Content-based Music Recommendation Using Probabilistic Models

Vaughn Ho

University of the Witwatersrand

Supervisors: Mr. Ritesh Ajoodha, Dr. Benjamin Rosman

November 2017, Johannesburg

WITS
UNIVERSITY

Honours Research Report

School of Computer Science and Applied Mathematics

Faculty of Science

Declaration

I, Vaughn Ho, (Student number: 382342) am a student registered for COMS4044 in the year 2017.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that ALL the work submitted for assessment for the above course is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand may take disciplinary action against me if there is a belief that this in not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.

Signature: _____

Signed on the 13th day of November, 2017
Johannesburg, South Africa

Abstract

Basic music recommendation systems typically implement collaborative filtering techniques which use metadata such as a music item's title, genre and artist in conjunction with listeners' music usage history. This methodology in general provides inadequate music recommendations. This is because metadata does not represent the uniqueness of each music item well enough to differentiate between preferences; and music usage data is biased towards popular music as there is insufficient usage data for unpopular music. In this paper, music recommendation is performed by using content-based features extracted directly from music audio signals. Music is recommended by implementing an off-the-shelf expectation maximisation algorithm. This algorithm is a non-linear optimisation technique which optimises the log-likelihood function relative to the data. This machine learning clustering technique was applied to the *GTZAN* dataset which is a benchmark dataset.

Acknowledgements

I would like to express my greatest gratitude to my supervisor Mr Ritesh Ajoodha for his invaluable knowledge that he shared with me. His mentoring, advice, guidance, patience and time have been vitally important to the completion of my research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the internet being readily accessible, and progressively becoming more so to each person worldwide, online service delivery and online product consumption are becoming more prominent. The music industry has shifted to distribute music (not exclusively) via online music stores such as iTunes and Google Play and via music streaming services such as Spotify and Deezer [Van den Oord et al. 2013]. Music recommendation systems are relevant to these digital music distribution channels in order for them to target individual user preferences effectively. This allows music listeners to discover new music that they personally prefer [Van den Oord et al. 2013].

1.1 Motivation

Basic music recommendation systems naïvely search for music to recommend using music metadata and listeners' music usage history. Music metadata includes the music item's title, artist or band, album that it belongs to, genre, year of release and it's composer. A common technique implemented in these systems to recommend music is collaborative filtering. Collaborative filtering aims to determine music listeners' preferences from historical usage data [Dieleman 2014]. It uses a database of other user listening patterns which matches up with the target user's usage and subsequently recommends music that other users have liked to the target user [Li et al. 2005]. This presumes that two users have similar music preferences since they listen to the same music. Conversely, it presumes that if two pieces of music are listened to by the same group of people, the two music items have some commonality between them [Dieleman 2014].

However, collaborative filtering recommendation systems require a significant amount of data for it to generate good recommendations [Eck et al. 2008]. This dependency on available usage data leads to a significant issue called the "cold-start" problem. New music or music by unknown artists with few listeners are not readily recommended as there is insufficient or no historical usage data for collaborative filtering to use. Whereas popular music items are more easily recommended as there is more usage data for them. This also results in recommendations being repetitive and predictable. Additionally, metadata does not represent the uniqueness of each music item well enough to differentiate between listener preferences. Exploration into alternative techniques to avoid the "cold-start" problem has highlighted the potential of content-based music recommendations.

1.2 The Research Problem

Matching music items to a listener's preference is dependent on the uniqueness and context of the user as well as the music item as a whole [Wang and Wang 2014]. In order for a person to like or dislike a piece of music, he or she must explicitly listen to the music otherwise that person cannot make any judgements that accurately reflect his or her taste in music. During this process, the listener's preference is developed from the characteristics of the music's audio content - the music's vocal, melody, rhythm, timbre, genre, instrument or lyrics - rather than metadata descriptions [Wang and Wang 2014]. This provides motivation to use content-based features of music as a basis for a music recommendation system.

The hypothesis presented in this research is: a music recommendation system using content-based features extracted directly from music audio signals can be successfully used to recommend music.

1.3 Research Overview

The research presented performs music recommendation using content-based features extracted directly from music audio signal. This is an unsupervised learning and clustering problem. The expectation maximisation algorithm is used to cluster music appropriately from the audio content-based features. So given a music item, clustering translates to recommending music from the same cluster. The recommendation system is ultimately solved using probabilistic graphical models.

13 content-based features were selected based on the information gain ranking feature selection method. Feature extraction was performed using the *jAudio* application. The *WEKA* software provided an off-the-shelf implementation of the expectation maximisation clustering algorithm to cluster the *GTZAN* music dataset. From 1000 songs that were uniformly classified into 10 genres, clusters were obtained using the mean, the standard deviation and the Mel Frequency Cepstral Coefficients feature representations. Content-based features of music can be used to specify music items' uniqueness more accurately. Clusters generated show common feature attributes among the music audio signals. This consequently allows for better music recommendations that align with listener preferences.

