Music Information Retrieval

Automatic Genre Classification From Acoustic Features

Daniel Rönnow

ronnow@kth.se
0768 44 37 47
Vintrosagatan 56
12474 Bandhagen

Theodor Twetman

twetman@kth.se
070 20 27 140
Studentbacken 21/1405
11557 Stockholm

May 21, 2012

Course:	Bachelor Degree Project in Computer Science, DD143X
School:	Computer Science and Communications
University:	Royal Institute of Technology
Supervisor:	Anders Askenfeldt

Abstract

The aim of the study was to find a combination of machine learning algorithms and musical parameters which could automatically classify a large amount of music tracks into correct genres with high accuracy. To mimic a real musical situation we used the Million Song Dataset as it contains pre-analysed data on a wide variety of tracks. On the basis of previous studies and our evaluations of the available musical parameters a selection of four algorithms and four combinations of parameters were made. All these combinations of parameters were evaluated with each of the algorithms.

The best algorithm used with the two best combinations resulted in 49% and 51% accuracy respectively. Compared to some of the previous studies in this field our results are not outstanding, but we believe our results are more relevant in a real musical situation due to our choice of dataset, parameters and genres. When we evaluated the parameters we discovered that the they differentiated very little between the genres.

Even though our results show that our implementation is not good enough to use in a real application, it does not exclude the possibility of implementing an application for automatic classification of tracks into correct genres with high accuracy. The fact that the parameters do not differentiate much indicate that it might be a very extensive task to achieve the goal of high accuracy.

Sammanfattning

Syftet med den här studien var att hitta en kombination av maskininlärningsalgoritmer och musikaliska parametrar som automatiskt kan klassificera en stor mängd låtar med rätt genre med hög noggrannhet.
För att efterlikna en verklig musikalisk situation använde vi the Million Song Dataset eftersom den innehåller resultaten från musikaliska analyser av en stor mängd samtida låtar. På basis av tidigare studier och våra utvärderingar av de tillgängliga musikaliska parametrarna gjorde vi ett urval av fyra algoritmer och fyra kombinationer av parametrar. Samtliga kombinationer av parametrar utvärderades med var och en av algoritmerna.
Den bästa algoritmen resulterade i 49% respektive 51% noggrannhet när den användes tillsammans med de två bästa kombinationerna av parametrar. Jämfört med resultat från några av de tidigare studierna inom samma område är våra resultat inte enastående, men vi anser att våra resultat är mer relevanta i en verklig musikalisk situation på grund av vårt val av datamängd, parametrar och genrer. När vi utvärderade parametrarna upptäckte vi att de differentierade väldigt lite mellan genrerna.
Även om våra resultat visar att vår tillämpning inte är tillräckligt bra för att använda i ett riktigt program utesluter det inte möjligheten att skapa en applikation för automatisk genreklassificering av spår med hög noggrannhet. Det faktum att parametrarna inte skiljer sig åt mellan genrer indikerar dock att det kan vara en mycket omfattande uppgift att uppnå målet om hög noggrannhet.

Statement of collaboration

This study was divided into two parts; the execution of the project and the writing of the report.

The execution of the project
Twetman wrote the software for reclassifying the tracks and calculating which genres were to be used. He also wrote the parser from the file format of the Million Song Dataset to the file format of WEKA. Rönnow wrote software for the manually finding out which features differentiate the most when grouped by genre. Most of the evaluations of the different combinations of features and algorithms we did together.

The report
As we were new to the subject of Music Information Retrieval we began by doing some background research and by writing the introduction and the problem statement together, as part of the project description. Rönnow then wrote about the previous studies, while Twetman wrote about the Million Song Dataset. The remaining part of the Method was divided equally between the two of us. Rönnow compiled the Results and together we discussed what to bring up in the Discussion. Rönnow then wrote it while Twetman, at the same time, edited and corrected the same part. We wrote the Conclusions and the Abstract together.

Contents

1 Introduction

Since music on its own is not searchable in an easy manner, music services providing large collections of music, such as Spotify, must assign searchable tags to every track (a specific song by a specific artist) in order to make them possible to find. This type of tags are called metadata. The metadata often includes information provided by the record company such as the artists name, the title of the track and the name of the album on which the track was released. But the metadata may also include tags describing the music such as genre, musical influences and mood.
Basically there are three ways to tag tracks with the latter form of metadata:

- Manually by an expert
- Manually by any user
- Automatically from an acoustic analysis

Letting a group of experts manually tag genres to large amounts of tracks is very time consuming and thus very costly but the tags will probably be correct. Letting any user do the same will be cheap but might lead to contradictions since all people do not have the exact same perception of the same genre. Automatic genre tagging from an acoustic analysis takes the best from the above mentioned methods as the tagging is done in a consistent way and being cheap to run on large collections of tracks. The problem with this approach is how well a machine can be set to determine the genre of a track.

1.1 Previous Studies

Many Music Information Retrieval (MIR) studies have been made on the subject of automatic genre classification. Each study with a different approach as to which acoustic features and which algorithms to base the classification upon and which tracks to use for the evaluation of the results. Because of this, the outcomes differ substantially.
A common issue in previous studies is the selection of which acoustic features to use to achieve the most successful result. Both low-level features and high-level symbolic features have been used to accomplish automatic genre classification[1]. Low-level features describe the characteristics of the audio signal and may estimate how a human ear perceives the music while the high-level features estimate the musical elements such as pitch and tempo.
One commonly used low-level feature is a group of Mel-Frequency Cepstral Coefficients (MFCCs). Each MFCC is a set of coefficients describing a short segment of an audio sample, typically 20 to 30 ms long[6]. Those coefficients are derived from the Mel-Frequency Cepstral which approximates the human auditory system's response. MFCC is therefore often used in speech

recognition systems[6]. Multiple MFCCs can be used to represent a whole track.

