveal interesting information about the music in a repository. Additionally, we can derive rules from these patterns, which indicate associations between tags and possibly between different acoustic features in a repository. Although association analysis is very common to some areas, like market analysis [21], it is new to MIR and that makes it interesting to study.

2.3.1 Problem Formulation

We follow Ness *et al.* [36] in their formulation of the music tag annotation process as follows. Given a set $T = \{t_1, t_2, ..., t_n\}$, where each $t_i \in T$ is a tag, and a set $M = \{m_1, m_2, ..., m_r\}$, where each $m_j \in M$ corresponds to a music piece. Then each music piece m_j can be considered as an annotation vector $A = (a_1, a_2, \cdots, a_n)$, where $a_k > 0$ if tag t_i has been associated with the piece, and $a_k = 0$, otherwise. These a_k's, referred to as *semantic weights*, describe the strength of the semantic correspondence between a tag and the music piece. When mapped to a binary assignment of $\{0, 1\}$, the semantic weights can be interpreted as class labels, i.e., whether a tag is assigned to the music piece or not; in this thesis, we always assume binary assignment. Naturally, each music piece can be annotated by multiple tags [36, 42], which makes them appropriate for association analysis.

Before proceeding with association analysis, we first derive a transactional style dataset $D = \{d_1, d_2, \cdots, d_r\}$ from the set of music pieces M. There are two ways to form it based on the set of n transaction items, which is already represented as the set of tags $T = \{t_1, t_2, ..., t_n\}$.

One way to form D is by creating two items, t_{i0} and t_{i1} from every t_i. In this way, a transaction d_j will contain the item t_{i0} if $a_k = 0$ and t_{i1} otherwise. In case of missing values, the transaction d_j will contain neither item t_{i0} nor t_{i1}. If any t_i has a non-binary assignment, then we would create additional items t_{i2}, t_{i3}, etc, so long as one discrete set of items/labels describes all possibilities of the characteristic corresponding to t_i.

The second way to form D is by adding an item t_i to d_j if the corresponding $a_k = 1$. The case when $a_k = 0$ is handled the same way as missing values, no item is added to d_j. This is the most common approach and it requires that all tags be of the second type, described in Section 2.2.1.

When we formulate our problem as described above, the music annotation set M becomes a transactional set D suitable for association mining.

2.3.2 An Example

There are many illustrations of association analysis in the data mining literature like the shopping basket analysis example [21]. Since MIR is at the center of this thesis, let us consider an appropriate hypothetical situation. Suppose we have a small music repository and tags from four distinct categories: mood/emotion, activity, instrument, and genre. For the sake of simplicity, we have only six music pieces with only one tag per category. We split the repository into two genre categories and annotate the music pieces accordingly, as illustrated in Table 2.1. This example will help demonstrate how to derive the *frequent itemsets* and *association rules* in the following two sections, titled accordingly.

2.3.3 Frequent Itemsets

We consider the itemsets that appear in many transactions to be frequent. The threshold that separates the frequent items from the infrequent depends on the task and is determined by whomever is overlooking the data mining process. This threshold

Music Piece	Mood/Emotion H	Activity D	Instrument V	Style/Genre P/C
#1	Happy	Dancing	Violin	Pop_Country
#2	Happy	Dancing		Pop_Country
#3		Dancing		Pop_Country
#4			Violin	Classical
#5		Dancing	Violin	Classical
#6	Happy	Dancing	Violin	Classical

Table 2.1: Example music repository.

is measured in terms of *minimum support*, which is the percentage of transactions that contain the item in question. Given a set of items I:

$$support(I) = \frac{\sum_{j=1}^{r} sign(I)}{r}, \text{ where } sign(I) = \begin{cases} 1 & I \subseteq d_j \\ 0 & otherwise \end{cases} \tag{2.1}$$

Figure 2.1 illustrates the transactional form of the example dataset in Table 2.1 and all of the combinations of five items with their corresponding support in the example dataset.

The set of frequent itemsets contains a set of *closed itemsets* [21], which are those whose support is greater than the support of each of its immediate supersets. The set of closed itemsets, in turn, contain a set of *maximal itemsets* [21]. A frequent itemset is considered maximal when none of its immediate supersets is also frequent. This relationship can be generalized as:

$$maximal\ itemsets \subseteq closed\ itemsets \subseteq frequent\ itemsets \subseteq all\ itemsets \tag{2.2}$$

In Figure 2.1, we illustrate all itemsets, even if they do not appear in the datasets. To visualize the relationship between different types of itemsets we draw their lattice in Figure 2.2. In bold are the items considered frequent when the minimum support threshold is set to 33%. From these we can derive a set of closed itemsets $\{\{D\}, \{V\},$

Dataset	
#	Transaction
1	H D V P
2	H D P
3	D P
4	V C
5	D V C
6	H D V C

Step 1		
Item	Count	Support
H	3	50.0%
D	5	83.3%
V	4	66.7%
P	3	50.0%
C	3	50.0%

Step 2		
Item	Count	Support
HD	3	50.0%
HV	2	33.3%
HP	2	33.3%
HC	1	16.7%
DV	3	50.0%
DP	3	50.0%
DC	2	33.3%
VP	1	16.7%
VC	3	50.0%
PC	0	0.0%

Step 3		
Item	Count	Support
HDV	2	33.3%
HDP	2	33.3%
HDC	1	16.7%
HVP	1	16.7%
HVC	1	16.7%
HPC	0	0.0%
DVP	1	16.7%
DVC	2	33.3%
DPC	0	0.0%
VPC	0	0.0%

Step 4		
Item	Count	Support
HDVP	1	16.7%
HDVC	1	16.7%
HVPC	0	0.0%
HDPC	0	0.0%
DVPC	0	0.0%

Figure 2.1: Transactional format of the example music repository and all possible combinations of its five items with their respective counts and supports.

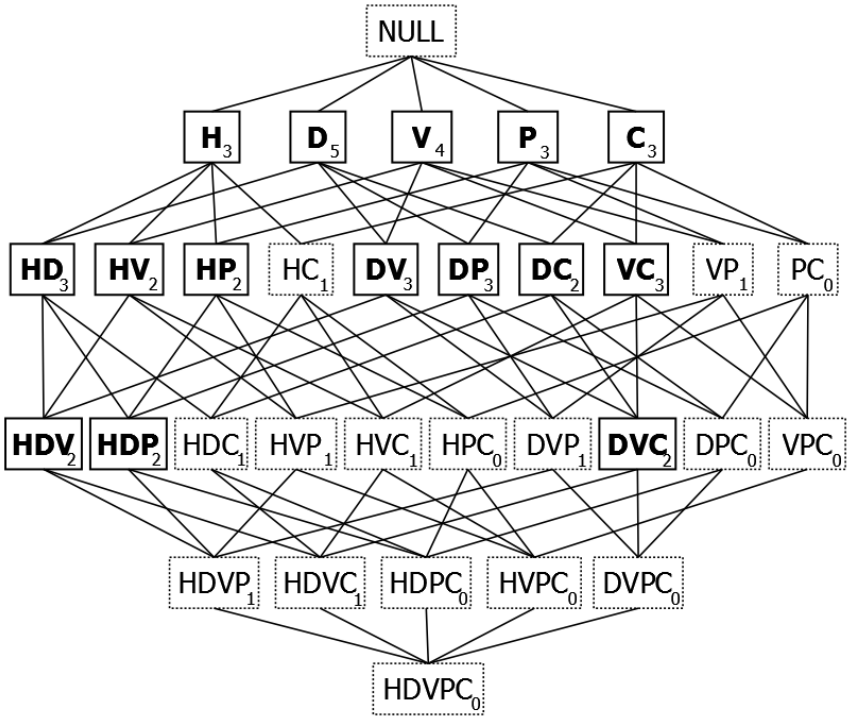

Figure 2.2: Tag patterns from the example represented as lattice, with item count listed in bottom right corner. The *frequent itemsets* with minimum support of 33% are in bold.

$\{HD\}$, $\{DV\}$, $\{DP\}$, $\{VC\}$, $\{HDV\}$, $\{HDP\}$, $\{DVC\}\}$ and from these a set of maximal itemsets $\{\{HDV\}, \{HDP\}, \{DVC\}\}$.

2.3.4 Association Rules

An association rule is of the form $A \rightarrow B$, where A and B are non-empty frequent itemsets, $A \subseteq T$ and $B \subseteq T$ and $A \bigcap B = \phi$. We call A the *antecedent* and B its *consequent*. Rule $A \rightarrow B$ holds for D with some support as described above, and *confidence*, which is the percentage of transactions containing A that also contain B.

$$confidence(A \rightarrow B) = \frac{support(A \cup B)}{support(A)} \tag{2.3}$$

In our example in Section 2.3.2, the rule $D \rightarrow P$ has confidence of 60% and the rule $P \rightarrow D$ has confidence of 100%. It should be noted that this is not a logical

conditional but a statistical relation rule.

Evaluating Association Rules

Support and confidence are the most popular methods of measuring significance of an association rule. However, there are other measures, which can be useful in measuring not only significance, but also usefulness or interestingness of a rule. Some of the more popular ones are lift, conviction, coverage, leverage, all-confidence, and collective strength [21]. In this thesis, we use support and confidence exclusively, and leave exploring other evaluation measures to future work.

How interesting a given association rule is, obviously, cannot be measured by simply applying some formulae. Which measure(s) to use and how to apply them properly always depends on the task. The evaluation measures above are a good point for where to start the exploration of data. However, the data itself usually dictates what are the best tools suited for it. This is the case with our approach, to be outlined in Chapter 3, where we use both support and confidence together to set the exploration boundaries, and then propose a new set of evaluation measures, detailed in Section 3.2.4, which are designed specifically for our task.

2.3.5 Association Analysis in MIR

Association analysis is very new to MIR probably because this research area emerged very recently. There are many data mining and machine learning methods, which still have not seen music data. However, there is a growing need to find associations between various objects in music data evident by several works outlined below.

Kuo *et al.* [25] propose a way to recommend music based on the emotion that it conveys. In their experiments, they use film music because, arguably, music is an important component of the emotional content in films. They first choose a set of representative emotions and use it as controlled vocabulary to annotate music

pieces. After extracting the relevant music features which represent melody, rhythm and tempo, they discover the relationship between these features and emotion by constructing a music affinity graph. Although no association rule mining is performed, they look for associations in their data that contain information perceived only by humans, namely emotion.

Xiao *et al.* [54] use a parameterized statistical model to look for associations between timbre and perceived tempo. First, they extract the timbre features directly from sound and manually annotate each feature set with tempo. Afterwards, they use the *Gaussian Mixture Model* (GMM) to model the relationship between the features and tempo. Then, they demonstrate how to use the derived model to aid in automatic tempo recognition. Although they do not use association rule mining, Xiao *et al.* [54] find the associations between timbre features and tempo.