Chapter 2

Background and Related Work

2.1 Music Feature Extraction

There are several feature extraction techniques that can be used for music information retrieval. Some of these methods are: reference feature extraction, content-based acoustic feature extraction, symbolic feature extraction and text-based feature extraction [Ajoodha 2014]. Reference feature extraction involves extracting the metadata of music. Metadata of music describes the music item's title, artist or band, album that it belongs to, genre, year of release and it's composer. Symbolic feature extraction is a technique where features are extracted from the music score. Text-based feature extraction extracts features from the lyrics of a piece of music. Content-based acoustic feature extraction is the method of interest for music information retrieval. Here, features are extracted directly from the music audio signal [Ajoodha 2014].

2.2 Music Recommendation

During the earlier time periods of investigating into music recommendation systems, Li et al. [2005] propose a collaborative music recommendation system based on an item-based probabilistic model where items are clustered into groups and predictions of recommendations are made for users by considering content information of audio signals and a Gaussian distribution of user ratings. The understanding is that since collaborative filtering involves clustering of users into similar usage patterns, probabilistic models can use this clustering to model the preferences of underlying users from which predictions are inferred. The clustering algorithm used in this music recommendation system design is the K-Medoids clustering algorithm.

User preference is a major factor in personalised music recommendation systems. And as such, Park et al. [2006] believe that user preferences of music changes according to the context. The authors believe that these changes in user preference according to context cannot be assumed to be fixed and that recommendation systems should take this dynamic into account. Context is defined to be "any information that can be used to characterise the situation of an entity such as a person, place, or object that is considered to be relevant to the interaction between a user and an application, including the user and applications themselves" [Park et al. 2006]. Consequently, Park et al. [2006] propose a context-aware music recommendation system using fuzzy Bayesian networks to infer the context and utility theory to consider the user preference by the context. Considering the context for music recommendation systems is a prominent trait of this research and it highlights the need to understand the variables of recommendation systems. Work by Park et al. [2006] provides an avenue

for the merging of context-aware recommendation systems with content based recommendation systems.

Current music usage is highly associated with user-generated keywords called social tags. This association of social tags and music is a source of information that could be exploited. Tags have been applied to music by music listeners to describe their personal sentiments towards a specific music item. Eck et al. [2008] have proposed a method to predict social tags using audio feature extraction and supervised learning. The idea is to generate the social tags automatically to provide additional information about music that is untagged or poorly tagged. This corresponds with new music or music by unknown artists which have low historical usage data. So, tags can be used to mitigate the "cold-start" problem and allow for unheard music to be recommended according to user preferences represented by tags. Tagging allows users to organise their music "in a personalised way" and consequently allows for music recommendation systems to recommend music in a similar "personalised" way if exploited. As Park et al. [2006] have highlighted the importance of user preference changing according to context, the tagging system can be used to complement the understanding of music, users and the contexts. Eck et al. [2008] use a meta-learning algorithm called AdaBoost to predict tags from acoustic features. The authors perform a one-vs-all binary classification for tagging.

Following from the automatically generated social tagging research by Eck et al. [2008], Ness et al. [2009] have extended on this area of interest for music recommendation systems. Ness et al. [2009] describe that the predictions from content-based music recommendation systems can be used to train a support vector machine and by stacked generalisation, train a second level support vector machine classifier to improve the performance of automatic tag generation by exploiting possible correlation between tags. Results obtained by the authors used two-fold cross validation and showed that each individual tag were equally important and that this method improves all per-tag evaluation metrics [Ness et al. 2009]. Stacked generalisation brings one possible alternative to improve and extend Eck et al. [2008].

Van den Oord et al. [2013] have contributed literature that directly focuses on content-based music recommendation. They propose a music recommendation system by training a deep convolutional neural network to predict latent factors from music audio. These latent factors determine user listening preferences. Van den Oord et al. [2013] used four models to predict latent factors: a linear regression model; a multi-layer perceptron; a convolutional neural network trained on log-scaled mel-spectorgrams to minimise the mean squared error of the predictions, and the same convolutional neural network but trained to minimise the weighted prediction error from a weighted matrix factorisation objective. Deep convolutional neural networks outperform the more traditional approaches of the regression model and multi-layer perception model. The authors also performed a qualitative evaluation on their latent factor predictions using a consine similarity.

The paper by Van den Oord et al. [2013] inspired research extension by Wang and Wang [2014]. The authors have identified that content-based music recommendation systems typically extract audio content features and then predict user preferences in two different stages. Wang and Wang [2014] propose combining these two separate steps into a single automated process to form a hybrid method. They believe that features learned automatically and directly from audio content to maximise music recommendation performance. The authors use a probabilistic graphical model and deep belief network to develop a content-based recommendation model to automatically learn the audio features. This paper presents very positive results showing that the proposed model outperforms existing deep learning models by taking into account music exposed to the "cold-start" problem and, on the other hand, music that have a "warm-start". The model developed by Van den Oord et al. [2013] was namely improved upon by Wang and Wang [2014].

