**Deep content-based music recommendation**

Mid Semester Major Project

Under the supervision of Dr. Minakshi Sharma, Assistant Professor

Ahmad Ali Abdilah (2140012), Ujwal Sasikumar (2130008), Kapil Ramnani (2130011)

**Abstract**: Automatic music recommendation has become an increasingly relevant problem in recent years, since a lot of music is now sold and consumed digitally. Most recommender systems rely on collaborative ﬁltering. However, this approach suffers from the cold start problem: it fails when no usage data is available, so it is not effective for recommending new and unpopular songs. In this paper, we propose to use a latent factor model for recommendation, and predict the latent factors from music audio when they cannot be obtained from usage data. We use the spectrogram representation of the audio signals with deep convolutional neural network on combined dataset from various sources. We show that using predicted latent factors produces sensible recommendations, despite the fact that there is a large semantic gap between the characteristics of a song that affect user preference and the corresponding audio signal.

**1. Introduction**

In recent years, the music industry has shifted more and more towards digital distribution through online music stores and streaming services such as iTunes, Spotify, Grooveshark and Google Play. As a result, automatic music recommendation has become an increasingly relevant problem: it allows listeners to discover new music that matches their tastes, and enables online music stores to target their wares to the right audience. Although recommender systems have been studied extensively, the problem of music recommendation in particular is complicated by the sheer variety of different styles and genres, as well as social and geographic factors that inﬂuence listener preferences. The number of different items that can be recommended is very large, especially when recommending individual songs. This number can be reduced by recommending albums or artists instead, but this is not always compatible with the intended use of the system (e.g. automatic playlist generation), and it disregards the fact that the repertoire of an artist is rarely homogenous: listeners may enjoy particular songs more than others. Many recommender systems rely on usage patterns: the combinations of items that users have consumed or rated provide information about the users’ preferences, and how the items relate to each other. This is the collaborative ﬁltering approach. Another approach is to predict user preferences from item content and metadata. The consensus is that collaborative ﬁltering will generally outperform content-based recommendation [1]. However, it is only applicable when usage data is available. Collaborative ﬁltering suffers from the cold start problem: new items that have not been consumed before cannot be recommended. Additionally, items that are only of interest to a niche audience are more difﬁcult to recommend because usage data is scarce.

**1.1. Content-based music recommendation**

Music can be recommended based on available metadata: information such as the artist, album and year of release is usually known. Unfortunately this will lead to predictable recommendations. For example, recommending songs by artists that the user is known to enjoy is not particularly useful. One can also attempt to recommend music that is perceptually similar to what the user has previously listened to, by measuring the similarity between audio signals [2, 3]. This approach requires the deﬁnition of a suitable similarity metric. Such metrics are often deﬁned ad hoc, based on prior knowledge about music audio, and as a result they are not necessarily optimal for the task of music recommendation. Because of this, some researchers have used user preference data to tune similarity metrics.

**1.2. Collaborative ﬁltering**

Collaborative ﬁltering methods can be neighbourhood-based or model-based [4]. The former methods rely on a similarity measure between users or items: they recommend items consumed by other users with similar preferences, or similar items to the ones that the user has already consumed. Model based methods on the other hand attempt to model latent characteristics of the users and items, which are usually represented as vectors of latent factors. Latent factor models have been very popular ever since their effectiveness was demonstrated for movie recommendation in the Netﬂix Prize.

**1.3. The semantic gap in music**

Latent factor vectors form a compact description of the different facets of users’ tastes, and the corresponding characteristics of the items. This representation is quite versatile and can be used for other applications besides recommendation. Since usage data is scarce for many songs, it is often impossible to reliably estimate these factor vectors. Therefore it would be useful to be able to predict them from music audio content. There is a large semantic gap between the characteristics of a song that affect user preference, and the corresponding audio signal. Extracting high-level properties such as genre, mood, instrumentation and lyrical themes from audio signals requires powerful models that are capable of capturing the complex hierarchical structure of music. Additionally, some properties are impossible to obtain from audio signals alone, such as the popularity of the artist, their reputation and and their location. Researchers in the domain of music information retrieval (MIR) concern themselves with extracting these high-level properties from music. They have grown to rely on a particular set of engineered audio features, such as mel-frequency cepstral coefﬁcients (MFCCs), which are used as input to simple classiﬁers or regressors, such as SVMs and linear regression [5]. Recently this traditional approach has been challenged by some authors who have applied deep neural networks to MIR problems.

