Music Recommender and Genre classification system

Sunil BN[*]
University of Colorado Boulder
Boulder, Colorado
United states
suba5417@colorado.edu

Praveen Kumar Devaraj[†]
University of Colorado Boulder
Boulder, Colorado
United states
prde1873@colorado.edu

Suresh Kumar[‡]
University of Colorado Boulder
Boulder, Colorado
United states
suku3476@colorado.edu

ABSTRACT

In the recent years, music recommendation systems are gaining popularity because a lot of music is now sold digitally. Most of the music recommender system are based on collaborative filtering or content based filtering. This project presents a hybrid music recommendation method that solves two of inherent problems in conventional methods. Collaborative approach cannot recommend music that has no ratings and suffers from cold start. The aim is to recommend a list of songs to a user given a little about his/her listening history and complete listening history of other 1 million users. Where as content based filtering recommends music based on the similarity of contents and doesn't truly reveal the user preferences.

Our approach combines both methods i.e based on user rating and content using discriminative model. We propose to train the collaborative filtering model such as Mlib (Machine learning library) and predict songs user might be interested in and over which we run the content based filtering in order to make the recommendations more accurate and closer to the user preferences.

We propose to try different classification techniques in content based filtering to generate a comparison of the efficacy of each of the techniques in the prediction a list that is closest to the user preferences. We compare a few conventional methods of collaborative and content based filtering with our hybrid model, and evaluate the predictions quantitatively and qualitatively on the Million Song Dataset(MSD).

Keywords

Hybrid model; collaborative filtering; content based filtering; Machine learning library(Mlib); Million song dataset, MSD

[*]Graduate Student
[†]Graduate Student
[‡]Graduate Student

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2016 ACM. ISBN .
DOI:

1. MOTIVATION

Given a huge collection of digital media like audio, video etc a recommender system help users find and select items of their tastes. The Music recommender system will present to the listeners a subset of the songs that are well suited to their interest based on mood/genre/time of day etc. Thus building a robust music recommendation and Genre classification system is complicated by the variety of songs, lyrics, content of the songs and size of dataset etc. There is no single solution that can be classified as the best solution for the recommendation as a lot of factors determine the liking of a song by the user.

Further more most of the recommendation system work on either collaborative or content based filtering, both have their pros and cons.

Collaborative Filtering recommend songs to the user by considering how other users have rated them. In this case those songs which have not been rated are not considered which reduces the training data to a large extent. Also this depends on the user preferences, the artists of the recommended pieces could be the same and most users tend to rate their songs highly. So the recommendations made are unrelated to the user preferences.

Content-based methods recommend songs similar to the users' listening history in terms of musical properties. This results in a large variety of artists i.e. various songs are recommended even when they have not been rated. However, these methods have low accuracy of recommendations because similarity in content is only one of many factors characterizing user preferences.

Thus, in this project we will consider best features of both the conventional methods to build a better and more accurate music recommender system and also compare this with the existing recommender systems.

2. LITERATURE SURVEY

With the explosion of Internet in recent decades, Internet has become a major source for information such as audio, video, books, geography etc. The music also has drifted more and more towards the digital distribution of musical pieces through on-line stores like iTunes, Spotify, SoundCloud, Shazam and Google play music. As a result music categorization and recommendation has become a persistent problem. Music categorization allow on-line music to be classified based on the genre, lyrics etc. so that the users' can browse through the similar contents. And recommender system enables users' to discover new music that matches their tastes based on previously listened music.

Though much of the studies and research has been done in recommender system, still the problem here is complicated by the variety of tons of songs, different genres, social and geographical factors that influence the users. The number of music that can be recommended to the user is very large.

The problem is to organize the humongous amount of music available on-line into categories like based on genre, artist, most popular etc.MIR techniques have been developed to solve problems such as genre classification, artist identification, and instrument recognition. Since 2005, an annual evaluation event called Music Information Retrieval Evaluation eXchange (MIREX1) is held to facilitate the development of MIR algorithms[3].