Probabilistic graphical models allow for generative models which offer a more natural interpretation of a domain. They are able to deal with missing values and unlabelled data and perform well enough when the data is sparsely available [Koller and Friedman 2009]. The deep belief network method of probabilistic graphical models used by Wang and Wang [2014] has been used for multiple music information retrieval and other music related applications [Wang and Wang 2014]. These include music genre classification, music autotagging, rhythm style classification and understanding melody of music from music audio content [Wang and Wang 2014]. Probabilistic graphical modelling has been chosen as the core methodology of this research based on the broadness of its applications and successes supported by Wang and Wang [2014].

Chapter 3

Research Methodology

This research method is organised into two key parts: feature extraction and data clustering. The audio dataset used in this research is described in Section 3.1. The selection of features and the feature extraction procedures make up the feature extraction component of the methodology. These core aspects will be detailed in Section 3.2. The music data clustering component involves the expectation maximisation clustering algorithm used to recommend music. This clustering which translates to the action of recommending music is explained in Section 3.3.

3.1 The Dataset

Publicly available music datasets such as the Million Song Dataset, the Echo Nest Taste Profile Subset and the Last.fm dataset [Bertin-Mahieux 2011] were not used because these datasets contain metadata and pre-processed music audio features and do not provide actual audio files. Audio files are necessary in order to extract a specific feature or set of features which are not limited to those extracted by others. A music corpus of more current music was not constructed due to the limitations of licensing laws and copyright rules which must always be adhered to.

The *GTZAN* dataset is a popular music audio dataset used for various music genre recognition related work. This dataset contains 1000 music audio files in Waveform Audio File Format (.wav format). Each file is a 30 second music clip. This collection of music is categorised into 10 genres with each genre having 100 music clips assigned to it. The genres are: blues, classical, country, disco, hip hop, jazz, metal, pop, reggae and rock. Literature has shown that there exists duplicates, incorrect labelling and music distortion within the dataset [Sturm 2012]. Despite these discovered issues, the *GTZAN* dataset is worthwhile to use as a point of reference to compare work as the faults are universally the same and left uncorrected [Sturm 2012]. However, this research focuses on the content-based features of music and the degree to which these problems from this dataset affect content-based feature related work is out of the intended research scope. The *GTZAN* dataset is used here.

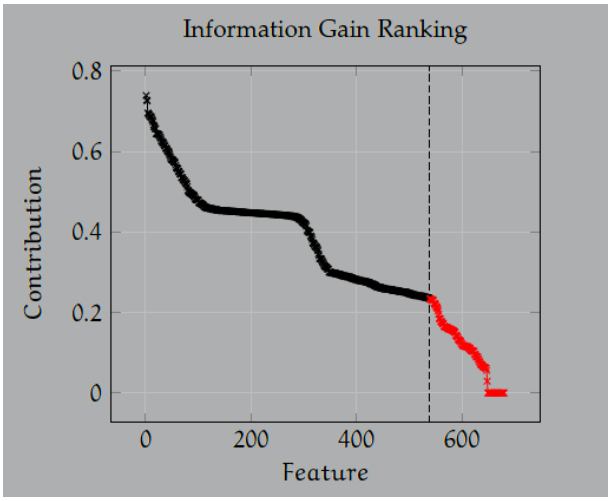

Figure 3.1: Contribution of features determined by information gain ranking [Ajoodha 2014].

3.2 Content-based Music Features

3.2.1 Feature Selection

Content-based acoustic features can be categorised mainly into timbre content features, rhythmic content features and pitch content features [Ajoodha 2014]. Timbre features are descriptors originally used for speech recognition but are extended to apply to music classification. Timbre features include Mel Frequency Cepstral Coefficients (MFCC), spectral centroid, spectral rolloff, spectral energy and spectral flux to name a few [Ajoodha 2014]. Rhythmic content features describe the regularity of the rhythm, beat and tempo of the music audio signal [Ajoodha 2014]. Pitch content features describe the melody and harmony aspects about musical signals. These features are extracted using pitch detection techniques [Li et al. 2005].

The *jAudio* application is an open source program for extracting audio features [McEnnis et al. 2006]. It is important to identify relevant features to extract that can effectively describe music items. The inclusion of irrelevant features may lead to over-fitting and thus lead to inaccuracies of results from the experiment. Feature selection methods such as principal component analysis [Van Der Maaten et al. 2009] are thus used for dimensionality reduction to help find an appropriate subset of features. The feature selection method used in this research is information gain ranking. This technique was not explicitly implemented but rather used with the backing up of existing literature; notably presented by Ajoodha [2014].