**1.4. Predicting latent factors from music audio**

Predicting latent factors for a given song from the corresponding audio signal is a regression problem. It requires learning a function that maps a time series to a vector of real numbers. Any traditional regression technique can then be used to map this feature representation to the factors. In this project we use a deep convolutional network.

**1.5. Convolutional neural networks**

Deep learning has become a popular topic talked by machine learning practicioners. One of the advantage of deep learning is its capability to perform complex operation. And also it elliminates the requirement of defining explicitly features of data makes it suitable to be used in a complex data.

Convolutional neural networks (CNNs) have recently been used to improve on the state of the art in speech recognition and large-scale image classiﬁcation by a large margin [6, 7]. It is efficient in doing its operation and also accurate when used to work with complex data such as image and audio.

Under several considerations mentioned ealier, we decided to employ deep convolutional neural networks into our project to serve the mean of giving song recommendation.

**2. Related Work**

So far, we have implemented 5 models in our project:

1. Popularity based collaborative filtering model. This model is giving recommendation to users merely based on the listening counts of each songs (popularity).
2. Item based collaborative filtering. In this we employ coocurance matrix to check the correlation of each songs.
3. Collaborative filtering using k-NearestNeighbor algorithm. The model will map each song in a space, then it returns the k-nearest song to the query song. The parameter we use in this model is listening count.
4. Collaborative filtering using Matrix Factorization.
5. Deep content-based music recommendation system [10]. This model is still under construction. The audio file preprocessor (to acquire MFCC representation of audio file) and recommender part of the model is done, only the neural network to obtain tags of song is still under work.

**3. Implementation**

In general, the steps of how the system works is as follows:

1. First of all, the input song file have to be converted into time-frequency representation as the input to the neural network. The representation we use is Mel-Frequency Cepstral Coefﬁcients (MFCCs). It is achieved by:
   1. Take the Fourier transform of (a windowed excerpt of) a signal.
   2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
   3. Take the logs of the powers at each of the mel frequencies.
   4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
   5. The MFCCs are the amplitudes of the resulting spectrum.
2. After that, the song representation is fed into neural network to acquire latent factors of the song (in this case predicting tags which are suitable for each song).
3. The tags are then associated to the song and the pair (song and tags) are run along with the user profile (listening history of the user) into another process to recommend a list of song which content is similar with the input song. The reommendation is happening as follows:
   1. Term Frequency (TF) of the song tags appering in each song is counted.
   2. Inverse Document Frequency (IDF) of tags appearing in songs are calculated.
   3. Similarity measure is used to calculate the distance of each song.
   4. Based on the distance, recommendation of songs are given.

The implementation of the neural network broadly is divided into 2 stages: training and prediction. In the training phase, the neural network is exposed to a set of songs with their associated tags so that the neural network will be able to give appropriate prediction in the future. While in the prediction stage, the user a song file is given to the neural network and it will analize the latent factors of the song and output the predicted tags for the song.

The dataset we use to implement this paper is based on several sources:

* A subset of Million Song Dataset (MSD) provides 10,000 metadata of songs , users and user-song listening data.
* Last.FM has tags information of songs provided in Million Song Dataset (MSD)
* Spotify API allows people to access preview of songs (30 seconds) that will be helpful to act as our raw song file data.

**4. Future Scope**

1. Running the neural network model on more dataset to increase its ability to predict tags.
2. Popularity prediction of a song.
3. Combining the content based recommender we have with collaborative filtering technique to achieve to varied recommendations.

**5. References**

1. M. Slaney. Web-scale multimedia analysis: Does content matter? *MultiMedia, IEEE*, 18(2):12–15, 2011.
2. Malcolm Slaney, Kilian Q. Weinberger, and William White. Learning a metric for music similarity. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, 2008.
3. Jan Schluter¨ and Christian Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine. In *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA)*, 2011.
4. Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
5. Eric J. Humphrey, Juan P. Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, 2012.
6. Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 2012.
8. Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.
9. James Bergstra, Olivier Breuleux, Fred´ eric´ Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.
10. Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13), C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), Vol. 2. Curran Associates Inc., USA, 2643-2651.