Liao *et al.* [31] use a dual-wing harmonium model to discover association patterns between MTV video clips and the music that accompanies those clips. They do not mine association rules in a conventional way as described in previous sections. Instead, they discover association patterns first, by extracting a combination of features from audio and video, and then, by clustering the sample data in this feature space. Then, they use the discovered patterns to match video to music automatically. This method can be used to create custom sound tracks for personal video collections. They demonstrate that the results of their approach are comparable to those of a commercial software package, *muvee autoProducer*[11].

To our knowledge, there is only one work published prior to ours that uses association rule mining as part of analysing music via association analysis. Neubarth *et al.* [37] present a method of association rule mining with constraints. They first come up with a specific rule template in the form of $A \rightarrow B$, where both A and B contain only one element, and region implies genre and genre implies region. Then they look

[11] www.muvee.com.

for rules that match this template in a 1902 Basque folk music corpus and find some interesting rule sets. Neubarth *et al.* use support, confidence and two evaluation measures to determine the quality or interestingness of the rules mined.

2.4 Classification

Classification is the process of organizing ideas or objects into a hierarchy of classes. More specifically, when we refer to classification, we mean its application in data mining and machine learning, where we distinguish between different data objects and then organize them according to pre-defined categories. Classification is a supervised type of learning in contrast to unsupervised and semi-supervised machine learning approaches. In all three, we form a computational model from a set of observations or examples from the world, and then use this model to make predictions about the world. The difference between them is whether the examples are labeled or not.

In a supervised learning task [41], the given observations, or training data, are correctly identified and labeled prior to building the model. In this way, the learner knows which data is an example of one situation or another and learns from this information. For example, when a child is learning the names of colors, his or her parents would present an object and then say the correct word for the color of that object; this continues until the child consistently and correctly identifies the colors of new, never seen before objects. We can say that the child is a learner and the parent is a supervisor in this situation, and we can call it supervised learning.

Then, unsupervised learning [41] means that there is no supervisor, because the examples are not labeled. One of the most common examples of unsupervised learning is clustering, where the task is to divide a given set of objects into groups or categories that are not defined ahead of time.

The third type of machine learning, semi-supervised learning [41], is a combination of the first two. Here, the given data consists of a small portion that is labeled and

a large portion that is unlabeled. We can then build our computational model, for instance, first, by clustering the unlabeled data, and then use the labeled data to name the clusters.

2.4.1 Classification in MIR

Classification is one of the older and more important tasks in MIR. It can be subdivided into the following categories [29].

Audio classification is where the focus is on distinguishing between speech, music and environmental sound. This usually precedes other classification tasks, as we would separate the music from all other sounds before further analysing the music.

Genre classification, also known as music genre recognition, a category where researchers are concerned with categorizing music audio into different genre, and lately style groups. Tzanetakis and Cook [52] were among the first to work on this problem, where the task is to label an unknown piece of music with a correct genre name. They show that this is a difficult problem even for humans and report that college students achieve no more than 70% accuracy. Although, currently, there is a bias towards classifying western music, recently MIR researchers started working with Asian and Middle Eastern music.

Artist identification task is to recognize the name of the singer or the band who performed a given music piece. This is very important to other MIR tasks related to organizing and retrieving data in large music collections and has numerous possible applications such as copyright management and music recommendation [29].

Mood detection is another classification category related to music recommendation. Here, the focus is on predicting the emotional state of a human listening to some music piece [50, 22]. This task is rapidly gaining popularity, probably because music affects people's mood, which is one of the main uses of music.

Last, but not least, is the *instrument recognition* category in MIR classification.

This very difficult task is far from being solved, and it is closely related to source separation in signal processing [29].

2.4.2 Features in Music Classification

The usual approach to music classification starts with content-based feature extraction, which is done via signal processing. First, each music piece is divided into segments, which may overlap. Then, each segment is processed via an algorithm, which produces a set of values describing some characteristic of the sound.

Content-based features can be divided into several categories, but the main three are *harmony*, *timbre* and *rhythm* [29, 16]. *Harmony* describes the rate of vibrations within the sound, also known as *pitch*. It is similar to the term *wave frequency*, as defined in physics. *Timbre* describes a certain texture or tone quality of the sound; for instance, two saxophones may sound different from each other even while producing sound of the same pitch and loudness. The third category has to do with the temporal nature of music, *rhythm*, which works together with pitch to form melody.

In addition to content-based features, as of recent, researchers use music tags for classification [55]. For example, many tags such as *dark*, *happy*, *energetic*, *mellow*, etc., describe the aspects of music that are perceivable only by humans. Thus, these tags carry meaningful description of music, which has proven to be helpful in classifying other tags, such as genre or instrument. The situation also works in reverse, where genre or instrument tags can be useful in classifying music by into mood or emotion categories. In our work, we focus on genre classification.

2.4.3 Discretization

Many classification models are compatible with both continuous and discrete data. However, association analysis requires discrete data, while most content-based features are not. Thus, we need to convert continuous values produced by signal processing into some finite number of labels. This process is called *discretization* and is usually

done algorithmically [21].

The idea here is to divide the continuous range of a given attribute into intervals, also referred to as *bins*. Each bin is then labeled, and the value of the attribute in each music piece is then replaced with the appropriate label, corresponding to the range that the actual value into which it falls. There are many ways of deriving these bins, such as equal-width or equal-frequency binning, and histogram analysis. These fall under the unsupervised discretization, because they do not use class information in order to determine the bin ranges. There are also supervised discretization approaches, such as entropy-based discretization, where the class labels information is used to determine which ranges are best to use; this may help improve the classification accuracy. Another example of supervised discretization is ChiMerge, which is based on χ^2, where the two adjacent bins are merged if they have a very similar distribution of classes [21].

There are many other supervised and unsupervised approaches to discretization in the literature. Currently, discretization is an open problem in data mining, where improvements are made on a regular basis. In our work, we use unsupervised attribute discretization implemented in the Weka software [20] and leave exploring other discretization methods to future work. The reason for using it is to avoid any possible bias based on class labels. This allows us to examine our proposed classifier in its purest form.

2.4.4 Multi-label Classification

Mixing musical styles to achieve unique artistic flavour has become quite popular in recent years, because of this, music pieces tend to fall into several genre categories simultaneously. For example, many Bob Dyllan[12] and Johnny Cash[13] songs belong to two genres, rock and country. Different genre labels reflect this. For example, the

[12] www.bobdylan.com.
[13] www.johnnycash.com.

jazz rap is a subgenre of *hip-hop*, which includes elements of *jazz*; similarly, *folk jazz* combines the elements of both *folk* and *jazz*.

Many approaches have been proposed to deal with this situation, following are two examples. Wang *et al.* [53] combine content-based features, music tag information and music style correlation to achieve a new multi-labeling classification model called *Hypergraph integrated Support Vector Machine*. Lukashevich *et al.* [33] study multi-labeling using the GMM classifier to examine assigning multiple labels to each music piece; moreover, recognizing the temporal structure of music, they look at labeling different segments of each music piece. Then, they propose an approach, which allows problem space decomposition from multi-label into multiple single-label classification problems in two dimensions.

The classifier that we propose in Chapter 4 is expected to behave as a single-label one because we remove the intersections between every genre pair. However, association analysis is suitable for multi-label situations and our classifier is capable of assigning multiple genre labels to a single music piece. We believe that with some minor modifications, our proposed classifier can handle the multi-label situation in music genres.

2.4.5 Music Genre Classification

Many machine-learning models are successfully applied to music classification. In this section, we describe some supervised learning approaches that appear in MIR.

Bayesian Model (BM) is a probabilistic model, which makes predictions based on prior knowledge, usually referred to as evidence. The usual approach is to build a directed acyclic graph, where each node is a variable corresponding to an attribute, and each edge is a conditional dependency between variables. Any two nodes not connected with an edge are conditionally independent of each other. Then, a probabilistic function is applied at each node, to determine its value based on its parent nodes and

their corresponding probabilities to affect the node in question. This model has been shown to work well for classification, where features accompanied by class labels are used as evidence to train the model, and then unlabeled data can be classified based on this model. DeCoro *et al.* [13] use BM to aid in hierarchical classification of music by aggregating the results of multiple independent classifiers and, thus, perform error correction and improve overall classification accuracy.

Support Vector Machine (SVM) classifiers map an input vector of attributes into a high dimensional space and then derive a hyperplane, which separates the examples of different classes in this space. Moreover, the SVM chooses the hyperplane in such a way that it maximizes the distance between classes. This distance is computed using a function commonly referred to as kernel. A new data instance is then mapped into the same space and assigned a label based on which side of this hyperplane it falls. Recent examples of using SVM for music genre classification include an investigation of Meng and Shawe-Taylor [35], where they explore different kernels used in a support vector classifier. Li and Sleep [28] implement normalized information distance into kernel distance for SVM and demonstrate classification accuracy comparable to others.

Decision Tree (DT) algorithms are an effective way to classify music into genres among other classification tasks. This approach selects attributes best suited for branching based on information gain. At each node of the tree a single attribute is selected along which new branches are formed, one per value of the attribute. After building the tree from training examples, a new, unlabeled, music piece can be classified by simply following the branches of the tree. The decision of which branch to follow is based on the value of the attribute corresponding to the current node. Thus, a set of rules can be derived, indicating which attribute values lead to which class. In addition to speed, one of the main advantages of using DT is the interpretability and intuitive understanding of the resulting classification rules by humans. Recently, Anglade *et al.* [4] used DT for music genre classification by utilizing frequent chord

sequences to induce context free definite clause grammars of music genres. They use a first order logic extension of a C4.5 DT algorithm, developed by Blockheel and De Raedt [9].

Artificial Neural Network (ANN) is another popular approach often used in classification tasks. It is modeled after biological neural networks, such as the human brain. ANN consists of multiple interconnected artificial neurons. Each one can compute values from inputs and thus, approximate a function. Sets of these neurons are often connected in multiple layers, and together they can represent a non-linear function. Recent examples of using ANN include Dieleman *et al.* [14], who use a *convolutional deep belief network* to learn the parameters for initializing a *convolutional multilayer perceptron* (MLP). Both of these are a variation of ANN. In their work, Dieleman *et al.* [14] demonstrate that their pre-trained MLP outperforms same MLP with randomly initialized weights when used for genre recognition, artist recognition and key detection.

To our knowledge, we are among the first to propose using association analysis for music genre classification. This approach is new to MIR, although other application areas of data mining have successfully applied it. The general idea is to mine association rules for each genre and then, to score a new music piece against all of them. This music piece is assigned the genre with the highest score. Chapter 4 details our approach to mine association rules from content-based features and then use them to predict genres of unlabeled music pieces.