Information gain ranking is a feature selection filter method. This technique ranks features according to their contribution by means of a feature attribute evaluator [Ajoodha 2014]. The greater the contribution, the better the ranking. The features that were chosen and the features that were excluded were determined by a cut-off point. This is illustrated in Figure 3.1. Features that had a contribution lower than the cut-off point were eliminated from the set of candidate features to be selected. The way that Ajoodha [2014] decided on the cut-off point was to take different numbers of the highest contribution features and measure their genre classification accuracy. Figure 3.2 shows this process. The details can be studied further in Ajoodha [2014].

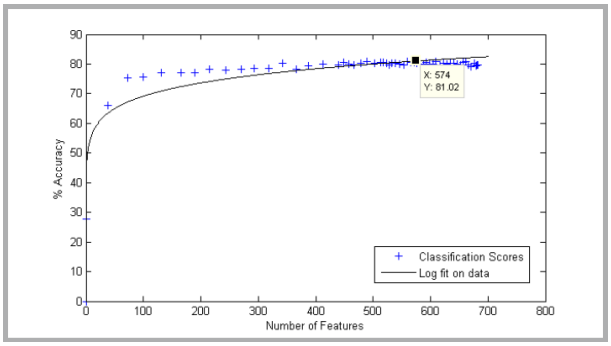

Figure 3.2: Trade-off between number of features and classification accuracy [Ajoodha 2014].

This research does not use all the features that satisfied the cut-off point criterion stated in Ajoodha [2014]. A subset of those selected features was used. Many feature extraction algorithms were implemented by the author and not available via *jAudio*. Hence, the following features were selected for extraction:

- Timbre content features:
  - Compactness: this is a measure of the noisiness of the audio signal.
  - MFCC: Mel-frequency cepstrum coefficients are the coefficients that collectively make up a Mel-frequency cepstrum. This is a widely used feature in music information retrieval.
  - Peak smoothness: this describes the smoothness of the peak of the audio signal.
  - Spectral centroid: this describes the centre of mass of the power spectrum which is used to calculate the brightness of the audio signal.
  - Spectral flux: this measures the rate of change of the magnitude spectrum.
  - Spectral rolloff point: this is the frequency at which 85% of the audio signal energy is contained below this frequency.
  - Spectral variability: this is the standard deviation of the magnitude spectrum of the audio signal which describes the spread of the signal.
- Rhythmic content features:
  - Beat histogram: this shows the rhythmic intervals in the audio signals from the signal strength.
  - Fraction of low energy: this describes the proportion of silence that the audio signal contains.
  - Root mean square: this measures the magnitude of the audio signal over a window.
- Pitch content features:
  - LPC: linear predictor coefficients are defined by auto-correlation tools and are used as features.
  - Strength of strongest beat: this describes how strong the strongest beat in the beat histogram is compared to other beats.
  - ZCR: zero crossing rate is the number of times that the audio signal changes sign within a window.

3.2.2 Feature Extraction

Using *jAudio*, aggregators are applied to the output of every feature. These aggregators are functions that approximate a high dimensional input feature vector by reducing it to a lower dimensional vector [Ajoodha 2014]. In general, each dimension of the output vector is the result of an aggregator being applied to all the values for that dimension from the input feature vector [McEnnis et al. 2006]. Three aggregate feature representations are used: the mean, the standard deviation and the MFCC representations. These were applied to the selected features stated in Section 3.2.1. The multiple feature histogram aggregate representation in *jAudio* was found to be faulty and hence not used. It should be noted that the optimal representation for each feature can be found from [Ajoodha 2014]. This distinction was not made in the scope of this research.

3.3 Expectation Maximisation Algorithm and Clustering

Probabilistic graphical models (PGM) are used to build the music recommendation system. PGMs are used because they allow us to develop a declarative representation of the music recommendation system. This declarative representation separates the model from the algorithm which allows the application of multiple algorithms to the model. Alternatively, it enables the improvement of the model for an algorithm [Koller and Friedman 2009]. The music recommendation system also contains a significant amount of uncertainty, namely the music to be recommended to a music listener which is unobservable.

The Naïve Bayesian Network model is used as a representation of the problem to be solved. This model is applicable to the clustering of music into appropriate recommendations problem. A Bayesian network is a directed acyclic graph whose nodes represent the random variables and edges that correspond to the direct influence of one node on another [Koller and Friedman 2009]. These random variables represent the content-based features extracted from the music audio signals and are observed. The goal is to infer the latent variable or, in this context, the class label that a music item belongs to, which is the recommendation. The Bayesian network is represented by Figure 3.3 where X_1, ..., X_n are our observed features. The Naïve Bayes model assumes that instances (music items) fall into one of a number of mutually exclusive and exhaustive classes [Koller and Friedman 2009]. There is also the assumption that the features are conditionally independent given the class. Unfortunately, a weakness to the strong assumptions underlying the Naïve Bayes model is the decrease in accuracy as it tends to overestimate the impact of certain features. Also, as the number of features increase, the performance of the model decreases [Koller and Friedman 2009].