Another common issue in the previous studies is the large amount of musical genres and subgenres which makes it difficult to classify a track with the exact genre. For example "rock" and "hard rock" have a very similar sound[6] but they differ sufficiently to be classified as separate genres. Therefore, to get more accurate results, most of the related studies have only used a certain small group of basic genres[6].

In a study by Tzanetakis and Cook[16] four different low-level parameters were used: Fast Fourier Transform (FFT), MPEG filterbank analysis, Linear Predictive Coding (LCP) and MFCC[16]. With these features, Tzanetakis and Cook used a supervised machine learning algorithm with a Gaussian Mixture Model (GMM) to classify the genres of the tracks[16]. The tracks were classified into three genres: classical, modern (essentially rock and pop) and jazz[16]. By this approach Tzanetakis and Cook got a 75% accuracy on classifying the tracks. They concluded that a jazz piece with vocals might be easy for a machine learning algorithm to not classify as a classical track due to the fact that the characteristics of the genres differ. For example; vocals and guitars are often used in jazz pieces but hardly ever in classical pieces. They also stated the fact that they did not include high-level features, such as beat, which they believe might have improved the results[16].

Salamon et al.[10] did a comparison between high-level and low-level features and a combination of those was evaluated. The pitch, vibrato and duration were used as high-level features, and the MFCC was used as a low-level feature[10]. The comparison was made with four different machine learning algorithms and the results were evaluated by using two different sets of tracks. The algorithms used were; Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbours (KNN) and Bayesian Network (BN)[10]. One of the datasets used was a 500 track dataset selected over the five genres; Opera, Pop, Flamenco, Jazz with vocals and instrumental Jazz[10]. The second dataset used was the GTZAN set[10], which was created and used by Tzanetakis in a later study than the one mentioned above[17].
The evaluation resulted in an overall accuracy of 95% with the combined features using the 500 track set[10]. With the GTZAN set an overall accuracy of about 80% was achieved with all algorithms except when using the KNN algorithm which resulted in 70% accuracy[10].
The results of the individual features were overall the same when using the 500 track set, about 90% accuracy. On the GTZAN set the high-level features outrun the low-level MFCC with an average of 65% against an average of 55% for the MFCC[10].

A study on genre classification using the Million Song Dataset (the MSD is explained in more detail below) was published by Liang et al[8]. Different combinations of the features available in MSD were tested in the study, including timbre (which is unique for MSD), tempo, loudness and a bag-of-words feature (derived from the lyrics)[8]. It was also the authors intention to explore unused algorithms in genre classification research, which yielded in the Baum-Welsh algorithm (BW) in comparison and combination with a spectral algorithm (SP) to learn a Hidden Markov Model for each one of the ten genres used[8]. The ten different genres used in the classification were; Classical, Metal, Hip hop, Dance, Jazz, Folk, Soul, Rock/Indie, Pop and Classic Rock/Pop[8]. The overall best combination of features and algorithms included the BW algorithm, the SP algorithm, the loudness, the tempo, and the lyrics which resulted in 39% accuracy[8]. Between the different genres the result was widely spread. The classical genre achieved the best result with 78% accuracy while the "Classic Rock/Pop" only got 16% accuracy[8]. In this study, Liang et al. used a large dataset with all tracks in MSD having the bag-of-words, the timbre, the loudness and the tempo features, 156 thousand songs[8]. This set had to be balanced to obtain good learning and evaluation procedures. The use of a large dataset of real tracks was one of the motivations of this study as Liang et al. mean that the GTZAN dataset and other datasets often used in state-of-the-art MIR are too small and too narrow and thus far away from use in practical application[8].

In summary, all of the above mentioned studies used quite similar approaches. All used the MFCC feature, except for Liang et al. who used the corresponding timbre feature from MSD. Additional high-level features were used in some cases to improve the results in combination with the low-level feature as in the study by Salamon et al[10]. in which the pitch was used and as in the study by Liang et al.[8] in which the lyrics, the tempo and the loudness were used. All mentioned studies used supervised machine learning to classify the genre of the tracks. One main difference is that each study chose to classify the tracks into their own unique set of genres although all of them used a quite small number of genres. The studies by Salamon et al.[10] and Tzanetakis and Cook[16] used five and three genres respectively which were very different from each other and achieved good results. Liang et al.[8] used ten genres which to some extent sound very similar and achieved much lower results.
The use of different datasets to teach the algorithms make the results hard to compare. Salamon et al.[10] and Tzanetakis and Cook[16] used small datasets with good results, while Liang et al.[8] used a large dataset, which they mean reflects real music better than most other datasets used in MIR

and got a poorer result.

1.2 Problem Statement

The purpose of this study was to examine the possibility to automatically classify tracks into genres, solely by using the information derived from an acoustic analysis, to such extent that it would be useful in a real world application. To be able to compete with manual classification and to be useful in practice an automatic classifier should be able to classify tracks into at least as many different genres as used in the studies mentioned above, and it should be very accurate.

