A Machine Learning based Music Retrieval and Recommendation System

Naziba Mostafa, Yan Wan, Unnayan Amitabh, Pascale Fung
Human Language Technology Center
Department of Electronic and Computer Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
nmostafa@connect.ust.hk, ywanad@connect.ust.hk, uamitabh@connect.ust.hk, pascale@ece.ust.hk

Abstract

In this paper, we present a music retrieval and recommendation system using machine learning techniques. We propose a query by humming system for music retrieval that uses deep neural networks for note transcription and a note-based retrieval system for retrieving the correct song from the database. We evaluate our query by humming system using the standard MIREX QBSH dataset. We also propose a similar artist recommendation system which recommends similar artists based on acoustic features of the artists' music, online text descriptions of the artists and social media data. We use supervised machine learning techniques over all our features and compare our recommendation results to those produced by a popular similar artist recommendation website.

Keywords: query by humming, similar artist recommendation, music information retrieval, machine learning

1. Introduction

Faster computational speed and increasing number of online users have resulted in a dramatic increase in music consumption. It is getting more and more difficult for the general public, especially non-experts, to find and retrieve music from the millions of songs available online. A lot of research is being done these days to find efficient music retrieval and recommendation methods. One music retrieval method that is gaining a lot of popularity these days due to its convenient usage is query by humming, which is a content-based music retrieval method that can retrieve melodies using users' hummings as queries. This allows users to find old songs that they only remember the tune of or retrieve obscure songs heard in public places. The Music Information Retrieval (MIR) community has also been doing a lot of work on automatic recommendation systems ranging from the content-based methods to social tagging and similarity networks (Cohen and Fan, 2000; Hong et al., 2008). One of the key research topics in this area that has gained a lot of traction is automatic similar artist recommendation.

Currently, there are several musical retrieval and similar artist recommendation apps. There are apps such as SoundHound, MusixMatch etc, that can retrieve songs using humming as a query, and websites such as All Music Guide (AMG)[1] and last.fm[2] that give similar artist recommendation. However, accuracy and efficiency of these music retrieval and recommendation systems still leave a lot of room for improvement. Therefore, we are planning to create a holistic music retrieval and recommendation system using machine learning techniques.

The biggest challenges of a query by humming system include i) queries sung by users often vary from the actual melody in pitch, tempo etc. so the melodic similarity matching must be done at a more abstract level in order to get meaningful results, ii) background noise is often present in users' queries which also makes it harder to identify the melody correctly and iii) efficient retrieval methods must be used that can search through a database and retrieve the correct melody in as little time as possible. Therefore, methods used to retrieve the melody in this case need to be robust to noise and inaccuracies in the singing or humming which is very challenging, and, for the system to be practical, the entire system should be very fast.

Therefore, we propose a supervised machine learning based method for the query-by-humming system, which can learn the common errors associated with human humming and build a model that is unaffected by these errors. For this task, we have collected humming data and transcribed them in order to train a Deep Neural Network (DNN) based Hidden Markov Model (HMM) for note transcription. This deep learning method allows us to learn the patterns present in the humming data and create a model that can detect the notes in a humming query. The proposed note transcription method is used along with a note-based retrieval method similar to Yang et al. (2010) in order to retrieve a ranked list of songs most similar to the query.

One of the biggest challenges faced by the current similar artist recommendation systems is that they perform poorly for relatively obscure artists. Therefore we are interested in using machine learning methods to build a recommendation system that can provide good similar artist recommendation even for relatively unknown artists. We propose a recommendation system that uses supervised machine learning techniques over features such as acoustics of music, the meta-data, and online texts related to the artist to find similar artists.

2. Previous Work

The main components of a QBH system consist of i) representation of the query and actual songs and ii) retrieval the songs efficiently and accurately from the database. A song or a query is mainly represented using frame-based and note-based methods. The frame-based methods use a representation of the extracted pitch to represent the query and the songs and then use some template-matching similarity measures such as DTW (Dynamic time warping) to

[1] http://www.allmusic.com
[2] http://www.last.fm/

measure similarity between the main songs and the query (Wang et al., 2008; Dannenberg et al., 2007). The note-based methods use the pitches as features to transcribe the notes present in a query or a song (Shih et al., 2002; Shih et al., 2003; Shifrin et al., 2002) and the notes in the query are then matched against the notes in songs using simple string matching techniques (Ghias et al., 1995; Shih et al., 2002) or linear scaling based methods (Yang et al., 2010).