2.4.6 Evaluating Classification

A practical way of examining the results of classification is in the form of a confusion matrix. Each row in this matrix corresponds to the actual class and each column corresponds to the predicted class. When an instance from the testing set is assigned a label, the appropriate cell in the confusion matrix is incremented by one. The diag-

onal of the confusion matrix represents correctly classified instances, and everything falling outside of the diagonal is incorrect. Several evaluation measures can be derived from this confusion matrix; in our work we, use the following three.

Recall [21], also known as *sensitivity*, is computed by dividing the number of correctly classified instances of a class by the sum of all of the cells in that class' row. It represents the percentage of correctly classified instances for that class.

We compute *precision* [21] by dividing the number of correctly classified instances of a class by the sum of all of the cells in its corresponding column. It reflects the percentage of correctly classified instances from all instances that are perceived as belonging to that class by the classifier.

Finally, *accuracy* [21] is obtained by dividing the number of all correctly classified instances for all classes by the total number of predictions made, in other words, the sum of all of the cells in the diagonal is divided by the sum of all of the cells in the confusion matrix.

Chapter 3

Verifying Tag Annotation via Association Analysis

In this chapter, we present an approach to verify music tags and tag annotation process via association analysis. For a more condensed version of this chapter, please refer to Arjannikov *et al.* [5].

First, we outline the complete approach in Section 3.2, which consists of four stages. It is usable as outlined in this chapter; however, we run into some major obstacles and are unable to test our approach directly. We expose these obstacles in Section 3.3 and suggest a series of simulations to overcome these issues in Section 3.5. We outline the data that we use in our simulations in Section 3.4.

3.1 Introduction

Tags for a given music piece reveal the inherent musical nature that it attempts to convey and express. As a coherent expression, these tags represent features that distinguish this music piece from others. This expression intuitively shows a strong association of these tags to the music piece or to a set of similar music pieces in terms of their musical nature. We work toward this intuition and aim to capture associations between tags and utilize them to verify the annotation process.

3.2 Proposed Approach

Music tags associated with a particular piece of music are the semantic indicators of its content, which people perceive and understand. Hence, we propose to solicit a group of experts on the subject of music to aid the verification process. In our approach, we propose four stages, as shown in Figure 3.1. To make our approach successful, we require that the experts initiate the process in Stage 1 and complete it in Stage 4.

Suppose we are given a task to verify the tag annotation process for some automatic tagger. Given a music repository and a group of music experts, we are ready to start the verification. In the first stage, we create a dataset, which is representative of the music repository, and a set of tags to be used during the annotation process. In this stage, the experts annotate the representative music pieces and produce a transactional style dataset. We mine association rules from this dataset in the second stage. Then, in the third stage, we use the automatic tagger on the rest of the music repository and produce a transactional style dataset of music annotations. In the fourth stage, we use the association rules from second stage and verify the results of automatic tagging. The experts would then examine the results and propose adjustments to the automatic tagger. This process is depicted in Figure 3.1 and described in further detail in the following sections.

3.2.1 Stage 1: Initialization

We start our verification process by asking the experts to select a representative set of music pieces from the given repository. This has to be done in such a way as to preserve the key aspects of the repository, for instance, genres. It may not be easy to select a good sample set, which is why we need the experts to do this. Afterwards, we ask them to select a set of tags T, as illustrated in Figure 3.1, to be used during the annotation process. Then, we ask the experts to annotate the sample set with the tags

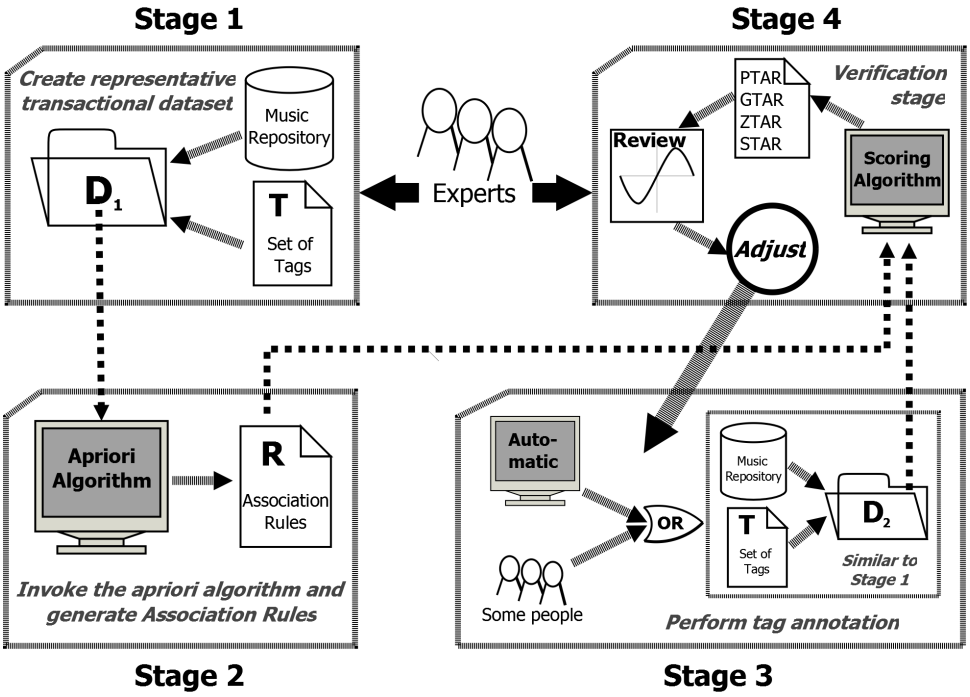

Figure 3.1: The four stages of our proposed approach to verification of tag annotation through association analysis.

from T and to produce a transactional style dataset D_1. They must reach a reasonable level of agreement before proceeding past this point. Thus, D_1 represents the expert knowledge about the selected music pieces and, in turn, the whole repository.

3.2.2 Stage 2: Association Rule Mining

In this stage, we simply mine the association rules from D_1, which we will use in the last stage. In our simulations, we use the *Apriori* algorithm [2, 3]. It find the association rules in a transactional database that satisfy user-specified *minimum support* and *minimum confidence*, which are adjustable by whomever is supervising the verification process. Each parameter pair may produce different amount of rules, which could also differ in how interesting they are. The association rules for R obtained in this way represent relationships within the given data. Our intuition is that similar data annotated using the same set of tags should contain the same relationships. Any disagreement would be indicative of disagreement between the annotators

who produced the metadata under analysis.

3.2.3 Stage 3: Tag Annotation

In this stage, one of two things can happen. If the given task is to verify tag annotations in an existing repository, then we convert them into a transactional format dataset D_2 and proceed to Stage 4. Otherwise, if the task is to verify new annotations or to verify the tag annotation process, then the new annotations must come from the same set of tags as in Stage 1. This ensures that the resulting dataset D_2 is compatible with the rules set R generated in Stage 2 of the proposed verification process.

3.2.4 Stage 4: Verifying Tag Annotation

Whether we want to verify existing annotations or the new ones generated in Stage 3, we need some indication of agreement between the tags to be verified and the experts' opinions represented by the association rules set R. To that end, we design a scoring algorithm to compute each music piece's agreement with the experts. We then derive a set of evaluation measures, which help interpret the score.

Scoring Algorithm

Algorithm that scores the annotations in D_2 against association rules set R

```
1.    for each d_j ∈ D_2 {
2.        S(d_j) = 0;
3.        for each rule p_k = (A → B) ∈ R
4.            if A ∈ d_j then
5.                if B ∈ d_j then {
6.                    record the rule in hit_list
7.                    S(d_j)++
8.                }
9.        for each rule p_k = (A → B) ∈ R
10.           if A ∈ d_j then
11.               if B ∉ d_j then
11.                   if A ≠ any antecedent in the hit_list then
14.                       S(d_j)--
15.   }
```

As described in Section 2.3.1, each music piece m_j in the repository is represented as a vector of annotation tags d_j in the transactional style dataset D_2, and the *annotation score* $S(d_j)$ represents the number of association rules in R, which are satisfied by the annotations of m_j. The higher the score, the better the annotation agreement between the tags in d_j and the association rules in R. Since these rules are derived from the metadata created by experts, higher score means better annotation quality. Then, the low score indicates problematic annotations or issues with the annotation process.

The annotation score $S(d_j)$, initialized to be 0, is calculated as follows. For each association rule $A \rightarrow B$, if the song m_i contains both the antecedent and the consequent, then we increment the song's score $S(d_j)$ by 1. However, if it contains the antecedent but not the consequent, then we decrement $S(d_j)$ by 1 instead. There are also situations where $A \rightarrow B$ and $A \rightarrow C$ coexist in R, representing the multi-label nature of tag annotation. If a music piece misses the first rule but satisfies the second one, we still increment its score by 1 instead of 0. To achieve this, we iterate through R twice. At first, we build a list of rules (*hit_list*), which are satisfied by the song m_j, and increment $S(d_j)$ accordingly. Then we look for the rules which are not satisfied by the song's annotations, such as $A \rightarrow B$, where $A \in d_j$ but $B \notin d_j$. If their antecedents are not found in the *hit_list*, then we decrement the song's score $S(d_j)$ by 1 for each problematic rule.

Evaluation Measures

If a music piece has a positive score, we say that it has a *sound tag annotation* (*STA*). Otherwise, we say that it has a *problematic tag annotation* (*PTA*). Furthermore, we attempt to distinguish music pieces that have a certain degree of ambiguity and subjectivity. A music piece has a *gray tag annotation* (*GTA*) if its annotation score is between [l,h], where l and h are user-specified range values. For instance,

they can be -1 and $+1$ respectively. In our simulations below we use $[-2, 0]$ for this range. A music expert, depending on her/his musical training and experience, may set a different range.

Based on the above, we calculate four measures. The first is the *Problematic Tag Annotation Rate* (*PTAR*); it is the ratio between the number of music pieces with PTA and the total number of music pieces. The second is the *Sound Tag Annotation Rate* (*STAR*), which is the ratio between number of STA music pieces and the total number of music pieces. These two ratios represent the quality of the tag annotations. As mentioned earlier, they indicate a level of agreement between the experts and the annotations in question. High values of STAR show high agreement and high values of PTAR indicate that there are problematic annotations in D_2. In addition, we calculate the *Gray Tag Annotation Rate* (*GTAR*), which is the ratio between the number of GTA music pieces and the total number of music pieces in the dataset. It represents the uncertainty in the annotation process. The fourth measure that we calculate is the *Zero Tag Annotation Rate* (*ZTAR*), which represents the percentage of music pieces that do not contain any of the association rules. It is easy to see that these measures divide the whole music set D_2 into partitions and add up to 1.