The aims of this study are (1) to try to find a combination of algorithms and musical parameters which makes it possible to automatically classify tracks into correct genres using a large dataset, and (2) evaluate the possibility of using it in an application which demands high accuracy.

1.3 Hypothesis

As a hypothesis, we believe that the low-level feature MFCC, or the corresponding timbre feature, will be a good basis for the classification since it is a feature most previous studies have used successfully, as well as being commonly used in speech recognition. But as proven in the studies above not only low-level features are good to use. Adding high-level features like tempo, loudness, key and pitch in combination with the low-level feature might improve the results. We believe that those features may vary between different genres. To accomplish the classification we believe in using machine learning as it fits the purpose well of analysing large amounts of data to find parallels based on a selected parameter, the genre. Among the above used algorithms there are some that performed better than others and we believe that focusing on those, i.e. the algorithms SVM, RF, BN and the KNN, might yield in good results. Considering the dataset it seems crucial to use an evenly spread set of tracks in the learning process to get the best and most reliable results.

2 Method

It is our intention to evaluate a number of combinations of algorithms and parameters. This will be done by testing different algorithms with different combinations of parameters to see which combination will yield the best result. In this section we will declare which dataset, which genres and which algorithms to base the classification upon, and how to validate the results.

2.1 Dataset

An essential problem in the previous studies is the choice of dataset in order to achieve good results which are reliable and reflects the reality. This is one of the essential problems in this study as well.

The dataset we have chosen in this study is the Million Song Dataset (MSD). This set has been selected because it contains a large amount of preanalysed tracks, described in more detail in the next section. Another reason to use this dataset is the fact that it is a part of the large database The Echo Nest which might be of use in a real application.

2.1.1 The Million Song Dataset

The MSD and the MSD subset are two freely available datasets containing metadata and results from audio analysis of one million and ten thousand contemporary music tracks respectively[2]. The tracks are analysed in a manner that simulate how people perceive the music[5].

The main purposes of the datasets are:

- to encourage research on algorithms that scale to commercial sizes;
- to provide a reference dataset for evaluating research;
- as a shortcut alternative to creating a large dataset with The Echo Nest's API;
- to help new researchers get started in the MIR field.

The MSD and the subset are derived from The Echo Nest database which contain the same metadata and musical analysis data as the two sets, but for about 30 million tracks[12]. The Echo Nest provides two APIs; one for their data to be used in third party applications and one for letting developers analyse music and getting a result in the same format as in the datasets. This is an important reason for choosing the MSD or the subset. To successfully use an automatic classifier in an application you need data for almost all available tracks. The fact that the sets are derived from The Echo Nest makes the evaluation more reliable and it is interesting to see whether or not a part of The Echo Nest could be used in an application since it is easy to access.

More precisely the MSD contains 1 000 000 tracks by 44 745 unique artists and the subset contains 10 000 tracks by 3 888 unique artists. For each track there is a set of metadata such as name of the artist, the title of the track, the recording year as well as acoustic tags describing the music[2]. There

are three main acoustic features: Pitch, Timbre and Loudness.

Each track is divided into a set of segments which are relatively uniform in timbre and harmony[5]. The segments are most often shorter than one second, typically in the span of 100 ms to 500 ms[15]. The three features above are provided for every segment of each track[2]. Furthermore there are a lot of other acoustic features such as tempo, energy and overall loudness. The most important features, from our point of view, are explained in detail in the next section. See appendix A for a complete list of the available information for each song in the datasets.

The MSD includes features similar to the features mentioned in the previous studies as well as a number of other features. By using this set we have easy access all to of these features and more combinations may then be evaluated. The fact that both the MSD and the MSD subset include large collections of tracks make the sets more usable in this study than the sets of around 1000 tracks used in previous studies. The results are more reliable the larger the dataset but since we have limited resources in terms of time and computer power we have chosen to work with the MSD subset.

2.1.2 Musical representation in MSD and feature selection

The selection of features is a critical choice in this study. It is important that the features differ between the genres for the learning algorithms to draw correct conclusions. Our choice of features is derived from both the previous studies and from manual testing of which features differentiate the most when grouped by genre.

Timbre
Timbre is a feature similar to MFCC, describing the musical texture[3]. Each segment includes a set of 12 separate timbre values[5], eg. the first value represent the average loudness and the second value emphasizes the brightness of the segment [5]. Each of the 12 values represents a high level abstraction of the spectral surface. They are ordered by their degree of importance[5]. Since the segments that the timbre feature describe are much longer than the segments that MFCC describe, a much greater part of the track can be described with timbre using the same amount of data compared to MFCC. Since none of the previous studies have mentioned how they used the MFCC or timbre feature, it is hard to know what a good usage of this feature is. In this study the feature will be used as the mean value of each timbre over the whole track, combined with the standard deviation of each mean. This representation might not be as good as using each timbre, which can not be used due to the limitations of this study, but it gives a good abstraction of the value.

Tempo
In musical terminology, tempo is the speed or pace of a given track measured in beats per minute (BPM)[11]. As the tempo varies during the track, the tempo feature is an overall estimation of the track's tempo[5].

Key
In musical terminology, the term key can be used in many different ways. In this case the meaning of the term key is the tonic triad, the final point of rest of a track[5]. The key feature is an overall estimation of the track's key[5].

Loudness
In musical terminology, loudness is the "quality of a sound that is the primary psychological correlate of physical strength"[5]. The loudness feature is an overall estimation of the track's loudness in decibel (dB).