Since the class label or recommendation is unobserved, the expectation maximisation (EM) algorithm [Koller and Friedman 2009] is used to assign parameters to latent variables for our Bayesian clustering task. To estimate the parameters for our Bayesian network, maximum likelihood estimation (MLE) will be used. The likelihood function in MLE for a given choice of parameters is the probability (or density) the model assigns the data [Koller and Friedman 2009]. The EM algorithm is specifically aimed to optimise likelihood functions. The parameters and the values of the missing variable both need to be estimated. Again, the missing variable here is the recommendation for a music item. The parameters can be estimated given the values of the class labels and the class labels can be estimated given the parameters. Since both of these are unknown, the EM algorithm initialises the set of parameters arbitrarily and uses these values to estimate and complete the missing values. This is known as the ”E-step” (expectation). The newly estimated missing values are then used to re-estimate

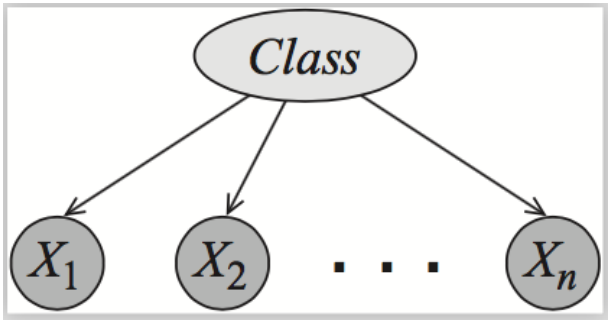

Figure 3.3: Bayesian network graph for a Naïve Bayes model [Koller and Friedman 2009].

the parameters. This is known as the ”M-step” (maximisation). MLE is involved in the ”M-step”. This process is then iterated until convergence. The EM algorithm is guaranteed to improve the likelihood function after each iteration [Kolller 2017].

The Bayesian clustering paradigm views this clustering task as a learning problem with a single hidden variable which represents the class from which an instance comes. Each class is associated with a probability distribution over the features of the instances in the class [Koller and Friedman 2009]. EM performs a soft cluster assignment of the data points. This means that EM assigns a probability distribution to each data instance; and each instance contributes a part of its weight to more than one cluster, in proportion to its probability that it belongs to each of the clusters. So, each music item belongs, with some probability, to multiple recommendation classes [Koller and Friedman 2009].

The *WEKA* explorer application provides an implementation of the expectation maximisation algorithm. *WEKA*'s implementation can automatically determine the number of clusters to be created from the given dataset by cross-validation or the maximum number of clusters can be manually set. In this case, the number of clusters was left to be automatically determined by *WEKA*. The cross-validation performed to determine the number of clusters is done in the following steps:

1. The number of clusters is set to 1.
2. The training set is split randomly into 10 folds.
3. EM is performed 10 times using the 10 folds by cross-validation.
4. The log likelihood is averaged over all 10 results.
5. If the log likelihood has increased, the number of clusters is increased by 1 and the process continues from step 2.

The number of folds is fixed to 10 while the number of instances in the training set is not less than 10. If the number of instances in the training set is less than 10, then the number of folds is set equal to the number of instances.

Chapter 4

Results and Discussion

4.1 Music Clustering Results

Some numeric results can be seen in Table 4.1. The greatest number of clusters (30 clusters) is obtained using only the standard deviation as the feature representation for the features extracted. This is followed by using only the mean which yields the the next greatest number of clusters (26 clusters), then by using the mean, standard deviation and MFCC together (20 clusters) and finally the MFCC aggregate on its own (16 clusters). Each feature representation's corresponding final log likelihood value is included in Table 4.1. The log likelihood values are presented as a point of reference as this value should increase after each iteration of the EM algorithm. The number of clusters obtained using certain feature representations or combinations of feature representations is not an indication of clustering performance. *WEKA* does measure the goodness of fit of the clustering by the log likelihood. In general, the larger the log likelihood, the better the model fits the data. However, increasing the number of clusters will increase the log likelihood but this may lead to overfitting so no solid interpretation from this can be made.

Figure 4.1 and Figure 4.2 are two example clustering visualisations obtained from our experiment using all three feature representations together. Figure 4.1 shows clouds of clusters based on each audio signal data point's average of it's root mean square and average of it's zero crossing rate. So given, the audio signals' root mean square average and zero crossing rate average, it is possible to assign that audio signal to one of the 20 clusters. These two features distinguish the clusters to a certain degree, however, these clouds of data clusters are biased to the person interpreting the visual. Thus two content-based features are not sufficient and accurate enough to make clear statements on how well they cluster the music audio dataset. Figure 4.2 shows how the standard deviation of compactness and the MFCC aggregate of fraction of low energy windows cannot be used to distinguish between clusters at all. A generalised group of features to cluster a music item was unable to be obtained. Recall that there was a limitation from *jAudio* where the multiple feature histogram aggregate feature representation was not functional. This hindered the experiment from obtaining more results.