3.2.5 Adjustments

At this point, the experts play an important role as they examine the evaluation measures and find problematic annotations or music pieces. They could also provide solutions, such as manually annotating problematic music pieces or recommending changes to the annotation process that produces problematic tags. We demonstrate the effect of such adjustments in Section 3.5.4, where some simple data cleaning produces an improvement in quality of annotations, which is reflected by our proposed evaluation measures.

3.3 Various Issues

We are facing some major challenges when examining the effectiveness of our proposed approach to verify tag annotation.

It would be ideal to examine our approach using the process as depicted by Figure 3.1. Involving a group of experts with extensive music background and experience could prove invaluable, as they would inevitably provide important feedback about our approach. However, such a process involves a great amount of financial and labor costs. Given our current situation and circumstances, it would be extremely hard, if not impossible, for us to deploy such a process. Hence, instead of employing music experts, we do the next best thing. We obtain already annotated datasets that differ in their annotators' levels of musical expertise.

However, it is often the case that different music repositories and datasets vary in the sets of tags used to annotate their music pieces. Therefore, it is possible for two datasets to be completely incompatible in such a way that one could not be used to verify another. Furthermore, association analysis captures each dataset's own associations that do not necessarily translate to other datasets.

As mentioned in Chapter 2, there is a lack of large music datasets with good quality ground-truth annotations, which could be used for benchmarking. They are either of good quality but small, or large but of poor quality. The datasets that appear in this work clearly depict this problematic situation in MIR. However, the MIR community is making efforts to come up with good datasets, two datasets serve as evidence of this: the Million Song Dataset [8, 34] and the Million Song Dataset Benchmarks [44].

We overcome the above issues by setting up a series of simulations to verify the different parts of our proposed approach. The details are in Section 3.5. But first, in the next section, we describe the datasets that we use in our simulations.

Dataset name	Number of songs	Number of labels	Label cardinality	Songs with at least 2 tags
D_{CAL500}	502	174	26.04	502
D_{CAL10K}	10886	1053	11.88	10886
D_{MAGNA}	25863	188	3.46	18097
$D_{\text{MAGNA-CLN}}$	25863	188	4.74	18097
$D_{\text{MAGNA-ADJ}}$	25863	166	5.15	13991
D_{LASTFM}	449503	522366	15.94	365878
$D_{\text{LASTFM-CLN}}$	449503	1046	8.76	280922
$D_{\text{LASTFM-ADJ}}$	449503	287	7.13	315254

Table 3.1: Music datasets and their statistics. The *label cardinality* of a dataset is the arithmetic mean of the number of labels per music piece in the dataset.

3.4 Datasets in Simulations

The task at hand requires music tag annotations and we find four MIR datasets of various annotation quality. These datasets are listed with their statistics in Table 3.1. Each contains music tags obtained via a manual annotation process. They also contain a reasonable amount of music pieces with at least two tags; we need at least two to form a complete rule in the form of $A \rightarrow B$.

The 500-song dataset from the Computer Audition Laboratory [51], D_{CAL500}, is a popular dataset among MIR researchers and it appears in many publications. It comes from a set of 502 songs by different artists from a collection of western popular music from a fifty-year span between 1950 and 2000. The author of the dataset paid 66 undergraduate students who manually annotated the music using a survey-style web form consisting of 135 concepts. At least three people annotated each song and each of the tags had at least 80% agreement rate amongst the annotators. There are 174 unique tags with an average of about 26 tags per song. In our work, we use the "hard" annotations found in D_{CAL500}, which give a binary value for all tags in every song indicating whether a tag applies to a song or not.

The 10,000-song dataset from the Computer Audition Laboratory [49], D_{CAL10K}, is comprised of metadata collected from the Pandora website. It consists 10,870 songs

annotated by expert musicologists who maintain a high level of agreement. This dataset comes in four parts, one containing 475 acoustic tags, one with 153 genre tags and one part contains both. The fourth part contains only the labels that correspond to the D_{CAL500} dataset, 55 in total. This means that we can use both datasets in experiments together; we further use this information and find the same subset of tags in the remaining datasets.

The *Magnatagatune* dataset [27], D_{MAGNA}, is a result of an online game, referred to as "TagATune", developed to collect tags for music and sound clips. It contains information about 21,642 music clips using 188 different tags. Although participants were mostly amateurs, the dataset maintains only the tags that are associated with at least 50 clips and only the ones that were generated independently by at least two players. It is important to note that, in the past, Magnatagatune has not been used as widely as D_{CAL500} due to its size and skewed tag distribution. Before using D_{MAGNA} in our simulations, we first derive $D_{\text{MAGNA-CLN}}$ from it by performing some simple data cleaning, such as removal of tracks with label cardinality less than 1. We also produce $D_{\text{MAGNA-ADJ}}$ by small adjustments, further discussed in Section 3.5.4.

The *Million Song Dataset* (*MSD*) [8] is the largest MIR dataset publically available to date. The purpose behind its creation is to encourage research of scalable solutions and to provide a reference dataset for benchmarking. Unlike all other datasets, MSD is a cluster of complimentary datasets contributed by the community. It contains a wide range of metadata such as genre tags, content-based features, song similarity, user taste profiles and even lyrics. For our simulations, we use the portion of MSD provided by LastFM, excluding the known duplicates; for this reason, we denote it as D_{LASTFM}. This portion contains a set of tags associated with the tracks in MSD. Here, the music pieces are annotated by the users of the Last.fm web site resulting in many tags that are not useful for our task, such as "my favorite song" or "awesome". Hence, before using this dataset in our experiments, we first derive $D_{\text{LASTFM-CLN}}$ from

it via some basic data selection and cleaning procedures, similar to D_{MAGNA}. Then, we produce $D_{\text{LASTFM-ADJ}}$ through small adjustments to $D_{\text{LASTFM-CLN}}$ further discussed in section 3.5.4.

3.5 Simulations

Considering the issues with our proposed approach outlined in Section 3.3, we design and implement a series of simulations to demonstrate the effectiveness of our proposed approach.

Through these simulations, we aim to achieve three goals: (G_1) demonstrate that our approach is stable, in that it will not behave arbitrarily when given different music datasets or, put in another way, given similar music datasets, it should behave similarly; (G_2) assess the four music datasets using our evaluation measures and confirm that they maintain different relationships among their tags; and (G_3) confirm that, when the quality of annotations in a music dataset improves, our proposed measures reflect this improvement.

We run our simulations with different minimum support and confidence value pairs; for each minimum support value ranging from 5% to 95% we use a different minimum confidence ranging from 5% to 95%. We explain each simulation in further detail and list the results in the following sections.

3.5.1 Simulation 1 - Stability Test

This simulation demonstrates G_1, that our approach is stable. Here, we randomly split a dataset in half and see if the resulting halves, H_1 and H_2, produce similar results in terms of our evaluation measures, which we outlined in Section 3.2.4. In this simulation, for each half, we go through all of the sages depicted by Figure 3.1 except the "review" and "adjust" steps in Stage 4. First, we split each half into two subsets at random: we call one (30%) the *training set*, it corresponds to D_1, and the

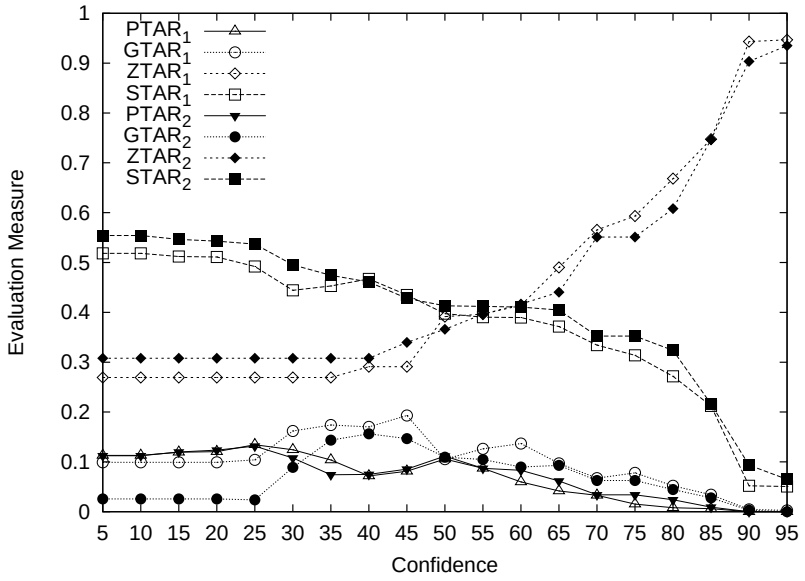

Figure 3.2: Comparison between H_1 and H_2, where $H_1 \subset D_{\text{CAL10K}}$, $H_2 \subset D_{\text{CAL10K}}$, and $H_1 \neq H_2$.

other (70%) corresponds to D_2 and we call it the *testing set*. Then we compare the results from the Scoring Algorithm between H_1 and H_2.

Figure 3.2 illustrates the different measure values that we obtain when we apply our approach to D_{CAL10K}. Here, the minimum support is set to 5% while the confidence increases from 5% to 95%. We observe that STAR values between the two halves are very similar at all confidence levels. The same applies to all other measures that we discuss in Section 3.2.4. We find similar results in the other three datasets across various minimum support thresholds, and thus, conclude our work toward G_1. Our approach is stable.

3.5.2 Simulation 2 - Datasets Evaluation

Here, similar to Simulation 1, we randomly divide each music dataset into two sets, training (30%) and testing(70%). They respectively correspond to D_1 and D_2 in Figure 3.1. We derive the association rules from the training set and score each music piece in the testing set against these rules.

Since D_{CAL500} is too small to produce good sample size, we perform our simulation

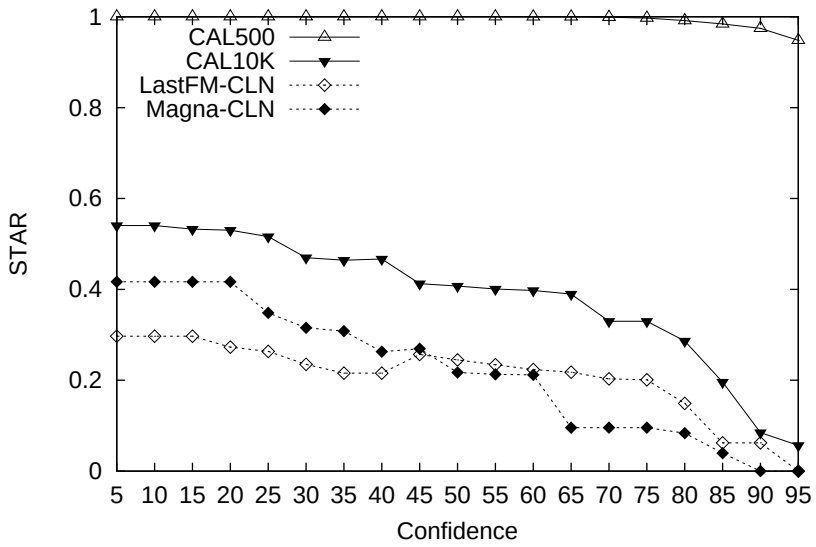

Figure 3.3: Comparison of STAR between four datasets. Note that D_{CAL500} is almost inline with top border of the graph

10 times and obtain an arithmetic mean for each of our evaluation measures, similar to the 10-fold cross validation. Each time we choose the training set at random, and the remainder of the dataset becomes the testing set. The other three datasets are large enough and do not require this kind of repeated random sub-sampling validation.