Additionally, music recommender system is to help the users find and discover the music according to their tastes. A good recommender system should be able to learn the listening styles/preferences of users and generate play-list accordingly. Meanwhile, the development of recommender systems provides a great opportunity for industry to aggregate the users who are interested in music. More importantly, it raises challenges for us to better understand and model users preferences in music.

Some music discovery websites such as Last.fm, Allmusic, Pandora and Shazam have successfully used these two approaches into reality. In the meantime, these websites provide an unique platform to retrieve rich and useful information for user studies.

3. PROPOSED WORK

The two main objective of the project is to recommend songs to users and genre classification of songs.

3.1 Building a Music Recommender System

There are three approaches to building a recommender system - Collaborative filtering, Content-based filtering and a hybrid approach. Each of these approach makes use of users and items data. In this paper, we use the term items and songs interchangeably. The following section describes each approach. For the project, we will use a hybrid approach combining collaborative filtering and content based filtering techniques.

3.1.1 *Collaborative Filtering Approach*

The first approach is to recommend songs using collaborative filtering (CF) technique. Collaborative filtering is based on the idea of determining user's preference using the listeners' historical data. Collaborative filtering technique recommends a song to the test user based on how other users rated the recommended song. For example, if a user listened to song A, B, C and another user listened to song B, C, D then song D would be recommended to 1st user and song A would be recommended to the second user.

There are two types of collaborative filtering approaches: Memory based and Model-Based. Memory based approach uses user rating data to compute the similarity between the users or the songs. Memory based approach is easy to implement but its performance decreases as the data gets sparse which is quite the case with our songs database.
A popular way to compute the similarity between two users x and y is to use the cosine similarity where I_{xy} is the set of items rated by both users x and y:

$$sim(x,y) = cos(x,y) = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in I_{xy}} r_{x,i}} \sqrt{\sum_{i \in I_{xy}} r_{y,i}}}$$

Model bases approach uses data mining techniques and machine learning algorithms to find patterns based on training data. The users and items are described by a small set of latent factors that can be used to predict missing entries. This technique handles sparsity better, is scalable with large dataset and improves prediction performance. We will consider model-based approach to CF and make use of the Alternating Least square algorithm to learn the latent factors. The play count of a song will be used as the rating value.

The low-rank matrix factorization is a minimization problem, in which the cost function measures the fit between a given matrix (user preference matrix) and an approximating matrix, subject to a constraint that the approximating matrix has reduced rank.

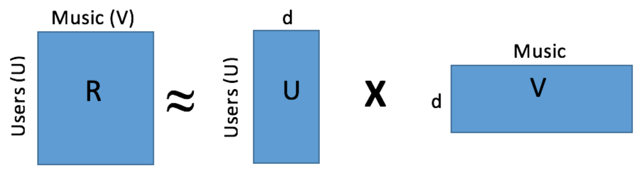

Figure 1: Low Rank Matrix Factorization

The low rank approximation problem is thus formulated as follows to learn the factor vectors (u_i, v_j),

$$(u_i, v_j) = \min_{x,y} \sum_{(u,i) \in K} (p_{u,v} - r_{u,v})^2$$

where, $p_{u,v}$ and $r_{u,v}$ are predicted and observed ratings for user u and item v respectively. K is a sparse matrix of known ratings.

Solving the low rank approximation problem as formulated above usually overfits the data. In order to avoid overfitting a common technique is to use Tikhonov regularization to transform the low rank approximation problem into the following:

$$(u_i, v_j) = \min_{x,y} \sum_{(u,i) \in K} (p_{u,v} - r_{u,v})^2 + \lambda(||u_i||^2 + ||v_j||^2)$$

where, λ is regularization parameter, which penalizes parameters with large magnitude.

