Discovery Oriented Model Selection in Music Recommendation

Shubham Gautam
Gaana
Noida, India
shubham.gautam@gaana.com

Pallavi Gupta
Gaana
Noida, India
pallavi.gupta@gaana.com

Satyajit Swain
Gaana
Noida, India
satyajit.swain@gaana.com

Rohit Ranjan
Gaana
Noida, India
rohit.ranjan@gaana.com

ABSTRACT

Music streaming services of the likes of Spotify[1], Apple Music[2], Gaana[3], JioSaavn[4], YouTube Music[5], Wynk[6] *etc.* have been the go-to platform choices for music listening in the Indian entertainment ecosystem spanning 16+ vernacular regional languages besides English language music. In such a diverse market with huge music catalog repository across multiple languages, enabling music discovery is a challenge by relying purely on recommendations based on user listening history. In this paper, we present an approach to improve music discovery in a language including long tail content discovery and improving relevancy ranking of songs in a given recommendation set through audio fingerprinting based techniques that tap into the inherent characteristics of a music audio file. We illustrate that model selection can be done in a suitable manner using appropriate parameter tuning strategies in machine learning based models by showing their effects on online as well as offline experimentation. We also present our evaluation approach on offline metrics and present our conclusion on the trade-off and balancing that needs to be considered as there is no one size fits all approach applicable in the diverse nature of music.

Keywords

Recommendation systems, model selection, audio processing, parameter tuning, signal processing, machine learning, deep learning, offline evaluation

1. INTRODUCTION

Audio signal processing is a sub-field of signal processing that is concerned with the electronic manipulation of audio signals. Recommendation system deals with suggesting a new item corresponding to the user consumption behaviour. Many recommendation systems focus on collaborative filtering kind of approaches in which users' crowd-sourced information is considered to provide new set of recommendations. However, these systems fall short of content discovery aspect and as they provide only those items which are frequently been consumed together among a large set of users. Often, this takes a user into a self serving loop in which user keeps listening to the same set of tracks and misses out on the discovery angle. Either user would have to search if he/she knows what to search for. But, in most of the cases, user prefers to see a new likeable item to be seen on its own because user also has some defined knowledge to what to search for.

Discovery and novelty in recommendation systems have gained popularity in recent years to improve long tail content discovery and serve relevant content from catalog repository to users which they do not get exposed to very frequently or have not consumed earlier. Most existing CF approaches to music recommendation are neighborhood-based, computing similarity between users or items. Alternatively, there were proposed model-based algorithms, in particular latent factor models based on matrix factorization techniques where users and items are simultaneously represented as feature vectors in a latent feature space learnt from the user-item preference data.

Model selection is an area that requires special attention in recommendation systems. Parameter tuning of an algorithm plays a vital role for achieving this objective as certain parameters of an algorithm giving best set of results might not perform well on other sets of data. There are standard approaches used for model selection such as: Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) *etc.*, both of these try to select inductive bias by penalizing model fit. We did not use these as such methods do not fit to our use case where we are trying to increase discovery in an unsupervised setting.

We present an approach that uses only raw audio features to predict similarity among songs that is independent of user interaction and can be used to provide discovery to end users on the platform. We emphasize the focus on selection of an appropriate model through ablation testing and parameter tuning in deep neural networks and how this can be validated in offline evaluation. This paper is organized as follows: section 2 presents the related work. Section 3 discusses about audio fingerprinting and its challenges. Section 4 demonstrates about how we have done audio features extraction. Section 5 focuses on model selection while experiments and results are provided in section 6.

[1]https://open.spotify.com
[2]https://www.apple.com/in/apple-music/
[3]https://gaana.com
[4]https://www.jiosaavn.com/
[5]https://music.youtube.com/
[6]https://wynk.in/music

2. RELATED WORK

For music recommendations, a lot of quantifiable work has been done by researchers for extracting audio fingerprint information from raw audio music files. This section highlights some of the work done earlier in the field of audio signal processing for recommendation systems and how it has been used and evaluated offline for discovery and serendipity tasks while serving recommendation for an end user. [7] discusses about latent factors from music audio using deep neural networks. [14] demonstrated a method of successfully injecting serendipity, novelty and diversity into recommendations whilst limiting the impact on accuracy. [1] talks about how social media information can be incorporated while serving music recommendations. The importance of serendipity in information systems was first recognized by [11]. [6] casually defined serendipitous recommendations as "surprisingly interesting items the user might not have otherwise discovered". [2] and [12] talked about novelty and discovery in recommendation systems and discussed different kinds of models and evaluation metrics for them. [13] provided an in-depth overview of how novelty can be defined and judged in recommendation systems. In the context of Indian music, there is some work which has been done to predict mood of a song. [10] provides a way to predict mood of Hindi songs. [9] tries to identify mood of Hindi songs from lyrics. Mood classification using unsupervised approach has been presented in [8].