When we compare the results in terms of our evaluation measures between all four music datasets, we observe that they maintain different relationships among their tags. We report the STAR values for all four datasets in Figure 3.3. The figure shows that D_{CAL500} clearly achieves the highest STAR values, when compared to the other three datasets. The same applies to other measures. For instance, we have observed lower PTAR values from D_{CAL500} and higher ones from $D_{\text{LastFM-CLN}}$, as expected.

This confirms that the different datasets used in our simulations are of different annotation quality. This is probably because they were created using different annotation processes and contain different tags. Hence, the datasets retain different relationships among their tags.

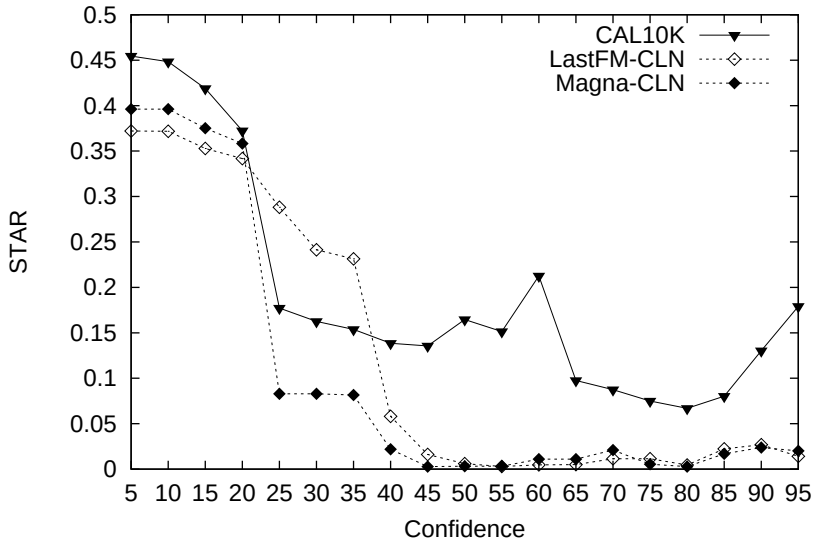

Figure 3.4: Using D_{CAL500} to verify the other datasets.

3.5.3 Simulation 3 - One Verifies Another

In this simulation, we use one dataset to verify another. Since CAL10K provides 55 tags that appear in all four datasets, we reduce all datasets to just those tags. Then we use one dataset as the training set D_1 and another as the testing set D_2; we perform all steps outlined in Figure 3.1 except the adjustment cycle.

Our previous simulations clearly show that dataset D_{CAL500} has a better tag annotation quality than the other three datasets, in terms of our evaluation measures. Therefore, we use D_{CAL500} as the representative dataset, D_1 in Figure 3.1, to evaluate the other datasets, each considered as D_2 in the same figure. As can be seen in Figure 3.4, except for a few lower confidence ranges, D_{CAL10K} outperforms the other two datasets in terms of STAR. The same applies to the other measures. This confirms that D_{CAL10K} maintains higher quality annotations than the other two.

3.5.4 Simulation 4 - Adjustment Demonstration

To fully demonstrate our proposed approach, we adjust the two datasets that seem to be the worst in terms of our evaluation measures in Simulation 2, namely the $D_{\text{LastFM-CLN}}$ and the $D_{\text{Magna-CLN}}$. This small adjustment consists of amalgamating

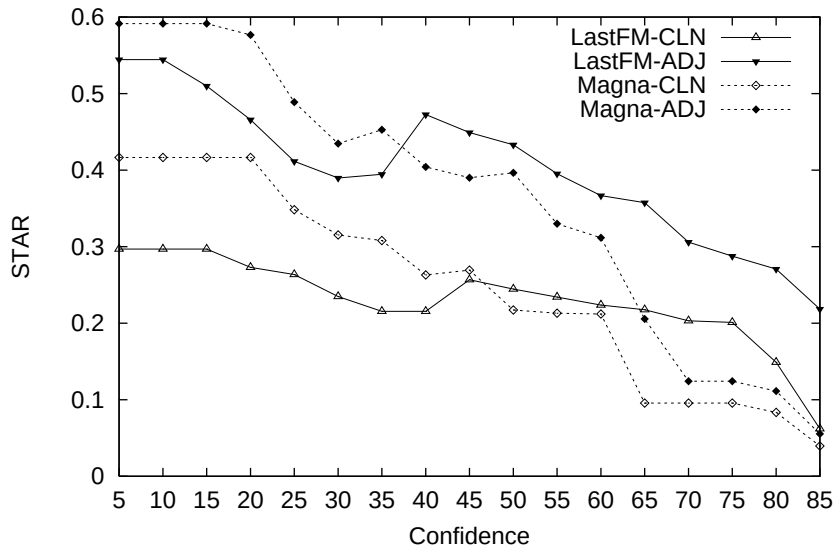

Figure 3.5: Comparison of STAR between two datasets before and after the adjustment step.

some semantically equivalent annotations into one tag, thus reducing the diversity of tags in the datasets. For example, we convert the tags {*male vocals*}, {*instrument singer male*}, {*male singer*} and {*male voices*} into one tag {*male vocals*}.

Note that after the adjustment step, $D_{\text{LastFM-ADJ}}$ has a larger number of tracks in than before, although the label cardinality is slightly lower. This is because after amalgamating some tags, one tag took place of several. This means that the tracks whose tags did not previously match any in our set are now included. Although, after the adjustment, the label cardinality becomes slightly lower, the overall quality of the dataset becomes higher, as evident in this simulation. After the adjustment, we run the Scoring Algorithm and observe some improvements in terms of our evaluation measures.

In Figure 3.5, we present the STAR values for both datasets before and after the adjustment steps. Both datasets show an improvement in terms of STAR after the adjustment step. The same can be said about the other measures. This demonstrates that we achieve our goal G_3, as we confirm that when the quality of annotations improves, our evaluation measures proposed in Section 3.2.4 reflect this improvement.

3.6 Discussions

In this chapter, we presented an approach to the verification of music tag annotation through association analysis. We believe that there exist inherent associations among music tags that can be further utilized to verify and monitor a tag annotation process. The above simulations demonstrated the effectiveness of our approach. We find that association analysis can be used for both: to verify the quality of tags in a music repository and to help analyze the annotation process through which music tags enter the repository.

An interesting observation can be made about the behaviour of ZTAR measure in response to label cardinality in a music dataset. When there are too few tags associated with each music piece, the ZTAR is high, and the opposite happens when there are many tags associated with each music piece. A high ZTAR is undesirable, because our approach relies on finding the level of agreement between the experts opinions represented as a set of association rules and the music piece that is being scored. If the music piece does not match any of the rules' antecedents, then we cannot gauge its level of agreement.

Chapter 4

Classifying Music Into Genres Via Association Analysis

In this chapter, we present an approach to classify music into genres via association analysis. For a more condensed version of this chapter and additional results, please refer to Arjannikov and Zhang [6].

This chapter is organized as following. First, a brief introduction to the use of association rules for the purpose of music genre classification. Then, in Section 4.2, we outline our experiment setup. We start with the goals for the experiments and the data that we use, followed by the details about the proposed classifier. We present the results of our experiments in Section 4.3. We then conclude this chapter with a brief discussion about the experiments.

4.1 Introduction

As suggested in Section 2.4, any discrete set of tags that are not correlated can be used as categories, or classes, into which we could split a collection of music pieces. This has been proven to work for, but is not limited to, mood and emotion related tags, instrument tags, and the ones we work with here – the music genre tags [16, 29].

In the previous chapter, we demonstrate that association analysis reveals patterns in music tags; this motivates our investigation of patterns in content-based music features. Because most of these features are not discrete, first, we use a discretization technique to form discrete sets of labels from each feature. Then we derive a transac-

tional style dataset from these labels, which is suitable for association analysis, and mine it for frequent patterns using the *Apriori* algorithm [2, 3]. Finally, we use these frequent patterns to identify to which genre a given music piece belongs.

4.2 Experiment Setup

4.2.1 Goals

Our aim is to test if association rules mined from content-based music features can be used for genre classification purposes. With this in mind, we designate three goals: (G_1) our classifier achieves a classification accuracy that is better than choosing genres at random; (G_2) we expect that our classifier will perform beter when given fewer classes to choose from; (G_3) our classifier achieves higher accuracy with better quality data.

4.2.2 Data

The classification task at hand requires content-based features paired with genre tags and we find two datasets that fit this description, as listed in Table 4.1.

Dataset name	Number of songs	Number of genres	Number of features	Type of Features
D_{LMD}	3000	10	26	MFCC
D_{MSDB}	1700	17	10	Methods of Moments

Table 4.1: Music genre datasets and their statistics.

The *Latin Music Database* [46, 47], denoted as D_{LMD}, is popular in the music genre classification task despite of its small size. There are many classification results available in the literature, which are based on a set of features that has already been extracted and circulated as part of D_{LMD}. Thus, we can test the feasibility of our approach without introducing variance based on difference in feature extrac-

tion techniques. This dataset usually results in high classification accuracy for many methods [47].

Million Song Dataset Benchmarking [45], denoted as D_{MSDB}, is much larger than D_{LMD} and boasts a large set of content-based features, which is an extension of the ones originally provided with MSD. This dataset is distributed for the purpose of comparing different approaches. There are several sets of features in D_{MSDB}, we use one of them in combination with genre labels, which were originally obtained from Allmusic [45]. Additionally, we restrict the number of tracks to 1000 per genre, in order to balance the number of training and testing examples between genres.