The advantage of Collaborative filtering approach is that it does not need to have any information about the songs. The key issue with this approach is that is needs to have a

Algorithm 1 ALS Algorithm

```
1: Set up vectors yUsers, yItems, xq, and xp
2: Initialize $\mu$, $a_i$, $b_u$, $y_{ui}$
3: while MSE not converged and iteration $\neq$ max limit do
4:     Create $q_i$ vectors
5:     Solve for $p_u$ vectors
6:     Create $p_u$ vectors
7:     Solve for $q_i$ vectors
8:     Rescale each $p_u$ and $q_i$ vector
9:     Update $\mu$, $a_i$, $b_u$, $y_{ui}$
10:    Find MSE
11: end while
```

large user data and thus suffer from cold start state for new users or new items. In order to make appropriate recommendations for a new user, the system must first learn the user preference by analyzing rating activities. This method cannot recommend new song releases, i.e., the songs which has never been listened by any user in the database.

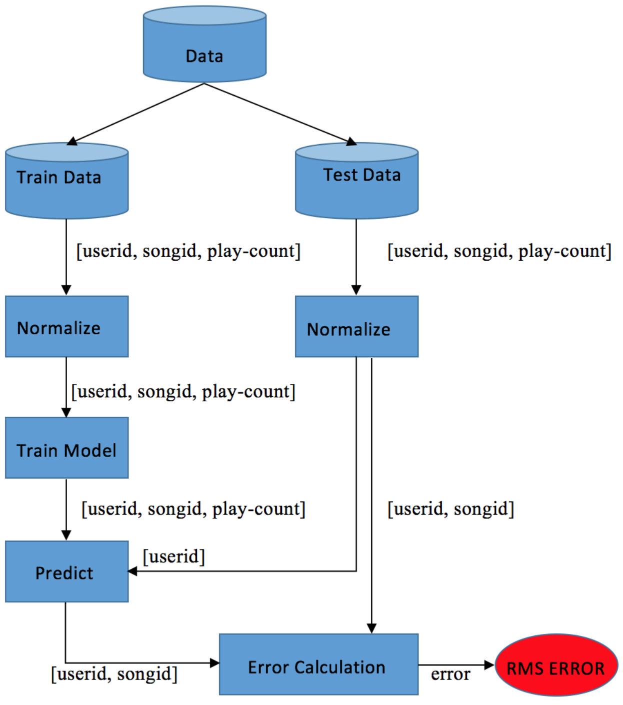

Figure 2: Collaborative Filtering model

3.1.2 *Content Based Filtering approach*

The second approach is to recommend songs using content-based filtering technique. This is based on the attributes and features of the songs and profile of user preferences. Using the various song attributes like danceability, tempo, energy, timbre, key, loundness, duration, etc. a user profile is built to indicate the type of songs the user likes. This approach will recommend songs to the user which are similar to the songs listened by user in the past. The features of the item would be represented using item presentation algorithms like vector space representation. The profile of the users is based on a weighted vector of item features. The weights denote importance of each feature to the user. The different machine learning algorithms like Bayesian Classifiers, cluster analysis, decision trees, artificial neural networks, etc., can be used to estimate the probability that a user is going like the recommended song. We would be experimenting with many of these machine learning algorithms and select a best algorithm based on evaluating metrics.

The advantage of content-based filtering approach is that it is user independent, i.e., it does not need to find similarities between different users. It only have to analyze items and profile of the given user. It does not suffer from cold start, i.e., new songs can be recommended before being rated by any users. The disadvantage is that if the songs does not enough attributes and information to discriminate it with other songs, the recommendation will not be precise. If there is not enough information to build a profile for the user, the recommendation could not be done correctly.