Second part of this paper focuses on model selection of recommendation systems through offline evaluation that has seen tremendous development in the past decade. Previously, people used to rely only on metrics after making algorithm live on platforms but it becomes hectic to measure every algorithm with tuning it on offline platform. [4] highlighted evaluation of recommendation in offline manner and how user feedback is incorporated. [5] discussed about A/B testing in offline scenario. [3] presented an offline evaluation framework for recommendation systems based on probabilistic relational models. Our work focuses on the extraction of different kinds of audio features and how that can be used for discovery oriented model selection using offline evaluation and experimentation as well as showing its effect on live population.

3. AUDIO FINGERPRINTING (AFP)

An audio fingerprint is a numeric representation of an audio file encapsulating inherent characteristics pertinent to the audio that it represents. The basic premise here is to capture the distinct digital signature of a sound, so as to be able to differentiate this with other sounds based on the difference in signatures. As per Wikipedia, "An acoustic fingerprint is a condensed digital summary, a fingerprint, deterministically generated from an audio signal, that can be used to identify an audio sample or quickly locate similar items in an audio database". Audio fingerprinting techniques extract the auditory relevant attributes of an audio content.

3.1 Challenges

This section discusses about the challenges in audio fingerprinting. There are some challenges in AFP which are as follows:

- Fingerprint should be *discriminative* in order to avoid false positives
- *Cropping* of audio to extract meaningful information can sometime lose important data
- *Efficiency* of computing fingerprint should also be considered. It should not be a time intensive process.

4. AUDIO FEATURES EXTRACTION

This section discusses about the various methodologies adopted in our quest to extract a set of mutually exclusive and collectively exhaustive list of fingerprint features from an audio file.

4.1 MFCC (Mel-frequency cepstral coefficients)

In sound processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC.[7]

4.2 Chorus

In this context, 'Chorus' is defined as the most repetitive set of frames from an audio file. Pychorus[8] was used (with modification as per our usecase) to extract chorus from the audio data. Once we get chorus from the audio file, then its MFCC coefficients would be extracted out for further processing. *So how is this different from the previous approach in which we were extracting MFCC coefficients?* Here, we'll be getting MFCC coefficients for a much smaller window (10-15 seconds out of the total duration of the audio) while in previous section, we were considering the whole audio file for getting the coefficients.

4.3 1D Auto Encoder

Autoencoder is an unsupervised artificial neural network that is used to efficiently compress and encode data then learns how to reconstruct the data back from the reduced encoded representation to a representation that is as close to the original input as possible. For our usecase, we used 50 subsequent windows (where 1 window is the collection of 4-5 sec of frames) of an audio file and their 20 MFCC coefficients. This way, we made 50 x 20, *i.e.* 1000 dimensional input data (which in turn is same as output data to be learnt through network).

Table 1 shows the architecture of our 1D auto encoder. From the above network, we get a feature vector of length 10 of an audio file from 4th layer.

$$WD = N/50$$

WD is the average window duration of number of frames to be considered

N is the length of audio file

$$W_{MFCC} = \frac{1}{N}\sum_{i=1}^{WD} MFCC_i$$

W_{MFCC} is the average MFCC vector of all frames in the window length WD

[7] https://en.wikipedia.org/wiki/Mel-frequency_cepstrum
[8] https://github.com/vivjay30/pychorus

Table 1: 1D Auto Encoder Architecture

Layer (type)	Output Shape	Param
$\text{input}_1(InputLayer)$	1000	0
$\text{dense}_1(Dense)$	500	500500
$\text{dense}_2(Dense)$	100	50100
$\text{dense}_3(Dense)$	10	1010
$\text{dense}_4(Dense)$	100	1100
$\text{dense}_5(Dense)$	500	50500
$\text{dense}_6(Dense)$	1000	501000
Total params: 1,104,210		
Trainable params: 1,104,210		
Non-trainable params: 0		