Pitch
Each segment includes a set of 12 separate pitch values, one for each of the 12 pitch classes C, C#, D to B, in the range 0 to 1 representing the relative dominance of every pitch class. A single tone will have one of the 12 values close to one and the rest close to zero, a chord will have a couple of the values close to one and the rest close to zero while a noisy sounds will have all 12 values close to one[5]. A good usage of this feature is hard to choose because of the amount of data it generates per track. In this study the feature will be used as the mean value of each pitch over the whole track, combined with the standard deviation of each mean. This gives a clear representation of which pitches are used the most during the whole track.

Genre
Tracks in MSD are not exactly classified to a genre. The classification is an estimation of genres connected to the track and how frequent the track is mentioned to be classified as that genre. The genre with the highest frequency will then probably genre best describing that track[14]. This is the only metadata used and will solely be used to train and evaluate the combinations.

2.1.3 Selection of genres and tracks

The selection of genres and tracks to classify is not evident. The choice of genres will be affected by the fact that the classification should be of possible use in practical application. The genres chosen will therefore have to be common ones on a high level of abstraction. The selection of tracks have to be evenly spread across the chosen genres so the possibility of unbalanced

test results are eliminated.

Because of how the genres are classified in the MSD some estimations have to be done to get an evenly spread set of tracks. This means that a classification of tracks classified as low level genres, such as "Classical Rock", needs to be included in a high level classification, in this case "Rock". This can be done by classifying on the last word in the low level genre, as it is a noun and all previous words are adjectives. By choosing the six most common genres in the MSD subset we got the following genres:

- Rock
- Pop
- Jazz
- Blues
- Hip Hop
- Electronic

Each of these genres consists of about 600 tracks except for the genre "Rock" which was decreased to 600 randomly chosen tracks. . Some tracks were also classified as a contradictory genre, e.g. "Pop Rock". These tracks were deleted from the set since it would confuse the learning process and may lead to misclassified tracks.

2.2 Supervised Machine Learning

Machine learning is used in many areas. Its purpose is to learn and draw conclusions like a human. In that way a machine can make correct assumptions, do analyses and find relationships between features only by looking at previously known data. On a given input it should be able to determine the output corresponding to the input[7].
The algorithms are trained with a dataset where each instance in the input set contains a set of attributes. If each instance in this teaching dataset also includes the output attribute to which the instance corresponds, the algorithm is a supervised learning algorithm because the output value is known to the input attributes[7].

2.2.1 WEKA Data Mining Software

WEKA, which is an abbreviation for the Waikato Environment for Knowledge Analysis, is an open source toolbox and a framework for learning algorithms. It provides easy access to state-of-the-art techniques in machine learning as well as it is meant to be easy for users to add new algorithms

to the software[4]. The software is written in Java and can therefore be integrated and used like a Java library. It also includes a rich graphical environment with methods for validation of results and visualisation of the results and the input data.

The above stated facts are the reasons why we chose to work with WEKA. As WEKA implements most of the common algorithms, including the ones we want to test, it was a natural choice. The easy to use graphical interface combined with the possibility of using WEKA as Java library will let us experiment in WEKA but keeping the door open for making external programs using it as a library.

2.2.2 Selection of algorithms

The prediction of which algorithms are good to use is hard, partly because of the amount of algorithms, but also because they should make good combinations with the features. All mentioned studies used different algorithms which indicates the lack of certainty of which algorithm to use.

The algorithms used will be some of the algorithms used in the previous studies which yielded good results. It is interesting to examine these algorithms since they only have been used on smaller sets of tracks in which the genres differ a lot more than the ones used in this study. By using previously used algorithms conclusions may also be made about how the selection of features stands in relationship to the results of the previous studies. The algorithms chosen were:

- Support Vector Machine
- Random Forest
- Bayesian Network
- K-Nearest Neighbours

2.2.3 Validation process

The validation process of the combinations of features and algorithms will be done by using K-fold cross validation. This essentially means that the set of tracks is divided into K equally large subsets. The subsets are then used to test and teach the learning machine K times, one time per subset. This means that in a 10-fold cross validation we get 10 tests in one and the results from each subset is combined into a final result[9].

This is a good way to get variations and do more tests on the same data. It will also decrease the impact on the validation for datasets of tracks whose

features group well between genres, since they are easy to classify into the correct genres. This is essential for the evaluation since the possibility of using it on any kind of music is examined. The fact that it is implemented into the WEKA software makes it easy to use.

2.3 Chosen Combinations

The combinations of which features to test are mainly derived from the results of previous studies. The combinations are also influenced of what may differ between the genres. Some manual examinations of which features differ the most between genres have been made to see whether the feature may be useful or not. These examinations are represented as pictures in Appendix B.
The combinations to be examined (also shown in Table 1 below) are:

1. Mean values of timbre and standard deviations of timbre across all segments of the whole track
2. Mean values of timbre and standard deviations of timbre across all segments of the whole track, tempo, key and loudness
3. Mean values of timbre, standard deviations of timbre, mean values of pitch and standard deviations of pitch across all segments of the whole track and tempo, key and loudness
4. Tempo, key and loudness.

Feature/Combination	1	2	3	4
Mean values of timbre	X	X	X	
Standard deviations of timbre	X	X	X	
Mean values of pitch			X	
Standard deviations of pitch			X	
Tempo		X	X	X
Key		X	X	X
Loudness		X	X	X

Table 1: The combinations of features evaluated in the study.

All of these combinations will be tested with the chosen algorithms described in section 2.2.2 and validated according to the section 2.2.3.