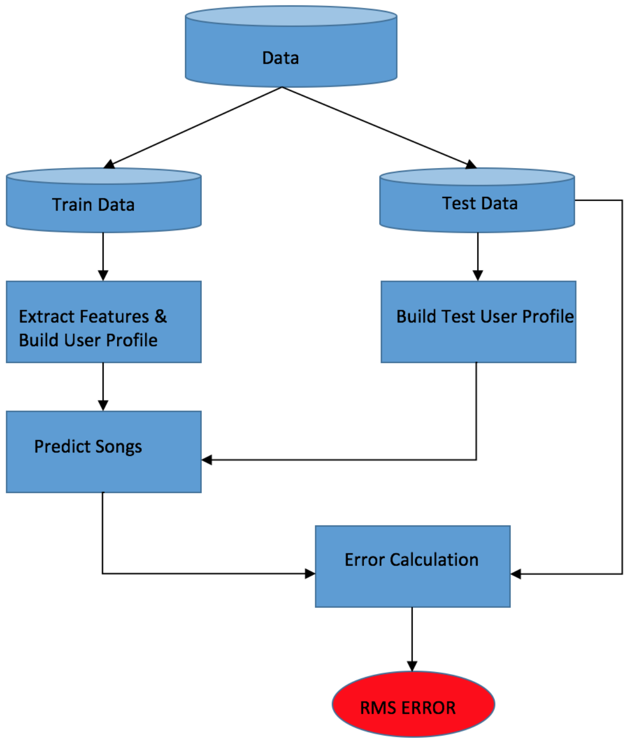

Figure 3: Content Based Filtering model

3.1.3 *Hybrid approach*

A more effective recommendation can be done by combining collaborative filtering and content-based filtering. Both of these techniques have some shortcomings as described above, but can be overcome using a hybrid approach. Hybrid approach can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa) or by unifying the approaches into one model.

This project would make recommendations based on collaborative filter and content-based filter separately and then combine them to give the final recommendations. The collaborative filtering would recommended the songs which have been listened by group of users, but it will not recommend new songs. The content-based filtering approach would be applied to newly released songs which has not yet been listened by a group of users. We will assign a score to the songs recommended by both of these techniques, then sort the songs in decreasing order of their scores and then give the final recommendation. This approach would make sure that the recommendation is for both newly released songs as well as popular listened songs.

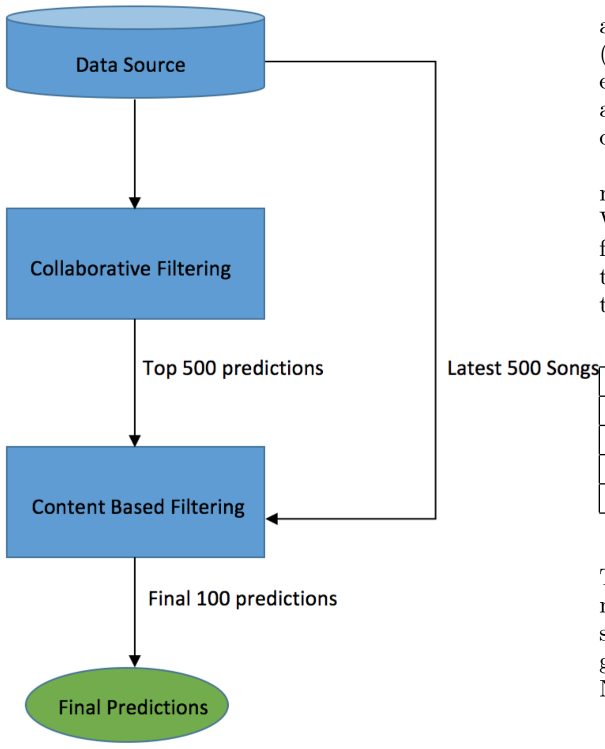

Figure 4: Hybrid Music Recommender System

3.2 Building an Automatic Genre Classification System

A music genre is conventional category that identifies some piece of music as belonging to some stylistic criteria. Automatic Genre Classification is the classification of a given piece of music into its genre using data mining techniques and machine learning algorithms. The music genre classification would be done by building a feature representation of the song followed by a classification system which would predict the genre.

The various feature classes to represent acoustic and musical aspect of the data are timbre (sound texture), loudness, tempo, danceability, key, energy, duration, artist name, etc. Commonly used classifiers are Support Vector Machines (SVM), k-Nearest-Neighbor (kNN) classifiers, Gaussian Mixture Models, Linear Discriminant Analysis (LDA), etc.