$MFCC_i$ is the MFCC vector of i_{th} frame

Thus, we can formulate the audio vector as follows (which is a collection of 50 subsequent windowed frames):

$$AV = [W_{MFCC1}, W_{MFCC2}, ..., W_{MFCC50}]$$

where, AV is the audio vector for a song

4.4 Convolutional Auto Encoder (CAE)

CAEs are a type of Convolutional Neural Networks (CNNs). The main difference between the common interpretation of CNN and CAE is that the former is trained end-to-end to learn filters and combine features with the aim of classifying their input. In fact, CNNs are usually referred to as supervised learning algorithms. The latter, instead, are trained only to learn filters able to extract features that can be used to reconstruct the input. Convolutional Neural Network (CNN) is generally used for feature extraction from images. We used it for extracting abstract representation of audio features. For that, 48 subsequent windows of audio files have been considered (1 window was a collection of 4-5 second frames) and their 16 MFCC coefficients were taken into account. This way, we have an input of dimension 48 x 16 as well as the output of the same dimension. Table 2 shows the architecture of our CAE based network. This is a 17 layer (inclusive of input and output layers) network where 9th layer is used for flattening the output given by 8th layer which has been used to determine the abstract representation of the audio data. This network uses 'Relu' as the activation function in all layers. From this network, we get a feature vector of length 12 of an audio file.

4.4.1 Architecture Diagram

Table 2 shows the layer by layer architecture of convolutional auto encoder.

The CAE architecture has shared weights just like CNN. For an input x the latent representation of the k-th feature map is given by:

$$h_k = \sigma(x * W^k + b^k)$$

where, the bias is broadcasted to the whole map, σ is an activation function (*relu* is our case), $*$ denotes the 2D convolution

A single bias per latent map is used, as we want each filter to specialize on features of the whole input. The reconstruction is obtained using:

Table 2: Convolutional Auto Encoder Architecture

Layer (type)	Output Shape	Param
input_1	(48, 16, 1)	0
conv2d_1	(48, 16, 8)	80
max_pooling2d_1	(24, 8, 8)	0
conv2d_2	(24, 8, 16)	1168
max_pooling2d_2	(12, 4, 16)	0
conv2d_3	(12, 4, 32)	4640
max_pooling2d_3	(6, 2, 32)	0
conv2d_4	(6, 2, 1)	289
flatten_1	(12)	0
reshape_1	(6, 2, 1)	0
conv2d_5	(6, 2, 1)	10
up_sampling2d_1	(12, 4, 1)	0
conv2d_6	(12, 4, 32)	320
up_sampling2d_2	(24, 8, 32)	0
conv2d_7	(24, 8, 16)	4624
up_sampling2d_3	(48, 16, 16)	0
conv2d_8	(48, 16, 1)	145
Total params: 11,276		
Trainable params: 11,276		
Non-trainable params: 0		

$$y = \sigma\left(\sum_{k \in H} h^k * \tilde{W}^k + c\right)$$

where, again there is one bias c per input channel. H identifies the group of latent feature maps; W identifies the flip operation over both dimensions of the weights.

The cost function to minimize is the mean squared error (MSE):

$$E(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(x_i - y_i)^2$$

Likewise in other kinds of networks, the backpropagation algorithm is applied to compute the gradient of the error function with respect to the parameters. This can be easily obtained by convolution operations using the following formula:

$$\frac{\partial E(\theta)}{\partial W^k} = x * \delta h^k + \tilde{h}^k * \delta y$$

where, δh and δy are the deltas of the hidden states and the reconstruction, respectively. The weights are then updated using stochastic gradient descent.

Max pooling was used for pooling layers of CAE and then layers were stacked using the formation depicted in Table 2.