We split each dataset into two partitions at random, while maintaining the genres balanced. Thus, each genre is represented by equal number of tracks in both partitions. One of the partitions becomes the training set and the other becomes the testing set. It is important to note that we do not perform any kind of repeated random sub-sampling validation in our experiments, because we consider the size of the datasets to be an acceptable sample size, which is representative of the chosen genres. If there are too many music pieces belonging to one genre as compared to others, we remove the extra tracks at random. If a genre is represented by fewer pieces than 300 for D_{LMD} and 1000 for D_{MSDB}, then we remove that genre from our experiments. This reduces the original D_{MSDB} dataset to 17 genres from 25. D_{LMD} remains at 10 genres because it was originally balanced at 300 music pieces per genre. We include some statistical information about these datasets in Table 4.1.

4.2.3 Classifier

We set up our proposed classifier during the training stage in several steps. First, we acquire content-based features from music; in this thesis, we use the features that have already been extracted and published for the purpose of comparing different classifiers on even ground. Then, we discretize any continuous features into some

pre-determined number of bins, as explained in Section 2.4.3. In our experiments, we try different values of this parameter to determine the number of bins that results in a reasonable classification accuracy. If the classifier performs better than random genre labeling, then we consider it a reasonable accuracy and attain G_1. After that, we split the training dataset into subsets, one for each genre label, and convert them into transactional format. Then, we invoke the *Apriori* algorithm [2, 3] and mine frequent patterns from these genre sets at some minimum support value. We try different values for this parameter in order to find the support values that produce reasonable classification accuracy during our experiments. From these frequent patterns we create classification rules of the form $A \rightarrow B$, where A is the frequent pattern and B is the genre associated with that pattern. Then, we compute the confidence value for each rule by dividing the number of songs that matches the rule $A \rightarrow B$ by the number of songs that match the frequent pattern component of the rule, A. We use it later in our scoring algorithm. Finally, we find any patterns that co-exist in two or more genres and remove them, thus ensuring that there is no intersection among the sets of patterns between any pair of genres. The resulting rules become representative of their respective genres. This concludes our training stage and we use the acquired rules for classification.

4.2.4 Scoring

To obtain a classification score for each genre, we use the following four components. *Pattern Percentage* (PP) is the percent of patterns that a given music piece matches for a given genre out of all patterns matched from that genre. *Support Sum* (SS) is the support sum of the matched patterns divided by the support sum of all patterns for the given genre. *Confidence Sum* (CS) is the current genre's confidence sum of the matched patterns divided by the sum of all patterns' confidence. Finally, *Length Sum* (LS), which is the sum of cardinalities of the matched patterns divided

by the sum of cardinalities of all patterns for the given genre.

We score each music piece against each genre's set of rules in two ways. (i) The first score, denoted as V_{SUM}, we obtain by first summing the four components for each genre and then taking a vote across all genres. Thus, each genre receives an independent score and the one with the highest score becomes the predicted class. (ii) The second score is denoted as V_{VOTE}. Here, we deal with each component individually. First, we create a voting vector, whose cardinality is equal to the number of genres, and compute the corresponding component's value for each genre. Then, the genre with the highest value is voted as a candidate for that component. Thus, the four components result in up to four votes per genre. The genre with the highest number of votes is declared as the winner and becomes the predicted class of the given music piece. The imaginary example in Table 4.2 illustrates the difference between V_{VOTE}, which picked *Forro*, and V_{SUM}, which picked *Salsa* as the predicted class. The example also shows how V_{VOTE} deals with ties. Of course, if there is a tie between two or more genres, then all of them are put forth as the predicted class. This situation is captured by the MLR evaluation measure.

GENRE	PP	SS	CS	LS	V_{SUM}	V_{VOTE}
Forro	0.8	**0.03**	**0.003**	**0.0001**	0.8331	**3**
Salsa	**0.9**	0.01	0.001	**0.0001**	**0.9111**	2
Tango	0.7	**0.03**	0.002	**0.0001**	0.7321	2

Table 4.2: Example scores for three genres. Bolded are the highest/winners.

4.2.5 Evaluation Measures

To evaluate our classifier, for each dataset we create two confusion matrices, one for scoring method. From these matrices, we derive *recall*, *precision*, and *accuracy*, as described in Section 2.4.6. We also compute an additional evaluation measure, which we describe below.

Because our proposed classifier can assign multiple genre labels to a single music

piece, we also measure the rate at which this happens. We compute the *Multi-Labeling Rate* (MLR) by dividing the sum of all cells in a confusion matrix row by the number of all test instances of the corresponding genre. MLR falls into the range between 1 and the total number of classes in the classification task. On the one hand, the closer it is to 1, the fewer multi-label assignments were made, which indicates that the classifier is performing more like a single-label one. This is desirable in our case, because we remove any patterns that belong to more than one genre. On the other hand, the larger the number, the more diluted is the resulting classification. If MLR is equal to the total number of classes, then the results of classification are least useful. Furthermore, if MLR is below 1, then there are music pieces, whose genres could not be predicted.

4.3 Experiment Results

In Table 4.3 and Table 4.4 we present the classification results of our proposed classifier, where we compare the two scoring methods, V_{SUM} and V_{VOTE}. We observe that there is no significant difference between all of our evaluation measures, which is evident from the confusion matrices. This is true for both datasets in all confusion matrices in our experiments; ergo, we present the results of only the combination vote between the four components thereafter.

In the following three sections, we demonstrate through our experiment results how we achieve the three goals formulated in Section 4.2.1.

4.3.1 G_1 Feasibility Test

During our experiments, we observe that for some values of minimum support and for some numbers of bins, our classifier performs much better than choosing genre assignment at random, which would be 1/10 for D_{LMD} and 1/17 for D_{MSDB}. However, with other values of these parameters, our classifier predicts majority of

Axe	Bachata	Bolero	Forro	Gaucha	Merengue	Pagode	Salsa	Sertaneja	Tango	Genres
24	6	0	2	39	34	5	13	26	1	Axe
5	98	3	5	5	3	0	18	13	0	Bachata
4	3	54	8	13	1	2	46	18	1	Bolero
10	7	3	24	16	19	6	39	25	1	Forro
22	8	3	8	64	8	7	18	12	0	Gaucha
9	17	0	3	13	86	5	5	11	0	Merengue
17	1	10	3	3	17	26	37	36	0	Pagode
5	9	6	12	16	6	7	75	12	2	Salsa
17	2	7	4	24	2	7	52	35	0	Sertaneja
0	0	25	1	10	0	1	3	3	107	Tango

Table 4.3: The V_{SUM} confusion matrix for D_{LMD} discretized into 2 bins with minimum support of 30%. Average MLR = 1.00; average precision = 42.7%; average recall = 39.6%; accuracy = 39.6%.

songs to be of one class. There are also cases where it votes for all genres equally, producing MLR that is equal to the number of genres. Because our classifier achieves accuracy that is much better than random with some parameter settings, we conclude our work towards G_1. Moreover, we observe that our proposed parameters affect the classification accuracy, and thus, they are effective. Since it is not interesting to see poor classification results, we do not include those, where MLR is too high. We also omit the results where our classifier predicts all music pieces to belong to only one or two genres and not any others. The latter can be observed in Table 4.9, where the *Blues* genre did not receive many predictions.

To demonstrate that our parameters affect the classifier performance we use the Table 4.4 as a starting point. Then, we change the minimum support from 30% to 80%, shown in Table 4.5 and observe a change in all of our evaluation measures. Next, we alter the number of bins from 2 in our starting point to 20 in Table 4.6 and then to 30 in Table 4.7, while keeping the minimum support at 30%. Again, we observe a change in our evaluation measures. Although MLR remains similar, the other measures differ. Moreover, we notice that at 30 bins, one of the genres, *Salsa*, is removed. Although we mine some patterns for it, all of these patterns are found in

Axe	Bachata	Bolero	Forro	Gaucha	Merengue	Pagode	Salsa	Sertaneja	Tango	Genres
23	6	0	2	39	35	6	13	26	1	Axe
5	99	3	5	4	3	0	18	13	1	Bachata
4	3	54	8	13	1	2	47	18	1	Bolero
10	7	3	24	16	19	6	40	25	1	Forro
22	9	3	8	64	8	7	18	12	0	Gaucha
9	17	0	3	13	87	5	5	11	0	Merengue
17	1	10	3	3	18	27	37	37	0	Pagode
5	9	6	12	16	7	6	78	12	2	Salsa
17	2	7	4	24	2	7	52	35	0	Sertaneja
0	0	24	1	10	0	1	3	3	108	Tango

Table 4.4: The V_{VOTE} confusion matrix for D_{LMD} discretized into 2 bins with minimum support of 30%. Average MLR = 1.01; average precision = 42.7%; average recall = 39.7%; accuracy = 39.6%.

other genres and are subsequently removed.

4.3.2 G_2 Reduced Number of Classes

As demonstrated in the literature, classification accuracy usually increases when the number of classes is reduced [30]. For G_2, we classify D_{LMD} and D_{MSDB} with reduced number of genres and confirm that our classifier performs better for both datasets in terms of our evaluation measures on the reduced set of genres, while everything else remains unchanged. This is clearly seen when we compare Table 4.4 and Table 4.8, where we reduce the number of genres in D_{LMD} from 10 to 5. This also holds for D_{MSDB}, when we compare Table 4.9 and Table 4.10, where we reduce the number of genres from 15 to 5. Thus, we conclude our work towards G_2. Our classifier achieves higher accuracy on a smaller number of classes.

4.3.3 G_3 Dataset Quality

To achieve G_3, we compare the classification results between the two different datasets, D_{LMD} and D_{MSDB}. This helps us confirm that our classifier achieves higher accuracy, when given better quality datasets. As documented in literature, classifiers usually achieve higher classification accuracy with D_{LMD} than most other datasets,

Axe	Bachata	Bolero	Forro	Gaucha	Merengue	Pagode	Salsa	Sertaneja	Tango	Genres
82	48	5	70	44	74	40	50	48	6	Axe
5	90	16	22	7	12	17	56	19	3	Bachata
5	25	61	39	15	4	35	97	31	33	Bolero
22	65	30	96	53	26	60	82	54	2	Forro
54	66	8	70	67	56	28	73	50	4	Gaucha
29	34	2	32	19	104	20	35	16	2	Merengue
36	37	35	80	47	26	78	86	59	9	Pagode
15	30	39	67	26	8	57	81	39	15	Salsa
44	38	29	63	43	24	51	65	76	11	Sertaneja
0	81	12	0	2	1	0	75	0	96	Tango

Table 4.5: The confusion matrix for the D_{LMD} discretized into 2 bins with minimum support of 80%. Average MLR = 2.60; average precision = 24.5%; average recall = 23.1%; accuracy = 21.3%.

among which is D_{MSDB}. To demonstrate this, we use the results from D_{LMD}, where we maintained constant minimum support at 30% while changing the number of bins from 2 to 20 and then 30. The results of these are presented in Table 4.4, Table 4.6, and Table 4.7. Then, we classify D_{MSDB} at minimum support of 30%, while changing the number of bins from 2 to 20 and then 30; these results are in Table 4.11, Table 4.12, and Table 4.13. Not only the accuracy is worse for all cases, and LMR is higher in Table 4.11, we also notice that, in Table 4.13, four genres, *Blues*, *Electronic*, *Jazz*, and *PopRock*, are removed from D_{MSDB}, because none of their patterns is unique among all genres. Moreover, we observe that in all of our experiments, our classifier never achieves better accuracy with MSDB than with LMD, listed in table 4.1. Thus, we confirm that our proposed classifier performs better with D_{LMD} than D_{MSDB}, and conclude our work for G_3.