3.3 Dataset Description

The Million Song Dataset (MSD), is used as the primary dataset for our project. The raw data consisted of listening history of a million users in HD5 format. The dataset consists of the listening history of million users. but we used only a part of the data in our project. The whole dataset is about 280 Gb which consists of 48 million entries in it. We have considered a subset of 1 million entries and we have divided the same into Train and Test data in the ratio of 4:1 for testing and evaluating the accuracy of the system.

The super set consists of 90 attributes, 12 timbre average and 78 timbre covariance. The first value is the year (target), ranging from 1922 to 2011.The songs features are extracted from the The Echo Nest API. We take the average and covariance over all segments, each segment being described by a 12-dimensional timbre vector.

It is a freely-available collection of audio features and metadata for a million contemporary popular music tracks. We would first load the user data and run a collaborative filtering methods on it such as Probabilistic Latent Semantic Analysis. For Collaborative filtering we considered attributes like SongId, UserId and the Play count of the songs.

Type of Data	**Number of Records**
Number of Songs	10000
Number of Records	1.45 million
Train Data	1.16 million
Test Data	0.29 million

This method would predict the songs the user might like. The next step we consider attributes like Dancebility, Loudness, Timbre, SongHotness, ModeofConfidence etc of the songs predicted and train a model using machine learning algorithm such as logistic regression, Support vector machine, Naive Bayes etc and then predict the final recommendations.

3.3.1 Data Collection and Preprocessing

Dataset named Million Songs is available here. The whole dataset is about 280GB of which we have considered a subset of 3GB for evaluating the system. We have reduced the dataset by considering less number of attributes from the dataset. Of the total 90 attributes we have considered only 15 attributes. We considered the following attributes for the reduced dataset

- Danceability: A number that ranges from 0 to 1, representing how danceable The Echo Nest thinks this song is.
- Duration: Length of the song, in seconds.
- Energy: A number that ranges from 0 to 1, representing how energetic The Echo Nest thinks this song is.
- Key: The key that The Echo Nest believes the song is in. Key signatures start at 0 (C) and ascend the

chromatic scale. In this case, a key of 1 represents a song in D-flat.

- Loudness: The overall loudness of a track in decibels (dB).
- Mode: Number representing whether the song is in a minor (0) or major (1) key. Use this in conjunction with 'key'.
- TimeSignature: Time signature of the key.
- SongId: Unique Identifier of the song.
- UserId: Unique Identifier of the user.
- PlayCount: Number of times the song has been played by a particular user.
- TrackId: Unique Identifier of the track.
- TimbreAverage: Superset has a total of 12 attributes for AverageTimbre. So we have reduced it to a single attribute which is a mean of all the 12 timbre values in the dataset.
- ArtistName: Name of the artist of the album.
- Title: Title of the song.
- Tempo: It gives the number of beats per second of the song.

3.3.2 Data Cleaning

The total size of dataset is around 280GB and it is possible to build a model using the whole dataset, but building this model takes approximately close to 16 hours of runtime. So in order to reduce the runtime we considered a subset.

Since we have considered a subset of data there could be instances where the features of the songs are not available. For these instances we are ignoring the whole song from being considered for the recommendation of the songs.

The other option we have included is that if a numeric attribute is missing we have considered a mean of the range of values for that attribute. For example, we consider "Danceability" attribute whose value ranges between 0 and 1, if its missing we consider 0.5 as the default value of the attribute.

3.3.3 Data Warehouse

We are considering a dataset for 10000 songs with around 1.45 million entries in the dataset. We have used a CSV (Comma Separated Value) file system to store this data.

We have considered two files, one for Collaborative Filtering where we consider only three attributes namely UserId, SongId and the PlayCount (TrainTriplets). For Content based filtering we have to consider various features of the song like tempo, timbre etc for which we have used another CSV file. The entries in these files are mapped using SongId.