4.5 Other Important Features

We used other audio features as well to extract meaningful information from our audio data:

- **Loudness:** It computes the loudness of an audio signal defined by Steven's power law. It computes loudness as the energy of the signal raised to the power of 0.67.
- **Tempo:** It is the speed or pace of a given piece and derives directly from the average beat duration. The

overall estimated tempo of a track in beats per minute (BPM).[9]

- **Pitch:** It is the fundamental frequency of an audio waveform.
- **Energy:** It is approximated by the root mean square (RMS) of the signal magnitude within each frame.
- **ZCR (Zero Crossing Rate):** It counts the number of times that the audio waveform crosses the zero axis.
- **Spectral Centroid:** The spectral centroid is a measure that indicates where the "center of mass" of the spectrum is. Perceptually, it has a robust connection with the impression of 'brightness' of a sound, and therefore is used to characterise musical timbre. It is calculated as the weighted mean of the frequencies present in the signal, with their magnitudes as the weights.[10]
- **Spectral RollOff:** This computes the roll-off frequency of a spectrum. The roll-off frequency is defined as the frequency under which some percentage (cutoff) of the total energy of the spectrum is contained. The roll-off frequency can be used to distinguish between harmonic (below roll-off) and noisy sounds (above roll-off).[11]
- **Chroma:** Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave.

5. MODEL SELECTION

This section discusses how a model can be selected out of a large number of models made during training phase and how tuning the parameters can be a viable option.

5.1 Parameter Tuning

Parameter tuning in machine learning plays a vital role in model selection for a particular objective. As illustrated in section 4, there is a wide variety of features being considered to solve our problem so it becomes essential to select a model which is tuned as per our use case. That could be achieved through rigorous set of experimentation that needs to be carried out for final model selection. There are following set of parameters which played important role for our model selection:

- **Activation Function:** Mainly 3 activation functions were considered:
  - ReLu (Rectified Linear Unit)
  - Sigmoid
  - Tanh

  Since we were building a deep neural network so activation function play its part considering number of layers in the network.
- **Iterations:** Number of iterations to be considered for training the network
- **Number of Clusters:** This was determined using elbow curve as discussed in next sub-section of 'Model Training'.
- Number of layers in deep neural network
- Number of hidden neurons/filters in each layer of the network

Above mentioned parameters were considered while training deep neural networks as mentioned in last section where we explained auto encoders for our use case. Next section discusses about various experiments performed.

5.2 Model Training

As we did not have any supervised data through which discovery/novelty can be learnt so we decided to go ahead with unsupervised learning by using k-means and KNN algorithms. Number of clusters for each model was decided using elbow curve as shown in Figure 1 for one of the model. *Out of the two auto encoder based architectures explained in the previous section, CAE was chosen as it was giving better MSE as compared to 1D AE.*

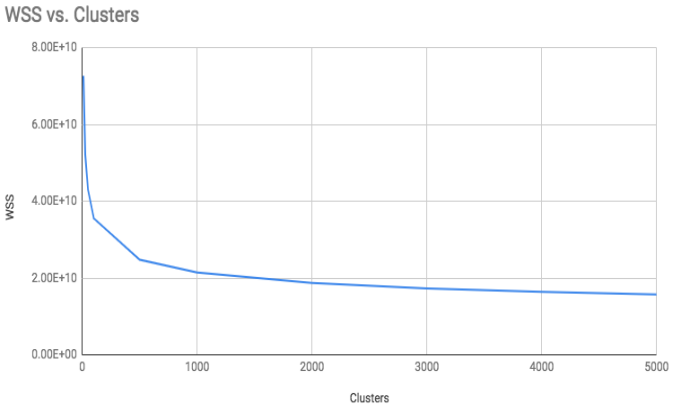

Figure 1: Cluster Selection using Elbow Curve

5.3 Ablation Testing

An ablation study refers to removing some feature of the model and seeing how that affects performance of the system as a whole. We've got a lot of features as discussed in section 4. As we worked in unsupervised domain, so this method provided a way to identify important set of features out of all considered features. Effects of ablation testing would be provided in next section where we evaluate our models on offline evaluation metrics as shown in Table 3.

6. EXPERIMENTS AND RESULTS

On the Gaana platform, whenever a played song is at the end of the queue then treating that song as a seed song, the platform automatically enqueues a set of recommended songs in the queue as the next best set of songs to be played by the user, we call this recommendation feature as 'Auto-Queue'.