In this paper, we focus on note-based methods since they are more efficient (Kharat et al., 2015; Yang et al., 2010) and there still seems to be room for improvement in accuracy in this case. These methods are often based on statistical approaches. Hidden Markov model (HMM) is one of the common methods that have been used for note transcription (Shih et al., 2002; Shih et al., 2003; Shifrin et al., 2002; Ryynänen and Klapuri, 2004). In Shih et al. (2002) and Shih et al. (2003), the note is segmented by modelling phonemes using mel-frequency cepstral coefficients, energy measures, and the derivatives of these as features, and then the average pitch of the note is found on the segmented segments, which is then used to represent the individual notes. However, this approach works well when each note is hummed using one syllable such as da or ta and is not very effective for handling a large variety of queries. The most effective of these statistical melody transcription approaches is proposed in Ryynänen and Klapuri (2004), which extracts prosodic features, that are used to train HMM-GMMs for modelling notes. Since recent studies in speech recognition field have shown that using DNN instead of GMMs in HMM significantly improves the recognition accuracy (Dahl et al., 2012), we propose to use Deep Neural Networks (DNN) with HMMs instead of simple HMM-GMM models as humming is similar to speech.

The methodology used in our artist recommendation system differs from previous work both in the source and target. We collect tons of news from mainstream news websites and calculate the co-occurrence of Bollywood artists' names in these articles, which is a plausible profound and comprehensive way to tell the relativeness of two singers.

On the other hand, related artists ought to influence each other in their musical style. Therefore, we are also extracting audio-based features to find related artists. Voice features such as Mel Frequency Cepstrum Coefficients (MFCCs) (Mermelstein, 1976) are wildly used in speech recognition and audio fingerprinting (Cano et al., 2005). Features of MFCCs include spectral flatness, tone peaks, which could represent the features and categories of the songs. In addition, musical features like loudness, pitch and brightness are also used for query of music (Wold et al., 1996). Su et al. (2013) have previously investigated piece-level features for determining the mood of a musical piece with high accuracy.

The rest of the paper is organized as follows. Section 3 describes the overall methodology, Section 4 describes the query by humming system, Section 5 describes the similar artist recommendation system, Section 6 explains the experimental setup and evaluation of the system and Section 7 summarizes the content of the paper

3. Methodology

The overall system takes a hummed tune as an input, which is then fed to the Query by Humming (QBH) system. The QBH system uses the input to output a ranked list of songs with highest similarities to the query. The user can then either manually choose the correct song from the ranked list or use the default setting of choosing the most highly ranked song as the song to be retrieved. The retrieved song along with its metadata is then used as an input to the similar artist recommendation system, which then outputs a list of most similar artists. An overview of the overall system is given below in Figure 1.

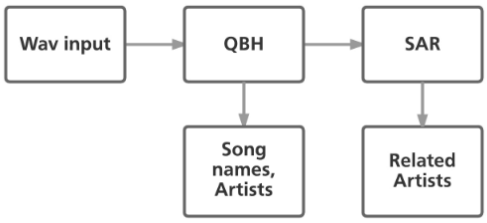

Figure 1: Overview of the overall music retrieval and recommendation system

The Query by Humming system and similar artist recommendation system are described in more detail in sections 4 and 5 respectively.

4. Query by Humming

The Query by Humming system takes a few notes from a melody hummed or sung by the user as the query. The notes of the query is transcribed using our note transcription method and is then passed onto the retrieval system, which uses the transcribed query and the melody database, which refers to the entire list of pre-transcribed melodies or songs that can be recognized by our system, to give a ranked list of melodies that match the input query.

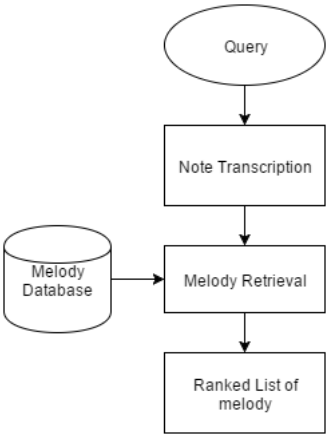

Figure 2: Overview of the QBH system

4.1. Note Transcription

4.1.1. Feature Extraction

As mentioned earlier, pitch is the most important characteristic of the melody. Currently, none of the pitch extrac-

tion algorithms is completely accurate. Therefore, we decided to use three of the best pitch extraction algorithms according to Molina et al. (2014) as features to improve our systems accuracy. Those features include the pitchyinfft (Brossier, 2006), melodia (Salamon et al., 2014) and pyin (Mauch and Dixon, 2014) algorithms.

4.1.2. Acoustic Modelling

For this task, we propose to train notes in the range of 35-85 since this range generally covers all the notes used for human humming. We use 3-state HMM monophone models to train each of the notes and a single-state HMM to train the silence model.

The extracted features are then used to train the models by using the greedy layer-wise supervised training (Dahl et al., 2012) method, which takes the extracted features as input and uses three hidden layers for training, which was found to be the optimal number of layers for this task. The DNN is trained using the Kaldi toolkit [3].

An overview of the acoustic model is shown in Figure 3.