Feature Representation	**No. of clusters**	**Log likelihood**
Mean	26	105.34769
Standard deviation	30	348.40583
MFCC	16	-459.81196
All 3 representations	20	-370.50731

Table 4.1: EM algorithm clustering results using different aggregate features.

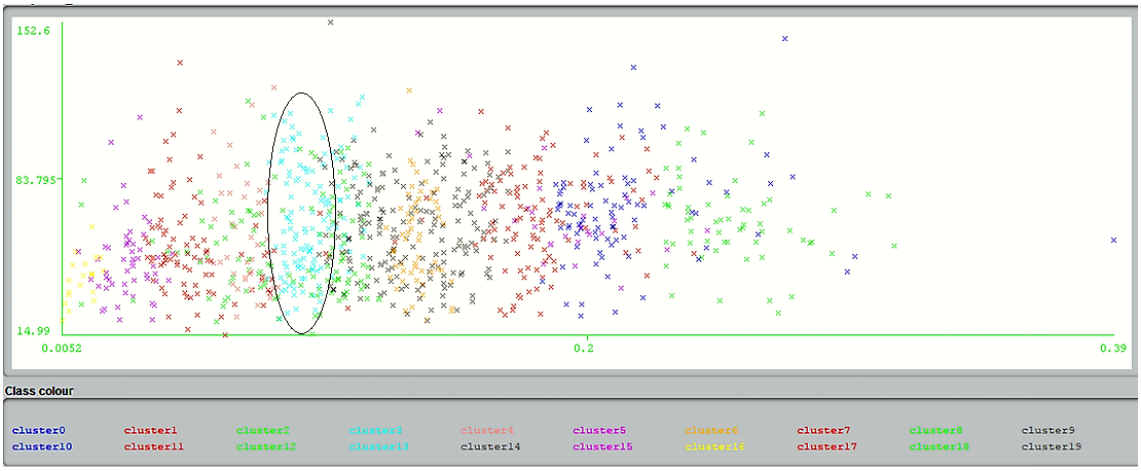

Figure 4.1: X-axis: Root mean square average. Y-axis: Zero crossing rate average. Cluster clouds obtained using the mean, standard deviation and MFCC feature representations together.

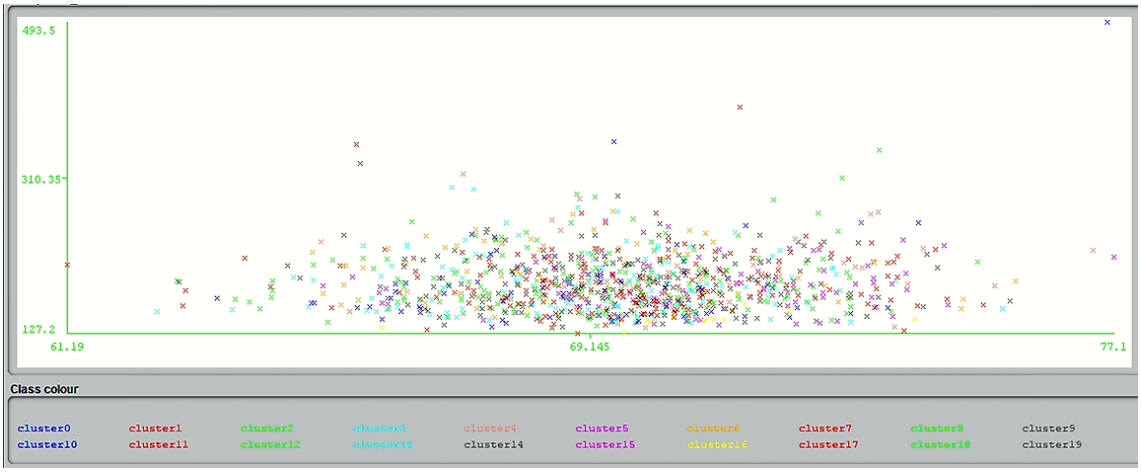

Figure 4.2: X-axis: Fraction of low energy MFCC. Y-axis: Compactness standard deviation. Obtained using the mean, standard deviation and MFCC feature representations together.

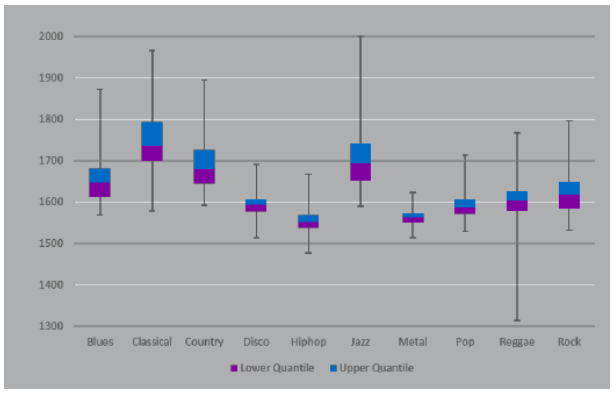

Figure 4.3: Compactness differentiates blues, classical and jazz from other genres [Ajoodha et al. 2015].