[9]https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/
[10]https://essentia.upf.edu/documentation/reference/streaming_Centroid.html
[11]https://essentia.upf.edu/documentation/reference/streaming_RollOff.html

6.1 AFP Elevation in Existing Recommendation System

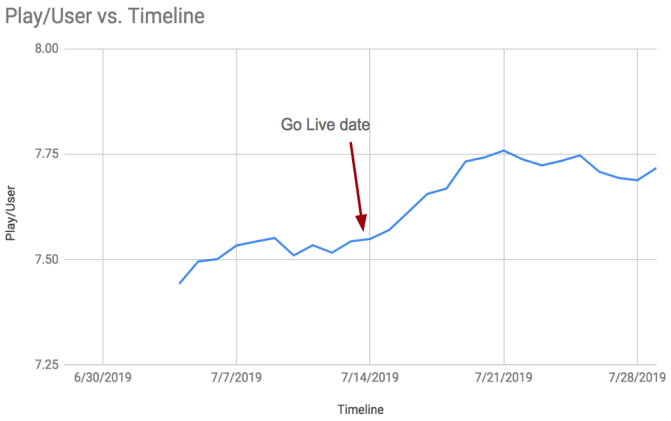

Figure 2: AFP Elevation in existing recommendation model

In India, Bollywood is the primary industry that drives movies and music consumption in the country. To test the efficacy of our AFP models, we decided to deploy AFP based discovery elevation for Bollywood Hindi music. In auto-queue functionality of Bollywood Hindi songs, based on the current seed song top 15 recommendations were being served using word2vec algorithm (trained on a set of 3 months user listening history). To improve the relevance within the result set of 15 songs, we applied AFP based sorting on the set of recommended songs to rank songs based on audio similarity with the seed song. Songs from the result set of 15 with higher AFP similarity score with the seed song were elevated to the top of the recommended set and were served to users through the Auto-Queue feature. Through this we were able to give boost to those songs which might not even be heard by users due to their lower positioning in the auto-queue.

Through this we observed that plays/user for users engaging through Auto-Queue for Bollywood Hindi language went up by $\approx 3.5\%$ (Figure 2), suggesting that user listening behavior improved and user was also open for discovery oriented tracks in their recommendation set.

6.2 Offline Evaluation Metrics

6.2.1 *NDCG (Normalized Discounted Cumulative Gain)*

NDCG is a popular method for measuring the quality of a set of search results. It is calculated as follows:

$$DCG = \sum_{pos=1}^{n} \frac{relevance_{pos}}{\ln(pos+1)} \tag{1}$$

where, pos is the position of the clicked item in the search results.

$$NDCG = \frac{DCG}{iDCG} \tag{2}$$

where, $iDCG$ is ideal DCG.

6.2.2 *Novelty*

Novelty has been defined in [13] very nicely and we took that metric to measure discoverability in our system. Metric is as follows:

Table 3: Offline Metrics for Model Selection of Hindi Songs

Model	*NDCG* based Ranking	*Novelty* based Ranking
MFCC	8	11
Chroma	6	10
CNN	7	9
MFCC + other features	4	8
MFCC + chroma + other features	1	7
Chroma + other features	2	6
Chorus + other features	3	5
CNN + other features	9	1
CNN + chroma + other features	5	2
MFCC + CNN + other features	11	3
MFCC + CNN + chroma + other features	10	4

$$Novelty(i,u) = p(i|unknown,u) * dis(i,pref_u) * p(i|like,u) \tag{3}$$

where, $p(i|unknown,u)$ is the possibility of an item being unknown,

$dis(i,pref_u)$ is the measure to calculate dissimilarity,

$p(i|like,u)$ is the measure of likeability of an item for a user.

6.3 Model Selection Trade-off

Table 3 illustrates ranking[12] of models based on different set of features with their corresponding *NDCG* and *Novelty* metrics. For evaluation of these metrics, we considered the 4 months user data in which previous 3 months data was taken as training set and remaining 1 month data was takes as test set. Values given in Table 3 provides metrics on test data. Section 5.3 illustrated the way these models have been built using ablation testing. We considered these two metrics to take care of trade-off between them as our objective is to introduce discoverability to the user's recommendations, so its essential to consider user behaviour as well as the novelty factor that we are trying to introduce. From an end user's perspective, we can not introduce a model which is getting highest novelty as it might hamper user's behaviour on the platform. This is because of showing much less popular song to the user ('novelty' consists of *popularity* as one of the factors), this might not look good to him/her.