4.4 Discussions

First point of note is that, during our experiments, we removed the intersections between pattern sets of every genre pair. These intersections may indicate similarities between genres, which could help reveal the multi-genre nature of music. Our ex-

Axe	Bachata	Bolero	Forro	Gaucha	Merengue	Pagode	Salsa	Sertaneja	Tango	Genres
36	6	11	25	15	10	21	18	11	8	Axe
3	101	10	4	8	4	5	19	1	2	Bachata
2	19	48	4	7	0	17	22	12	28	Bolero
14	13	6	37	9	12	14	31	15	9	Forro
18	15	10	17	39	3	15	19	15	12	Gaucha
12	21	1	38	11	28	7	32	6	1	Merengue
11	24	9	16	5	1	42	36	14	3	Pagode
7	14	13	25	22	4	20	39	11	5	Salsa
16	7	17	26	11	2	14	24	31	7	Sertaneja
0	19	13	0	5	0	5	18	1	95	Tango

Table 4.6: The confusion matrix for the D_{LMD} discretized into 20 bins with minimum support of 30%. Average MLR = 1.06; average precision = 32.4%; average recall = 31.3%; accuracy = 31.2%.

Axe	Bachata	Bolero	Forro	Gaucha	Merengue	Pagode	Sertaneja	Tango	Genres
39	2	13	24	23	8	13	22	7	Axe
9	85	8	13	12	5	15	4	0	Bachata
42	2	24	18	7	0	10	41	9	Bolero
40	3	9	45	14	6	16	25	2	Forro
42	1	6	36	22	3	4	36	6	Gaucha
17	9	6	41	7	46	9	15	2	Merengue
39	2	5	29	8	6	32	32	2	Pagode
49	2	10	20	15	0	5	52	4	Sertaneja
18	0	16	16	7	0	4	17	72	Tango

Table 4.7: The confusion matrix for the D_{LMD} discretized into 30 bins with minimum support of 30%. Average MLR = 1.03; average precision = 37.6%; average recall = 30.2%; accuracy = 30.1%.

periments were designed to test the classifier, however, with some modifications, our approach could be used to explore the multi-label situation in genre classification.

Through our experiments, we observe that, when the number of discretization bins is high, the accuracy is very low. Furthermore, the MLR is also high. This suggests that lower number of bins is more advantageous for classification. We also notice that with high number of bins, for example, 150 bins at 30% minimum support in D_{LMD}, there are many frequent patterns, for instance, *Tango* has 30378 and *Forro* has 53254. However, after removing the intersections between them, *Tango* is left with only 168

Bachata	Bolero	Merengue	Pagode	Tango	Genres
109	8	6	25	2	Bachata
16	70	1	52	12	Bolero
18	1	104	20	6	Merengue
9	14	28	97	4	Pagode
2	29	0	3	118	Tango

Table 4.8: The confusion matrix for the reduced D_{LMD} discretized into 2 bins with minimum support of 30%. Average MLR = 1.01; average precision = 67.1%; average recall = 66.1%; accuracy = 66.0%.

Blu	Com	Cou	Eas	Ele	Fol	Int	Jaz	Lat	Pop	Rap	Reg	Rel	RnB	Voc	Genres
2	22	7	96	18	52	81	39	36	47	9	42	20	9	15	Blues
0	125	2	30	16	8	26	29	62	64	3	61	44	23	14	ComedySpoken
1	34	20	94	1	37	69	20	54	34	8	27	51	19	29	Country
0	20	0	188	7	49	54	32	17	34	2	49	23	5	22	EasyListening
2	17	1	31	93	9	35	52	44	30	50	113	3	11	5	Electronic
3	30	4	98	16	54	101	62	18	20	4	38	16	10	35	Folk
1	29	4	83	22	34	61	41	46	44	19	68	19	16	14	International
0	13	2	111	23	31	56	89	16	18	3	66	45	13	16	Jazz
1	43	13	22	13	20	38	26	74	66	26	66	40	40	15	Latin
2	28	23	35	16	12	37	20	48	144	22	55	21	21	11	PopRock
2	22	4	0	54	3	3	13	69	22	104	146	9	41	3	Rap
2	37	1	13	39	11	22	24	38	11	37	187	28	44	1	Reggae
0	35	15	29	7	14	49	16	66	79	26	54	54	34	18	Religious
2	36	4	27	23	21	38	19	60	35	23	84	54	59	14	RnB
0	30	1	160	14	25	65	64	5	19	4	28	22	11	59	Vocal

Table 4.9: The confusion matrix for the D_{MSDB} discretized into 5 bins with minimum support of 10%. Average MLR = 1.00; average precision = 17.9%; average recall = 17.5%; accuracy = 17.5%.

Com	Eas	Ele	Pop	Rap	Genres
267	79	83	54	37	ComedySpoken
77	282	74	49	18	EasyListening
81	97	182	26	139	Electronic
82	113	61	205	56	PopRock
97	16	201	49	255	Rap

Table 4.10: The confusion matrix for the reduced D_{MSDB} discretized into 5 bins with minimum support of 10%. Average MLR = 1.07; average precision = 45.3%; average recall = 44.7%; accuracy = 44.4%.

Blu	Com	Cou	Eas	Ele	Fol	Int	Jaz	Lat	Pop	Rap	Reg	Rel	RnB	Voc	Genres
459	459	459	459	459	459	459	459	459	459	459	459	459	459	459	Blues
455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	ComedySpoken
482	482	482	482	482	482	482	482	482	482	482	482	482	482	482	Country
483	483	483	483	483	483	483	483	483	483	483	483	483	483	483	EasyListening
455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	Electronic
485	485	485	485	485	485	485	485	485	485	485	485	485	485	485	Folk
462	462	462	462	462	462	462	462	462	462	462	462	462	462	462	International
480	480	480	480	480	480	480	480	480	480	480	480	480	480	480	Jazz
455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	Latin
369	369	369	369	369	369	369	369	369	369	369	369	369	369	369	PopRock
477	477	477	477	477	477	477	477	477	477	477	477	477	477	477	Rap
479	479	479	479	479	479	479	479	479	479	479	479	479	479	479	Reggae
430	430	430	430	430	430	430	430	430	430	430	430	430	430	430	Religious
463	463	463	463	463	463	463	463	463	463	463	463	463	463	463	RnB
482	482	482	482	482	482	482	482	482	482	482	482	482	482	482	Vocal

Table 4.11: The confusion matrix for the D_{MSDB} discretized into 2 bins with minimum support of 30%. Average MLR = 13.83; average precision = 6.7%; average recall = 6.7%; accuracy = 6.7%.

and *Forro* with 50. This suggests that a large portion of the patterns mined does not reflect the difference between genres, but rather, they may be reflecting music in general, if anything at all.

We also observe that, as minimum support rises, the accuracy lowers. This clearly visible when we compare tables 4.4 and 4.5. Although we only present the results on D_{LMD}, this is true for both datasets in our experiments. We conjecture that at high minimum support, not only there are fewer frequent patterns, but also that association analysis captures many patterns that are true for all music. However, at low minimum support values, there are more patterns among which there are those that are more representative of the individual genres.

In our experiments, we notice that it may take a long time to initially analyse the data and build the classifier. However, the resulting classification model is very fast, because it is linear, where the classification complexity is equal to the number of classification rules multiplied by the number of music pieces to be classified.

Blu	Com	Cou	Eas	Ele	Fol	Int	Jaz	Lat	Pop	Rap	Reg	Rel	RnB	Voc	Genres
30	9	22	21	70	93	58	43	15	38	6	15	82	28	28	Blues
5	11	22	10	40	48	43	64	18	16	9	16	190	59	11	ComedySpoken
26	4	50	26	50	88	34	30	18	24	7	13	93	55	23	Country
19	5	21	71	37	77	47	74	9	21	8	15	51	33	64	EasyListening
14	21	11	7	105	39	25	31	18	31	26	10	121	61	21	Electronic
33	4	25	28	62	94	62	63	22	13	11	8	42	49	29	Folk
29	6	18	23	81	63	47	36	19	35	13	15	100	45	23	International
34	8	16	27	46	74	33	84	11	15	11	12	55	65	42	Jazz
10	5	27	8	88	48	34	19	34	33	20	26	149	54	12	Latin
21	5	25	8	52	43	31	18	19	116	10	9	169	38	15	PopRock
5	21	6	1	91	32	17	11	22	13	65	22	182	71	3	Rap
10	15	7	6	92	48	29	21	13	7	42	31	146	63	11	Reggae
14	7	26	8	64	48	39	19	30	54	29	15	148	46	16	Religious
13	12	22	11	63	54	23	21	29	18	37	19	149	59	16	RnB
22	3	21	40	23	114	19	92	9	15	6	8	32	48	86	Vocal

Table 4.12: The confusion matrix for the D_{MSDB} discretized into 20 bins with minimum support of 30%. Average MLR = 1.10; average precision = 14.1%; average recall = 12.5%; accuracy = 12.5%.

Blu	Cou	Eas	Fol	Int	Lat	Rap	Reg	Rel	RnB	Voc	Genres
33	71	106	128	125	59	4	172	172	172	111	ComedySpoken
12	65	37	134	133	88	12	186	186	186	66	Country
12	26	88	122	147	44	4	163	163	163	95	EasyListening
18	28	60	84	146	93	9	131	131	131	90	Folk
3	24	32	74	134	89	37	138	138	138	73	International
6	30	24	85	171	108	51	148	148	148	54	Latin
0	14	21	34	108	67	162	142	142	142	48	Rap
6	19	34	63	93	54	93	159	159	159	88	Reggae
12	37	28	84	151	97	41	148	148	148	49	Religious
11	28	46	75	123	85	71	169	169	169	74	RnB
22	27	96	100	83	49	6	150	150	150	146	Vocal

Table 4.13: The confusion matrix for the D_{MSDB} discretized into 30 bins with minimum support of 30%. Average MLR = 1.44; average precision = 15.1%; average recall = 12.3%; accuracy = 12.0%.