3 Results

The best result we got was with combination number 3 with the algorithm "Support Vector Machine", which classified 51% of the tracks into correct

genre. The best overall algorithm was the SVM, which classified about 50% correct on each combination except for combination number 4. The other algorithms classified about 40% of the tracks correctly on all combinations, except for combination number 4 which achieved the lowest results. The highest result on combination number 4 was 31% with the algorithm "Bayesian Network".

All results can be retrieved from the following tables. Appendix C contains a confusion matrix for the classification with the SVM on combination number 3.

Algorithm	Result(%)
Support Vector Machine	48
Random Forest	43
Bayesian Network	43
K-Nearest Neighbours	40

Table 2: Accuracy results, in percent, of the evaluation of combination 1 including the mean value of each timbre and the standard deviation of each of the means features.

Algorithm	Result(%)
Support Vector Machine	49
Random Forest	41
Bayesian Network	43
K-Nearest Neighbours	35

Table 3: Accuracy results, in percent, of the evaluation of combination 2 including the mean value of each timbre, the standard deviation of each of the means, the tempo, the key and the loudness features.

Algorithm	Result(%)
Support Vector Machine	51
Random Forest	41
Bayesian Network	45
K-Nearest Neighbours	37

Table 4: Accuracy results, in percent, of the evaluation of combination 3 including the mean value of each timbre, the standard deviation of each of those means, the mean value of pitch, the standard deviation of each of those means, the tempo, the key and the loudness features.

Algorithm	Result(%)
Support Vector Machine	29
Random Forest	24
Bayesian Network	31
K-Nearest Neighbours	23

Table 5: Accuracy results, in percent, of the evaluation of combination 4 including the tempo, the key and the loudness features.

Additional evaluations were done with different usage of some features, including timbre and pitch. The usage of timbre was changed to be represented as the Riemann sum of each timbre value. A combination of the mean and the Riemann sum was also tested, both of the new representations without any improvement of the result. An alternativ usage of the pitch feature was also tested. Instead of using the mean of the twelve pitch values the index value of the pitch with the largest mean value was used. This usage of the pitch did not improvement of the results.

4 Discussion

As seen in Table 6 our best result (51%) is in the same region as the result of Liang et al. who obtained 39% and who also used the MSD[8]. Neither of our results are similar to the results obtained by Tzanetakis and Cook or by Salamon et al., 75% [16] and 95% [10] respectively, even though the features and the algorithms used in this study are similar to the ones used by Salamon et al [10].
The only two differences between our study and the one by Salamon et al.[10] are which dataset and genres used, which was on purpose. A reason to why our results are not as good as the results of Salamon et al.[10] could be the large dataset, which is a lot bigger than the ones used in that study, and that our dataset is not chosen to consist of tracks whose genres are of great difference and hence more easily classified.

Study	#Tracks	#Genre	#Conflicting genres	Result(%)
This study	3500	6	4	51
Tzanetakis	NA	3	None	75
Salamon et al.	500	5	2	95
Liang et al.	156000	10	6	39

Table 6: Comparison of the best results from this study and the previous studies. #Tracks is the number of tracks used in the study, #Genre is the number of genres, #Conflicting genres is the number of conflicting genres, Result indicates the percentage of tracks that were correctly classified.

As the results show the classification only achieved 51% accuracy at best, which we believe is a rather poor, but a reliable result. Using this classifier in practice in an application would lead to a lot of misclassified tracks. The main reasons for the not so successful results are:

- The large dataset
- The genres chosen - some of them sound similar
- The selection and usage of the features
- The WEKA software and the usage of the algorithms

The choice of dataset, genres, features, software and algorithms is, however, well motivated which contributes to the reliability of the results. In particular, the choice of genres and the large dataset reflect realistic conditions for automatic classification of music tracks.

As previously mentioned it is an essential problem choosing the dataset to obtain a reliable classification. The dataset in this study was chosen because of the connection to a real music situation and the amount of tracks, which we believe give results closer to the reality. The purpose was not to evaluate how good an automatic classifier could be when running on a certain dataset. The purpose was to be able to classify any kind of dataset which might not have been the purpose of other studies. The results partly confirm what Liang et al.[8] stated in their study. A reason to why Salamon et al.[10] and Tzanetakis and Cook[16] achieved good results might be the fact that they chose their own composed datasets which might have been chosen explicitly for gaining good results and not for reliability.

The fact that the dataset used in this study is larger than the ones used in other studies is not the only possible reason why our results are not as good as the results of Salamon et al[10] and Tzanetakis and Cook[16]. The genres used in this set are more similar to the ones used by Liang et al.[8] but

not at all similar to the ones used in the other studies mentioned. Table 6 show that both we and Liang et al.[8] used more conflicting genres than non conflicting, in opposite to Salamon et al[10] and Tzanetakis and Cook[16]. Genres chosen widely spread and not at all similar, i.e. non conflicting, will make the classification easier since the musical characteristics of such genres will differ more.
Some of the genres used in this study have similar sound which makes it hard for the automatic classifier to correctly classify those tracks. The evaluation process showed the percentage of each genre that was classified as another. In Table 9 in Appendix C it clearly shows that many of the genres sound similar to a machine. We can also see that some genres, for example Hip Hop and Electronic, have a much more differentiated sound as tracks of those genres are not confused with each other very often. The pictures in Appendix B is taken from the WEKA software and shows the spread of some features in relationship to the genres, the rest of the features had similar outcome. These pictures shows that the features do not differ much between the genres, although they differed the most of the examined features. This also indicates that the genres sound similar.