6.4 Illustrative Examples

We considered two examples to show in this paper. One belongs to the dance category while other is of non-dance (slow) category. These examples have been considered to show the diversity of different models (which were selected out of 25 models using metrics defined earlier). Table 4 shows top similar songs corresponding to one of the famous

[12]Ranking was provided instead of scores due to different scale of the metrics.

Table 4: Similar songs for seed song: 'Dilliwali Girlfriend'

Model 1	Model 2	Model 3
Twist - Love Aaj Kal (Score: 0.9989)	Nachan Farrate - All Is Well (Score: 0.9978)	Te Amo - Dum Maaro Dum (Score: 0.9889)
Nachan Farrate - All Is Well (Score: 0.9984)	God Allah Aur Bhagwan - Krrish 3 (Score: 0.9967)	Aapka Kya Hoga - Housefull (Score: 0.9879)

Table 5: Similar songs for seed song: 'Kalank Title Track'

Model 1	Model 2	Model 3
Tum Ho - Rockstar (Score: 0.9977)	Yaaram - Ek Thi Daayan (Score: 0.9983)	Baaton Ko Teri - All Is Well (Score: 0.9981)
Tujhe Bhula Diya - Anjaana Anjaani (Score: 0.9973)	Hai Koi - Chor Bazaari (Score: 0.9978)	Hasi - Hamari Adhuri Kahani (Score: 0.9969)

dance song of Bollywood: 'Dilliwali Girlfriend'[13] (Album: Yeh Jawaani Hai Deewani) while Table 5 shows top similar songs for another Bollywood song (slow song) 'Kalank Title Track'[14] (Album: Kalank). All song names have been provided with their album names and similarity scores that these songs got from the respective model corresponding to the given seed song.

6.5 English Songs Experimentation

Taking a cue from our learnings in Hindi language music, we tried replicating the same success in English music song similarity based on auditory aspect by combining different features such as MFCC, CNN, chroma, loudness, BPM *etc.* Table 6 provides the similar songs for the seed song **Eastside**[15] (Benny Blanco *et al*). All song names have been provided with their artist names and similarity scores for the provided seed song. As we used similar approach for model training and model validation here as well, we found that the features that were playing major role in case of Hindi music, it might not be the case that those features would be performing better here also.

6.6 Findings and Analysis

The diversity and unique characteristics of Bollywood[16] (Indian Hindi language film industry) music make it challenging compared to English and Western music where homogeneity of song is a common pattern. Bollywood music, usually has become heterogeneous in nature over the past decade where a fusion of different music types such as EDM, rock, pop *etc.* are a common occurrence or trend. These nuances have kept varying from era to era due to the evolution of Bollywood Hindi music over the years. Keeping these unique challenges in consideration, we conducted manual testing of our AFP based song similarity models to validate them from qualitative aspect. Below are the findings based on our manual qualitative testing:

- The AFP model based on 'chorus' features performed well for dance songs released in the 2000 & above year era, as dance songs of this era are characterized by high tempo and repetitive patterns in the music.
- 'MFCC' features based model performed qualitative testing for non-dance songs released in the era of 2000 & above.
- A distinctive observation that we came across was 'CNN' based model was giving the best similarity results for Bollywood 90s era and Bollywood retro era songs. This may be a consequence of the way CNN was trained using convolution windows over subsequent frames of song, it is able to identify multiple repetitive patterns in a music file than just one dominant repetitive pattern (which is the case with 'chorus' based model). Additionally, retro songs in general do not have single chorus but have many patterns that are repeated periodically in the whole song.

One of the findings that we got in English songs is that integrating CAE based features gives better results across all eras and genres while in case of Hindi music, it was mostly limited to the songs of the era of 2000 & above.

6.7 Comparison of Bollywood and English Songs

English music traditionally has been known to follow a consistent rhythmic approach within a music track , whereas Bollywood Hindi music over a period of time especially has evolved into a fusion of sorts with many songs released in the post 2000 era have a mix of EDM, pop, rap, etc. fused in a single music. This presents a unique challenge in the Bollywood music space, where clustering of songs into distinctive genres based on AFP becomes tad difficult as there is no one size fits all approach. The challenges become even more pronounced in recently released Bollywood music songs where fusion of different genres in the same music track is becoming more rampant. To solve this through AFP has required us to experiment with different AFP modelling based approaches to tackle nuances of Hindi as well as English music from different eras.