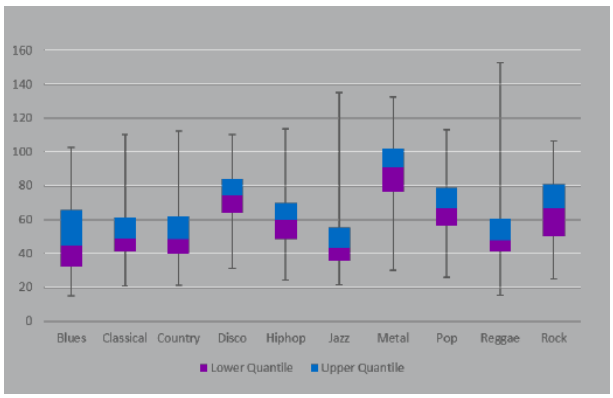

Figure 4.4: ZCR differentiates metal, disco and hip hop from other genres [Ajoodha et al. 2015].

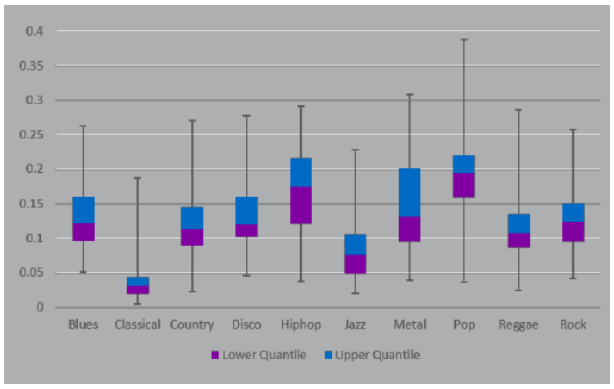

Figure 4.5: Energy differentiates classical, pop, hip hop, metal and jazz from other genres [Ajoodha et al. 2015].

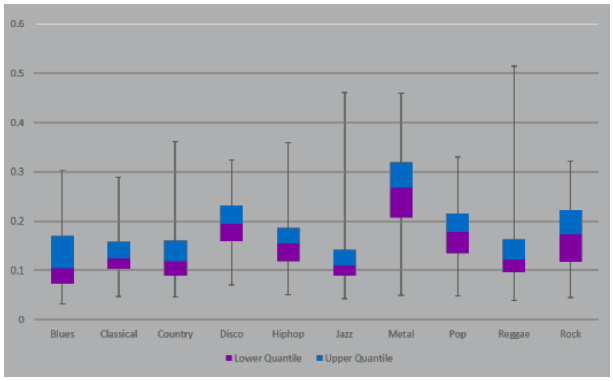

Figure 4.6: Spectral rolloff differentiates metal from other genres [Ajoodha et al. 2015].

4.2 Evaluation

An explicit measurement of performance of recommending music using content-based features by the expectation maximisation clustering algorithm was not obtained. This is a result of not having the ground truth generative distribution of the *GTZAN* dataset to conduct an evaluation. Thus, it was not possible to determine the accuracy of the clusters achieved without the underlying true generative distribution. More heuristic evaluation methods are required.

An alternate method of evaluation would be to use online music services such as Spotify and Deezer. The procedure would be to observe the music items that those systems recommend, then compare them to the music items that this modelled music recommendation system would suggest; relative to the same music taste objective. Another method of performance measurement would be conducting a survey. The survey would involve giving volunteers a selection of music to listen to and based on their choice, recommend the appropriate music closest to their preferred selection. The evaluation will require sufficient participants to give a reasonable indication of the performance of the recommendation system. However, this method could be subject to each participant's biased judgement of rating the appropriateness of the recommendation. Alternatively, a mainstream dataset of music with well-defined clusters could be used to show how well the statistical recommendation model can predict recommendations. Similarly to a well-defined dataset, synthetic data with specific characteristics could be generated which the music recommendation model would analyse. The recommendation clusters would be evaluated based on the specifically designed data. Finally, an evaluation conducted by existing literature is another approach. This is done here.

Recall that the *GTZAN* dataset was categorised into 10 genres. Using any combination of the feature representations for clustering always yielded more than 10 clusters in this experiment. See Table 4.1. This indicates that there are music instances from the *GTZAN* dataset that have common feature attributes but have different, overlapping genre metadata classification labels. Audio signal data points belonging to the same cluster means that they have similar content-based feature composition. So given a particular music item, other music items that are part of the same cluster will be recommended to the listener.