From our experience and other works in the literature, we find that many classifiers, such as the J48 decision tree, perform poorly on unbalanced data [7, 11]. However, D_{LMD} was already balanced and, in order to avoid this issue, we balanced D_{MSDB} as well. Hence, we can compare our results from the two, and leave to future work the study of how unbalanced datasets affect our proposed classifier.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we introduced two novel approaches to MIR, one to verify music tag annotation and the other to assign genre tags to music automatically. At the core of both is a data mining technique, association analysis, which reveals frequent patterns in data. These patterns represent the similarity between all music pieces in a database and, moreover, the similarity of all music pieces in a given genre.

Through simulations and experiments, we demonstrate the effectiveness of both approaches and confirm that association analysis can be applied to music data. Furthermore, we confirm that both content-based and social tags are appropriate for association analysis. However, there is room for improvement in both.

5.2 Future Work

In our work, we use the support and confidence measures exclusively. However, there are others, as mentioned in Section 2.3.4, which we leave to our future work. In addition, determining the interestingness of a derived rule remains an open problem in association analysis, and it is still unexplored in MIR. We conjecture that some rules are more important than the others, based on the task at hand. Thus, in music, some evaluation measures may be more relevant to the verification of tag annotation and some to the classification tasks.

5.2.1 Verifying Tag Annotation

We can extend our work along several directions. It would be very interesting to explore whether we can use content-based information, such as MFCC or ZCR [15], in our analysis and the verification process. We conjecture that this additional information will help improve our approach. Furthermore, we also plan to examine the individual rules that were generated for each music piece. Although we did not pursue these details in our current work, we believe that they could enable us to understand the music pieces in a repository better.

In addition, we could calculate the tag annotation rate for a specific category, such as style, mood and instrument. Furthermore, we could consider the representative music pieces of a single tag. For example, we could examine the tag annotation rules of the music genre *pop*. These rules may provide more insight into the nature of the genre and help us understand why a music piece is associated with it as opposed to others.

5.2.2 Classifying Music into Genres

We believe there are many ways to improve our proposed method, which we leave to future work. These include the improvement of feature extraction, feature selection and discretization. All of these are open problems in MIR and we believe that as they improve, our method will also improve. Also, based on the literature [12, 40], we expect that using social tags in conjunction with content-based features will improve the classification accuracy.

We can also take some immediate steps to improve our classifier by tuning the two parameters, minimum support for frequent pattern mining and the number of discretization bins. Our experiments demonstrate that these two parameters are directly related to the performance of our classifier, and they vary depending on the data. Hence, tuning those parameters to each specific dataset will improve the clas-

sification accuracy.

Another way we could try to improve our classifier is by using various ensemble learning methods. For example, Silla *et al.* [23] show an improvement in classification accuracy by combining classification results from the features extracted from different music segments of the same track, such as first 30 seconds, middle 30 seconds, and the last 30 seconds. Furthermore, our classifier is compatible with existing ensembles, such as, *bootstrap aggregating* [10], also known as *bagging* and *boosting* [43]. In bagging, first, each base classifier is trained using different uniform samples of the training dataset, and then predictions are combined by voting. In boosting the ensemble is built incrementally and each new classifier focuses on errors from the previous one, then the final classifier is used to make predictions. We expect that using our classifier as base classifier in an ensemble will show better accuracy than using our classifier alone.

Bibliography

[1] *Table 102-0512 - Life expectancy, at birth and at age 65, by sex, Canada, provinces and territories annual (years)*, CANSIM (database). Statistics Canada, accessed 2012-05-28.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 22, pages 207–216. ACM, 1993.

[3] Rakeshl Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, volume 1215, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.

[4] Amélie Anglade, Rafael Ramirez, and Simon Dixon. Genre classification using harmony rules induced from automatic chord transcriptions. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 669–674. ISMIR, 2009.

[5] Tom Arjannikov, Chris Sanden, and John Z. Zhang. Verifying tag annotations through association analysis. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 195–200. ISMIR, 2013.

[6] Tom Arjannikov and John Z. Zhang. An association based approach to genre classification in music. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 95–100. ISMIR, 2014.

[7] Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[8] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 591–596. ISMIR, 2011.

[9] Hendrik Blockeel and Luc De Raedt. Top-down induction of first-order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, May 1998.

[10] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[11] Nitesh V. Chawla. C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *In Proceedings of the International Conference on Machine Learning Workshop on Class Imbalances*, 2003.

[12] Ling Chen, Phillip Wright, and Wolfgang Nejdl. Improving music genre classification using collaborative tagging data. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 84–93. ACM, 2009.

[13] Christopher DeCoro, Zafer Barutcuoglu, and Rebecca Fiebrink. Bayesian aggregation for hierarchical genre classification. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 77–80. ISMIR, 2007.

[14] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 669–674. University of Miami, 2011.

[15] Antti Eronen. *Signal Processing Methods for Audio Classification and Music Content Analysis*. PhD thesis, Tampere University of Technology, Tampere, Finland, 2009.

[16] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, 13(2):303–319, 2011.

[17] Ferdinand Fuhrmann. *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2012.

[18] Joe Futrelle and Stephen J. Downie. Interdisciplinary communities and research issues in music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 121–131. ISMIR, 2002.

[19] Robert O. Gjerdingen and David Perrott. Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*, 37(2):93–100, 2008.

[20] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Exploration Newsletter*, 11(1):10–18, 2009.

[21] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., the second edition, 2006.

[22] Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann. Lyric text mining in music mood classification. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 411–416. ISMIR, 2009.

[23] Carlos Nascimento Silla Jr., Celso A. A. Kaestner, and Alessandro L. Koerich. Automatic music genre classification using ensemble of classifiers. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 1687–1692. IEEE, 2007.

[24] Youngmoo Kim, Erik Schmidt, and Lloyd Emelle. Moodswings: A collaborative game for music mood label collection. In *Proceedings of the International Conference on Music Information Retrieval*, pages 231–236, 2008.

[25] Fang-Fei Kuo, Meng-Fen Chiang, Man-Kwan Shan, and Suh-Yin Lee. Emotion-based music recommendation by association discovery from film music. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 507–510. ACM, 2005.

[26] Cyril Laurier, Mohamed Sordo, Joan Serrà, and Perfecto Herrera. Music mood representations from social tags. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 381–386. ISMIR, 2009.

[27] Edith Law and Luis von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 1197–1206, 2009.

[28] Ming Li and Ronan Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 252–259. ISMIR, 2005.

[29] Tao Li, Ogihara Mitsunori, and George Tzanetakis, editors. *Music Data Mining*. CRC Press, 2012.

[30] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 282–289. ACM, 2003.

[31] Chao Liao, PatriciaP. Wang, and Yimin Zhang. Mining association patterns between music and video clips in professional mtv. In Benoit Huet, Alan Smeaton, Ketan Mayer-Patel, and Yannis Avrithis, editors, *Advances in Multimedia Modelling*, volume 5371 of *Lecture Notes in Computer Science*, pages 401–412. Springer Berlin Heidelberg, 2009.

[32] Stefaan Lippens, Jean-Pierre Martens, Tom De Mulder, and George Tzanetakis. A comparison of human and automatic musical genre classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 233–236, 2004.

[33] Hanna M. Lukashevich, Jakob Abeer, Christian Dittmar, and Holger Gromann. From multi-labeling to multi-domain-labeling: A novel two-dimensional approach to music genre classification. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 459–464. ISMIR, 2009.

[34] Brian McFee, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet. The million song dataset challenge. In *Proceedings of the 21st International Conference Companion on World Wide Web*, pages 909–916. ACM, 2012.

[35] Anders Meng and John Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 604–609. ISMIR, 2005.

[36] Steven R Ness, Anthony Theocharis, George Tzanetakis, and Luis Gustavo Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proceedings of the 17th ACM International Conference on Multimedia*, pages 705–708. ACM, 2009.

[37] Kerstin Neubarth, Izaro Goienetxea, Colin Johnson, and Darrell Conklin. Association mining of folk music genres and toponyms. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 7–12. ISMIR, 2012.

[38] Franois Pachet. Content management for electronic music distribution. *Communications of the ACM*, 46(4):71–75, 2003.

[39] Franois Pachet. Knowledge management and musical metadata. In Dabid G Schwartz, editor, *Encyclopedia of Knowledge Management*. Idea Group, 2005.

[40] Franois Pachet and Pierre Roy. Improving multi-label analysis of music titles: a large scale validation of the correction approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(2):335–343, 2009.

[41] Stuart Russel and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson education Inc., the third edition, 2010.

[42] Chris Sanden and John Z. Zhang. An empirical study of multi-label classifiers for music tag annotation. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 717–722. ISMIR, 2011.

[43] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[44] Alexander Schindler, Rudolf Mayer, and Andreas Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 469–474. ISMIR, 2012.

[45] Alexander Schindler and Andreas Rauber. Capturing the temporal domain in echonest features for improved classification effectiveness. In *Proceedings of the 10th International Workshop on Adaptive Multimedia Retrieval*, 2012.

[46] Carlos Nascimento Jr. Silla, Celso A. A. Kaestner, and Alessandro L. Koerich. The latin music database: Uma base de dados para a classificação automática de gêneros musicais. In *11th Brazilian Symposium on Computer Music*, pages 167–174. SBCM, 2007.

[47] Carlos Nascimento Jr. Silla, Celso A. A. Kaestner, and Alessandro L. Koerich. The latin music database. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 451–456. ISMIR, 2008.

[48] Mohamed Sordo, Òscar Celma, Martín Blech, and Enric Guaus. The quest for musical genres: Do the experts and the wisdom of crowds agree? In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 255–260. ISMIR, 2008.

[49] Derek Tingle, Kim Youngmoo E., and Douglas Turnbull. Exploring automatic music annotation with acoustically-objective tags. In *Proceedings of the international conference on Multimedia Information Retrieval*, pages 55–62. ACM, 2010.

[50] Konstantinos Trohidis, Grigorios Tsoumakas, George Kalliris, and Ioannis P. Vlahavas. Multi-label classification of music into emotions. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 325–330. ISMIR, 2008.

[51] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, February 2008.

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

[53] Fei Wang, Xin Wang, Bo Shao, Tao Li, and Mitsunori Ogihara. Tag integrated multi-label music style classification with hypergraph. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 363–368. ISMIR, 2009.

[54] Linxing Xiao, Aibo Tian, Wen Li, and Jie Zhou. Using a statistic model to capture the association between timber and perceived tempo. In *Proceedings of the International Society for Music Information Retrieval*, pages 659–662. ISMIR, 2008.

[55] Chao Zhen and Jieping Xu. Solely tag-based music genre classification. In *International Conference on Web Information Systems and Mining*, pages 20–24, 2010.