7. FUTURE WORK

This work can be extended to quantify important features in songs such as: danceability, mood prediction *etc.* Similar kind of models will need to be built for other music languages including western music, Indian vernacular music as well as other global language music as same model can not be replicated across considering nuances keep differing as we have observed earlier for English and Hindi language songs.

[13]https://gaana.com/song/dilliwaali-girlfriend
[14]https://gaana.com/song/kalank-title-track
[15]https://gaana.com/album/eastside-english-1
[16]https://en.wikipedia.org/wiki/Bollywood

Table 6: Similar songs from different models for seed song: 'Eastside'

Model 1	Model 2	Model 3
Takin' Back My Love - Enrique Iglesias *et al* (Score: 0.9993)	I'm Yours - Jason Mraz *et al* (Score: 0.9989)	Takin' Back My Love - Enrique Iglesias *et al* (Score: 0.9981)
I Know You Want Me - Pitbull *et al* (Score: 0.9984)	Takin' Back My Love - Enrique Iglesias *et al* (Score: 0.9986)	Hotel California - Eagles *et al* (Score: 0.9978)

8. CONCLUSION

In this paper, we presented our approach of how audio similarity can be calculated based on raw inherent characteristics and features extracted directly from an audio music file. We discussed about challenges that can be there while working on such kind of problems. We discussed how MFCC based features can be used for this purpose and how deep learning based features can be explored in this approach using deep autoencoders (1D and CNN). We highlighted the importance of parameter tuning for selection of machine learning based model for our recommendation system. Offline evaluation is really important since it provides a way to measure the effectiveness of an algorithm before making it live and as we are working on discovery related tasks in recommendation systems so it becomes more important to measure the impact offline only. We have also highlighted the differences in approach adopted for music across different eras to take care of nuances of music evolution with time and generational changes in the diverse Bollywood music ecosystem. Consequently, we discussed the trade-off between two metrics: 'NDCG' and 'Novelty', because in such kind of tasks we can not rely on user data only as users might not have consumed or interacted with items which might be of interest to them.

9. REFERENCES

[1] J. Bu, S. Tan, C. Chen, C. Wang, H. Wu, L. Zhang, and X. He. Music recommendation by unified hypergraph: Combining social media information and music content. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 391–400, New York, NY, USA, 2010. ACM.

[2] P. Castells, S. Vargas, and J. Wang. Novelty and diversity metrics for recommender systems: Choice, discovery and relevance. *Proceedings of International Workshop on Diversity in Document Retrieval (DDR)*, January 2011.

[3] R. Chulyadyo and P. Leray. A Framework for Offline Evaluation of Recommender Systems based on Probabilistic Relational Models. Technical report, Laboratoire des Sciences du Numérique de Nantes ; Capacités SAS, Dec. 2017.

[4] A. Ferraro, D. Bogdanov, K. Choi, and X. Serra. Using offline metrics and user behavior analysis to combine multiple systems for music recommendation. *CoRR*, abs/1901.02296, 2019.

[5] A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham, and S. Dollé. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 198–206, New York, NY, USA, 2018. ACM.

[6] L. G. T. J. L. Herlocker, J. A. Konstan and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 2004.

[7] A. v. d. Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pages 2643–2651, USA, 2013. Curran Associates Inc.

[8] B. Patra, D. Das, and S. Bandyopadhyay. *Unsupervised Approach to Hindi Music Mood Classification*, volume 8284, pages 62–69. December 2013.

[9] B. Patra, D. Das, and S. Bandyopadhyay. Mood classification of hindi songs based on lyrics. December 2015.

[10] B. G. Patra, D. Das, and S. Bandyopadhyay. Automatic music mood classification of Hindi songs. In *Proceedings of the 3rd Workshop on Sentiment Analysis where AI meets Psychology*, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing.

[11] E. G. Toms. Serendipitous information retrieval. *In Procs. of 1st DELOS Workshop on Information Seeking, Searching and Querying in Digital Libraries*, pages 11–14, June 2000.

[12] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 109–116, New York, NY, USA, 2011. ACM.

[13] L. Zhang. The definition of novelty in recommendation system. *Journal of Engineering Science and Technology Review*, 6:141–145, June 2013.

[14] Y. C. Zhang, D. O. Séaghdha, D. Quercia, and T. Jambor. Auralist: Introducing serendipity into music recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pages 13–22, New York, NY, USA, 2012. ACM.