Two other factors that may contribute to why the genres may be a possible reason for the low accuracy are the genre classification in MSD and the way we reclassified tracks to more abstract genres. The classification in MSD might not be completely accurate since it is solely based on the frequency of which a genre is used in context with the track. The genre with the highest frequency is the one used the most to describe a certain track but that does not guarantee that it is the best genre to describe that track. The fact that we reclassified some tracks makes the genre of the tracks even more doubtful. If a more reliable genre classification of the tracks were available the evaluation would have been more reliable, since both the learning process and the validation process would have gained from it.

If the genres is a reason to why we obtained lower results in comparison to some of the other studies, adding more genres would probably make it even worse. Indications of this can be seen in Table 6. We used 6 different genres while the similar study of Liang et al[8]. used 10 different genres and we obtained the better result of 51% compared to 39%. We chose to use only a few but similar genres to simulate a real world application. If the classifier should be possible to use in an application it would have to be able to classify tracks into more genres and subgenres. Our examination indicates that such an implementation is very hard to achieve due to the small differences between the genres. One major problem may lie in the genre classification itself, the definitions of the genres are too vague and therefore too many of the genres overlap.

Our usage of the features might be another contribution to the low results. Even though it seems that our usage of the features was fairly good in comparison with the study by Liang et al[8].
Our usage of the timbre feature was probably the most unreliable since the standard deviation of each timbre value was almost as large as the value itself. If the timbre feature would have been used in a way that describe each segment, better than the mean value, it would probably have increased the accuracy. We tried to use the Riemann sum which, because of the constant value of each timbre over each segment, will be equal to the integration of the timbre curve. This did not improve the results though, probably because of the sum of the values of two curves may be equal although the curves look different.
One thing to notice is that the results of the "Support Vector Machine" increased when using combinations with more features. As seen in Tables 2, 3 and 4 the results increased from 48% to 49% to 51%. By adding a parameter which does not contribute to the separation of genres, the results should at best stay unchanged if not decrease. Therefore the parameters added seems to be useful. In contrast to the results of "Support Vector Machine" the results of "Random Forest" decreased by adding features. This might indicate that the selection of features depends on which algorithm to use. However the increase and decrease of the results are very small. It might therefore be an outcome of the particular dataset we used and not a reliable indication.

The other feature on which the usage could be improved is the pitch feature. If this feature also could be improved in a similar way as timbre, it would describe the track in more detail. This would probably be better to use than the mean value. The standard deviation for this feature was also very high and indicates that the usage of this feature was not optimal.

One thing not yet discussed is the usage of external software. Using the WEKA software was a delight, but it might be used in a better manner. By using the possibility of implementing algorithms on our own, which could be optimised for our purpose, might have improved the results. The WEKA software did offer the possibility to change some parameters of existing algorithms but the lack of experience with such algorithms and the lack of time made us run the tests with standard settings.

5 Conclusions

With the chosen combinations the classifier managed to achieve at best about 50% accuracy. The best combination was number 3, including all chosen features, using the Support Vector Machine algorithm. We used approximately the same algorithms and combinations of features as in some

previous studies, but with a different dataset and classified into other genres, yielding in partly inferior results.
These results may have been caused by the selection and usage of the musical features since their values were too similar between each genre to obtain good results.
Previous studies in the same area of research, also using the Million Song Dataset, have obtained results in the same region as ours. This could indicate that the Million Song Dataset is an unreliable source of data for usage in this context.

The fact that some of the genres we chose sound similar was probably one of the causes for the not so good results. This was on purpose as it would contribute to the reliability of this study since many genres and subgenres, with similar sound, would have to be used in real application for it to be useful.

Our classification turned out to be not as accurate as an application would demand. Finding other musical characteristics that differ more between the genres than the ones used in this study may be difficult, but it would definitely improve the results.

As an answer to the questions in the problem statement; the combinations of features, dataset and algorithms evaluated in this study are probably not the combinations to use in an automatic genre classification application as the classifier did not achieve high enough accuracy. This does not mean that the features used in this study are not the ones to use, only that the way we used them might not be optimal. The same reasoning applies to the use of algorithms.

Our results do not exclude the possibility of implementing an application of automatic classification of tracks into correct genres with high accuracy as there could exist better approaches, but the one we chose did not achieve the goal of high accuracy. Our results indicate that the genres overlap and that the parameters we evaluated did not differentiate much between the genres. By adding more genres the overlapping will probably increase making it even harder to distinguish one genre from the other. Taking all this into consideration it might be a very extensive task to achieve the goal of high accuracy.