3.4 Sub-Task

3.4.1 Completed Tasks

- **Recommending popular songs to the user using Collaborative Filtering**
  For this task we used Apache Spark model-based collaborative filter which uses Alternating Least Squares (ALS) algorithm to learn the latent factors. We normalized the play-count attribute from the traintriplets, in order to reduce the values to a same scale, we have used min-max normalization which is given by,

  $$X = \frac{x_i - min}{max - min}$$

  We ran the ALS algorithm on the traintriplets with normalized play-count to build a model, which was used to predict the songs for the test data.

- **Implemented a Genre classifier**
  For Genre Classification we extracted various features of songs from MSD. This feature set includes "Loudness", "Tempo", "TimeSignature","Mode", "Key", "Average Timbre", "Duration" and "Variable Timbre", for classifying the songs.

  We split the data in the ratio of 4:1 for train and test data respectively and run random forest classifier on the train data to build the model. We predict the genre labels for the test-data using this model.

- **Evaluation of Model-Based Collaborative Filter**

  We used the model to predict the songs for a particular user for whom we had the listening history. Then correctness of the model was computed using Root Mean Squared Error.

  The MSE of the model with different values of regularization parameter, rank are computed and tabulated below.

Lambda	**Rank**	**MSE**
1.0	8	15.23
10.0	8	16.41
5.0	8	12.17

  The Mean squared error of the collaborative filter model with testing data is given below in the table.

Mean Squared Error	12.31
Mean Absolute Error	2.41
Root Mean Square Error	3.51

- **Evaluation of Genre Classifier**

  We used the model to predict the genre of the songs for which we had the genre labels. The model predicted labels were compared with the actual labels to compute the accuracy of the classifier.

  The accuracy of the genre classifier model using Random forest is show below,

Feature-Id	Feature vector	Accuracy
F_0	genre, track id, tempo, time signature, duration, loudness, key, mode, danceability	50%
F_1	F_0, average timbre, variable timbre	56%

3.4.2 To-Do Tasks

- Recommending songs to the user using Content based Filtering.
- Integrating Content based filtering with Collaborative Filtering to generate a hybrid model for recommendation.
- Integration with existing music systems (iTunes, Google music, etc).
- Comparison of various techniques we used in music recommender system.
- Recommending songs based on the category with mood tags such as "sad", "happy", "joyful", "romantic", etc.

3.5 Evaluation

To evaluate how well our system would perform, we can make use of different metrics like Mean Squared Error, Root Mean Squared Error, precision and recall or DCG. We will split the input data into training set and test set.

$$RMSE = \sqrt{\frac{1}{n}\sum_{p,q}|P_{u,v} - R_{u,v}|^2}$$

The test set should be only a small fraction of the complete data. The idea is to hide the test set from the system and see how well the system is able to reproduce it. We will use the training set as input data to the recommender system to build the classifier. The system would then produce the recommendations, which we will compare to what the user actually listened to and use the above mentioned metrics to compute a accuracy of our model.

4. MILESTONES

Below are the proposed miles for Hybrid Music recommender system.

- April 10th, 2016 - Integrate Collaborative Filtering with Content based filtering to generate a hybrid model for recommendation.
- April 15th, 2016 - Comparison of the techniques proposed with existing recommendation systems.
- April 20th, 2016 - Project documentation and reporting.

Future Scope

- Implement Music Recommendation system using Deep learning techniques.
- Recommend songs based on what kind of music a user likes to listen during different time periods in a day.
- Mood tags "sad", "happy", "joyful", "romantic", etc. are not available in MSD. The approach would be to manually tag the songs with the moods and build a model which can be used for recommendation of songs based on moods.
- Integration with existing music systems(iTunes, Google music, etc).

5. PEER EVALUATION

The peer evaluation was done in collaboration with project team working on - 'Mining stack overflow data'. Authors: Nikhil Mahendra, Sachin Muralidhara, Aadish, Saurabh sood.

The goal of peer review is to help improve classmate's work by pointing out strengths and weaknesses that may not be apparent. During this session we had to answer several questions and few of the interesting ones are listed below.

What data is used to train the model for genre Classification?