Despite conflicting genre labels, content-based feature clustering is able to describe music items more specif-

ically. Content-based feature genre classification literature from Ajoodha et al. [2015] shows that certain content-based features are able to differentiate some genres from other genre types. Figure 4.3 shows the compactness feature values distributed over 10 *GTZAN* genres. Here, the compactness content-based feature distinguishes the blues, classical, jazz and country genres from other genres. The boxes of box-and-whisker plots for blues, classical, jazz and country do not intersect with the boxes of the other genres.

This puts forward the argument that music listeners who prefer to listen to classical music, may also prefer blues, jazz or country music based on the music item's compactness content-based feature. So, given the classical music item the user prefers, a blues, jazz or country music item with a similar compactness feature would be a suitable music item to recommend to that listener. Figure 4.4, Figure 4.5 and Figure 4.6 show the zero crossing rate, energy and spectral rolloff feature values distributed over 10 *GTZAN* genres respectively. The zero crossing rate feature distinguishes metal, disco and hip hop from other genres. The energy feature distinguishes classical, pop, hip hop, metal and jazz from other genres. Finally, spectral rolloff tells metal apart from other genres. This gives positive analysis on the use of content-based features extracted directly from music audio signals to recommend music.

Chapter 5

Conclusion

Automatic music recommendation systems are important tools for e-commerce companies to promote their databases of music to the right music listeners and for music listeners to find new music that they enjoy. Content-based features extracted directly from music audio signals have been analysed for music recommendation here. The consensus is that naïve metadata recommendation techniques and collaborative filtering techniques have shown significant weaknesses that content-based recommendation can overcome. Recommending music using probabilistic graphical models via the expectation maximisation algorithm to cluster music has shown favourable merits as a methodology for a recommendation system.

Music items from the same clusters share similar physical audio feature composition. Content-based features are able to describe music items more uniquely which allow for music to match listener preferences more closely based on similar feature composition. Content-based music genre classification [Ajoodha et al. 2015] has helped to verify the usefulness of such features for music applications. The hypothesis (Section 1.2) presented above can be accepted.

Some limitations to note from the work done here: not all the possible features were able to be extracted as *jAudio* did not have the implementations of the feature extraction algorithms to do so; the multiple feature histogram aggregator was found to be unusable for the feature extraction process; additionally, the optimal feature representation for each content-based feature was not used in the EM clustering algorithm. A combination of the optimal feature representations would potentially present more positive and different results. Improvements to these issues would be beneficial to this area of research.

Possible extensions of this work would be to incorporate the context of user preferences for music into a content-based recommendation system. To create a more robust system, including a context-aware aspect to the statistical recommendation model is suggested for further investigation [Park et al. 2006]. There is room for adaptations and optimising the recommendation technique presented in this work such as new feature extraction implementations as well as a new implementation of the EM algorithm. Other clustering algorithms could be applied and compared to the methodology shown here. Machine learning techniques such as support vector machines [Ness et al. 2009] and deep convolutional neural networks [Van den Oord et al. 2013] can be used and evaluated in this area of interest. The use of a different and verified dataset with a ground truth generative distribution may be useful for future work. This dataset could include a wider variety of classical, contemporary, commercial and ”underground” music that better reflects popular, less popular and unpopular music.

Bibliography

Ritesh Ajoodha. *Automatic music genre classification*. PhD thesis, University of the Witswatersrand, 2014.

Ritesh Ajoodha, Richard Klein, and Benjamin Rosman. Single-labelled music genre classification using content-based features. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), 2015*, pages 66–71. IEEE, 2015.

Thierry Bertin-Mahieux. Million song dataset. http://labrosa.ee.columbia.edu/millionsong/, 2011.

Sander Dieleman. Recommending music on spotify with deep learning. http://benanne.github.io/2014/08/05/spotify-cnns.html, August 2014.

Douglas Eck, Paul Lamere, Thierry Bertin-mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 385–392. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3370-automatic-generation-of-social-tags-for-music-recommendation.pdf.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, Cambridge, Massachusetts, 2009.

Daphne Kolller. Probabilistic graphical models. https://www.coursera.org/learn/probabilistic-graphical-models, 2017.

Qing Li, Sung Hyon Myaeng, Dong Hai Guan, and Byeong Man Kim. A probabilistic model for music recommendation considering audio features. In *Asia Information Retrieval Symposium*, pages 72–83. Springer, 2005.

Daniel McEnnis, Cory McKay, Ichiro Fujinaga, and P Depalle. jaudio: Additions and improvements. In *ISMIR*, pages 385–386, 2006.

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.

Han-Saem Park, Ji-Oh Yoo, and Sung-Bae Cho. *A Context-Aware Music Recommendation System Using Fuzzy Bayesian Networks with Utility Theory*, pages 970–979. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-45917-0. doi: 10.1007/11881599_121. URL http://dx.doi.org/10.1007/11881599_121.

Bob L Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012.

Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in neural information processing systems*, pages 2643–2651, 2013.

Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 627–636. ACM, 2014.