6 References

[1] Aucouturier, Jean-Julien & Pachet, Francois, *Representing Musical Genre: A sate of the Art,* France, Paris, SONY Computer Science Labratory, 2003.
URL:http://jjtok.io/papers/JNMR-2003.pdf (Viewed: 2012-02-25)

[2] Bertin-Mahieux, Thierry & Ellis, Daniel P.W. & Whitman, Brian & Lamere, Paul. *The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011),* 2011,
URL:http://www.columbia.edu/~tb2332/Papers/ismir11.pdf(Viewed: 2012-03-03)

[3] Field List, The Million Song Dataset
URL:http://labrosa.ee.columbia.edu/millionsong/pages/field-list (Viewed: 2012-03-1)

[4] Hall, Mark & Frank, Eibe & Holmes, Geoffrey & Pfahringer, Holmes & Reutemann, Peter & Witten, Ian H. *The WEKA Data Mining Software: An Update, 2009*, SIGKDD Explorations, Volume 11, Issue 1.
URL:http://www.cs.waikato.ac.nz/~eibe/pubs/weka_update.pdf (Viewed: 2012-03-01)

[5] Jehan, Tristan & DesRoches, David, Analyzer Documentation, 2011,
URL:http://docs.echonest.com.s3-website-us-east-1.amazonaws.com/_static/AnalyzeDocumentation.pdf (Viewed: 2012-02-26)

[6] Klautau, Aldebaro, The MFCC, 2005
URL:http://www.cic.unb.br/~lamar/te073/Aulas/mfcc.pdf (viewed: 2012-02- 29)

[7] Konsiantis, Sotiris, *Supervised Machine Learning: A Review of Classification Techniques*, Peloponnese, Universitt of Peloponnese, 2007.
URL:http://www.informatica.si/pdf/31-3/11_kotsiantis%20-%20supervised%20machine%20learning%20-%20a%20review%20of...pdf (Viewed 2012-04-01)

[8] Liang, Dawen & Gu, Haijie & O'Conner, Brendan, *Music Genre Classification with the Million Song Dataset*, Pittsburgh, Carnegie Mellon University, 2011.

URL:http://www.cs.cmu.edu/~music/dawenl/files/FINAL.pdf (Viewed 2012-02-28)

[9] Refaeilzadeh, Payam & Tang, Lei & Liu, Huan, *Cross-Validation, Arizona, Arizona State University,* 2008. URL: http://www.public.asu.edu/~ltang9/papers/ency-cross-validation.pdf(Viewed 2012-04-01)

[10] Salamon, Justin & Rocha, Bruno & Gomez, Emilia, *Musical Genre Classification Using Melody Features Extracted From Polyphonic Music Signals*, Barcelona, Universitat Pompeu Fabra, 2012.
URL:http://mtg.upf.edu/system/files/publications/SalamonRochaGomezICASSP2012.pdf (Viewed 2012-02-25)

[11] Tempo, Nationalencyklopedin,
URL:http://www.ne.se/lang/tempo (Viewed: 2012-03-8)

[12] The Echo Nest,
URL:http://the.echonest.com/ (Viewed: 2012-02-29)

[13] The Echo Nest Developer Blog, *Danceability and Energy: Introducing Echo Nest Attributes*,
URL:http://blog.developer.echonest.com/ (Viewed: 2012-02-20)

[14] Top_terms, The Echo Nest,
URL:http://developer.echonest.com/docs/v4/artist.html#top-terms (Viewed: 2012-04-04)

[15] Tingle, Derek & Kim, Youngmoo E. & Turnbull, Douglas, *Exploring Automatic Music Annotation with "Acoustically Objective" Tags*, Swarthmore, Swarthmore College, 2010.
URL:http://web.cs.swarthmore.edu/~turnbull/Papers/Tingle_Autotag_MIR10.pdf (viewed: 2012-03-01)

[16] Tzanetakis, George & Cook Perry, *Audio Information Retrieval Tools*, Princeton, Princeton University, 2000.
URL:http://www.ee.columbia.edu/~dpwe/papers/TzanC00-airtools.pdf (Viewed 2012-02-25)

[17] Tzanetakis, George & Essl, Georg & Cook, Perry, *Automatic Musical Genre Classification Of Audio Signals*, Princeton, Princeton University, 2001.
URL:http://ismir2001.ismir.net/pdf/tzanetakis.pdf(Viewed 2012-04-03)

A The Million Song Dataset Field List

Below is a list of all fields available in the MSD, the MSD subset and The Echo Nest.

Field name	Description
analysis sample rate	sample rate of the audio used
artist 7digitalid	ID from 7digital.com or -1
artist familiarity	algorithmic estimation
artist hotttnesss	algorithmic estimation
artist id	Echo Nest ID
artist latitude	latitude
artist location	location name
artist longitude	longitude
artist mbid	ID from musicbrainz.org
artist mbtags	tags from musicbrainz.org
artist mbtags count	tag counts for musicbrainz tags
artist name	artist name
artist playmeid	ID from playme.com, or -1
artist terms	Echo Nest tags
artist terms freq	Echo Nest tags freqs
artist terms weight	Echo Nest tags weight
audio md5	audio hash code
bars confidence	confidence measure
bars start	beginning of bars, usually on a beat
beats confidence	confidence measure
beats start	result of beat tracking
danceability	algorithmic estimation
duration	in seconds
end of fade in	seconds at the beginning of the song
energy	energy from listener point of view
key	key the song is in
key confidence	confidence measure
loudness	overall loudness in dB
mode	major or minor
mode confidence	confidence measure
release	album name
release 7digitalid	ID from 7digital.com or -1
sections confidence	confidence measure
sections start	largest grouping in a song, e.g. verse
segments confidence	confidence measure
segments loudness max	max dB value
segments loudness max time	time of max dB value, i.e. end of attack

segments loudness max start	dB value at onset
segments pitches	chroma feature, one value per note
segments start	musical events, note onsets
segments timbre	texture features (MFCC+PCA-like)
similar artists	Echo Nest artist IDs (sim. algo. unpublished)
song hotttnesss	algorithmic estimation
song id	Echo Nest song ID
start of fade out	time in sec
tatums confidence	confidence measure
tatums start	smallest rhythmic element
tempo	estimated tempo in BPM
time signature	estimate of number of beats per bar, e.g. 4
time signature confidence	confidence measure
title	song title
track id	Echo Nest track ID
track 7digitalid	ID from 7digital.com or -1
year	song release year from MusicBrainz or 0

Table 7: Complete Field List in the Million Song Dataset[3]

B Feature grouping by genre in WEKA

The figures describes the distribution of a feature grouped by genre. Each row represent a genre and each cross represent the value for one song of that genre.