As we know in Million songs dataset, we do not have an explicit attribute genre for all the songs in the dataset. There is a small dataset with genre attributes (the GTZAN dataset). We intend to classify the songs in MSD based on genre using a cross-modal retrieval framework which combines features of lyrics and audio available in MSD. Our likelihood-based training, combining audio sequence features and text features, is done using techniques like Hidden markov model(HMM) and Bag of words model for lyrics of the songs available in MSD.

What is unique about our approach in Hybrid recommendation?

Popular companies have been using collaborative filtering or content based filtering to come up with the recommendations. We propose a hybrid music recommender system that theoretically integrates collaborative data (rating scores of users) and content-based data (acoustic features of audio signals) to meet the our requirements. To integrate these methods (Collaborative and Content based filtering), we first represent user preferences by using content-based data and then make recommendations in a collaborative way by computing the similarities of the music based user preferences.

For this we use a probabilistic generative model, called a three-way aspect model, proposed by Popescul et al. which explains the probabilistic generative mechanism for the observed data (rating scores and acoustic features) by introducing a set of latent variables. As part of the generative mechanism, the model directly represents user preferences (latent favorite genres) estimated statistically with a theoretical proof. This estimation makes the method after predicting unknown rating scores by using content what techniques do we use to generate the recommendations.

What are the evaluation metrics?

To evaluate the performance of the system, we plan to make use of different metrics like Mean Squared Error, Root Mean Squared Error, precision and recall or DCG. We split the input data into training set and test set. The test set should be only a small fraction of the complete data. The idea is to hide the test set from the system and see how well the system is able to reproduce it. We use the training set as input data to the recommender system.

The system would produce the recommendations, which we compare with the user history and use the above mentioned metrics to compute a value. Also we would be comparing the accuracy measure with previously existing systems whose accuracy is expressed in terms of RMS Error.

6. CONCLUSIONS

In this project, we propose a hybrid model for music recommendation, which combines features from audio and lyrics for the task of music genre classification and recommendation. HMM's, lyrics bag-of-words, and loudness/tempo features would be used. Here we proposed a hybrid music recommender system that ranks musical pieces by comprehensively considering collaborative and content-based data, i.e., rating scores derived from users and acoustic features derived from audio signals.

We proposed a probabilistic generative model called a three-way aspect model to create our model which can theoretically explain the generative mechanism for both kinds of the observed data.One possible interpretation of the generative mechanism is that a user stochastically selects a genre according to his or her preferences and then the genre stochastically generates a musical piece and an acoustic feature.

Thus the joint probability distribution over users, pieces, and features is decomposed into three independent distributions, which are respectively conditioned by genres.This allows us to incrementally train the aspect model according to the increase in users and rating scores at low computational cost.

Thus main gist of the project can be summarized below -

- From features of audio and lyrics predict the genre of the music which is used for classification and recommendation.
- A Hybrid Recommender system which offers more accurate recommendations than existing systems.
- Apply Deep Learning techniques for computing the similarity in musicals (Future work).
- Compare the accuracy of the the different techniques with few evaluation metrics like RMS error etc.

7. REFERENCES

[1] Dawen Liang, Haijie Gu and Brendan OC̀onnor. *Music Genre Classification with the Million Song Dataset.*

[2] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata and Hiroshi G. Okuno. *An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model.*

[3] Aāron van den Oord, Sander Dieleman, Benjamin Schrauwen. Electronics and Information Systems department (ELIS), Ghent University *Deep content-based music recommendation*

[4] Yading Song, Simon Dixon, and Marcus Pearce. Centre for Digital Music. Queen Mary University of London. *A Survey of Music Recommendation Systems and Future Perspectives*

[5] Asela Gunawardana and Christopher Meek. Microsoft Research, One Microsoft Way, Redmond, WA 98052. *A Unified Approach to Building Hybrid Recommender Systems*

[6] Data Source - http://labrosa.ee.columbia.edu/millionsong

[7] Project Repository - https://github.com/Sunhick/music-cognita