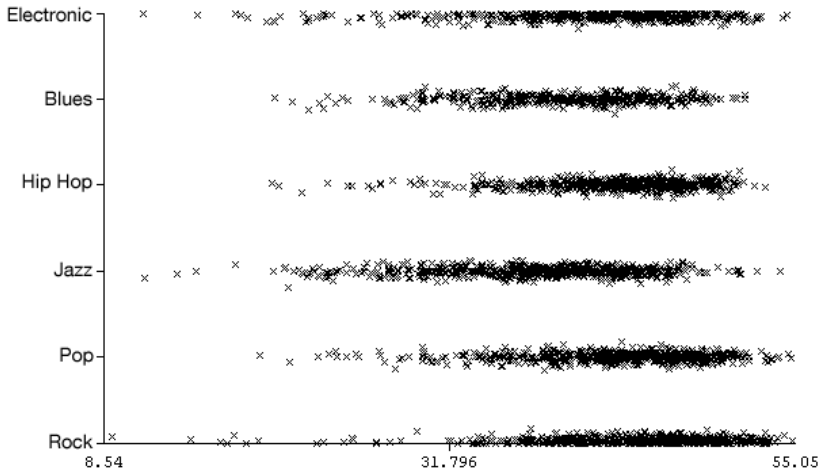

Figure 1: Describing the distribution of the first mean timbre value grouped by genre according to Table 8

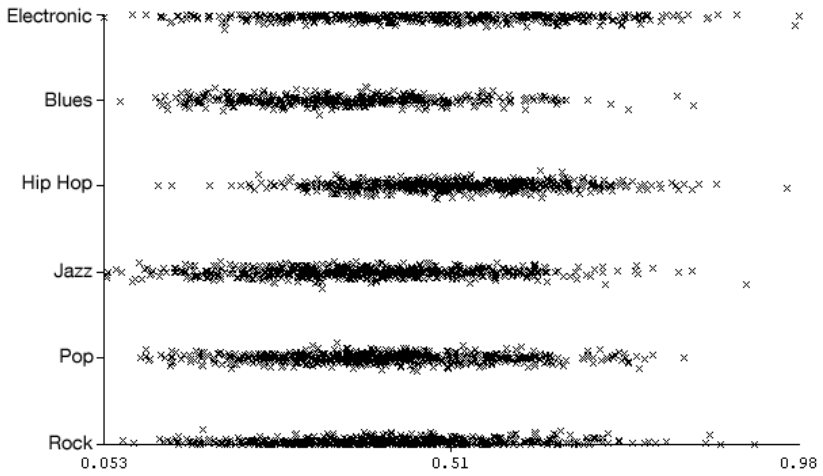

Figure 2: Describing the distribution of the first mean pitch value grouped by genre according to Table 8

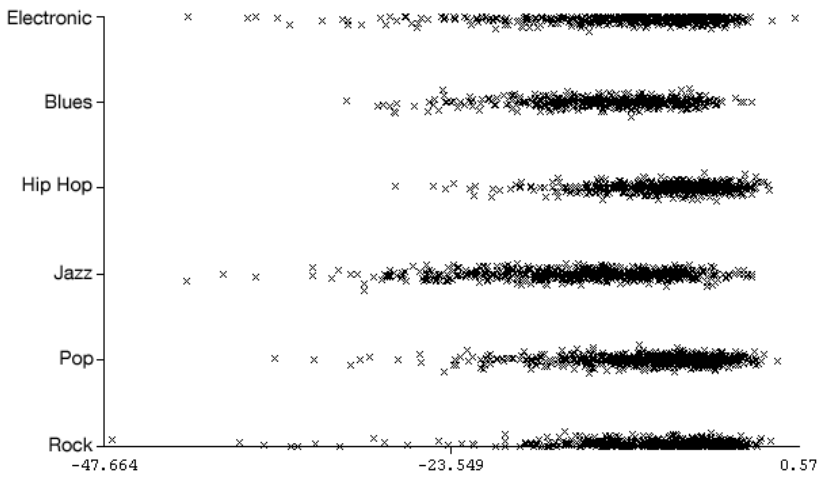

Figure 3: Describing the distribution of the loudness value grouped by genre according to Table 8

C A confusion matrix

A confusion matrix shows how many of the classified tracks of a genre(rows) that were classified as each of the genres(columns) in percent. The matrix shown below is the confusion matrix derived from the most accurate combination, combination number 3 with Support Vector Machine, which obtained 51% accuracy.

Genres	Rock	Pop	Jazz	Blues	Hip Hop	Electronic
Rock	**55**	21	8	6	4	6
Pop	20	**42**	15	10	5	9
Jazz	8	16	**55**	5	10	7
Blues	6	12	4	**72**	1	6
Hip Hop	16	14	22	3	**44**	0
Electronic	16	17	19	10	2	**35**

Table 8: Confusion matrix for Support Vector Machine on combination 3 in percent