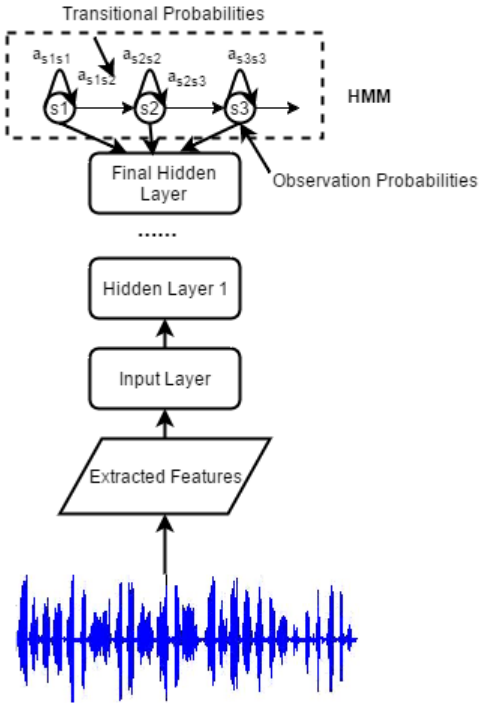

Figure 3: Overview of the acoustic modelling of the note transcription system

4.1.3. Musicological Modelling

The musicological model controls transitions among the note models and the rest in a manner similar to the language model used in speech recognition.

The transition probabilities among note HMMs are defined by note bi-grams, which were estimated from a large database of MIDI files containing melodies similar to Ryynänen and Klapuri (2004). Since key is important in determining note transitions as some note sequences are more common than others in a certain musical key, the model first estimates the key of the musical piece. Then different note bigrams are defined for each key. Therefore, given the previous note i and the estimated key k, the note bigram probability $P(n = j|n = i, k)$ gives the probability of moving from note i to note j.

[3] http://kaldi.sourceforge.net/

4.2. Candidate Melody Retrieval

The final step is to retrieve the candidate melody represented by the hummed query. For this purpose, the melody contour of the query is matched against those of all the songs in the database. The melodies ranking the highest similarity scores are presented as the candidate melodies.

The retrieval method used is similar to Yang et al. (2010). It mainly uses note-based linear scaling (NLS) and note-based recursive alignment (NRA). It uses the pitch and time information of the note and recursive-alignment combined with linear scaling to match the query with the melody. However, instead of using absolute pitch values like in Yang et al. (2010), we use the note transition values to match the similarities.

The note based linear scaling algorithm basically uses different scale factors to stretch and contract the humming query input. The distance between the humming and the song is calculated by adding those between all the intervals. The smallest distance is then used. The basic principle behind the note based linear scaling method is shown using Figures 4 and 5. The same humming query is used in both figures with different scaling and the main melody. As it can be seen from the figures, the scaling of the query has a huge impact when we calculate its distance from the main melody.

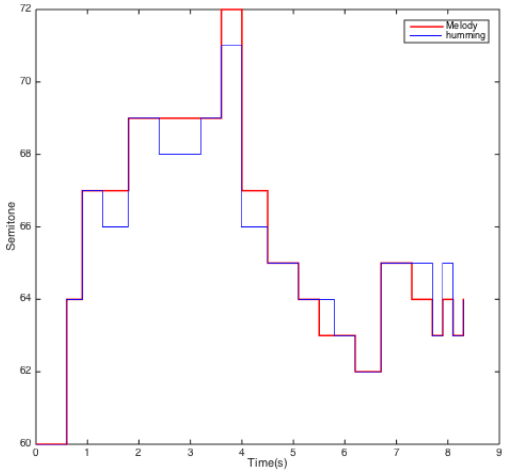

Figure 4: Principle behind NLS

The note based linear scaling distance calculates the global distance between humming and the main song. Note based recursive alignment is used for the local alignment. Linear scaling using a single value is generally not so effective because the duration of the note segments often varies greatly. Therefore, the humming query input is generally divided into several segments and linear scaling is used on each of the segment to get the optimal distance between the query

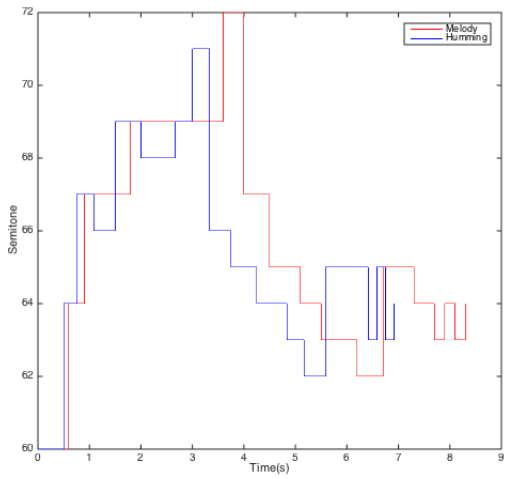

Figure 5: Principle behind NLS

and the melody. Figure 6 shows the general principles of NRA.

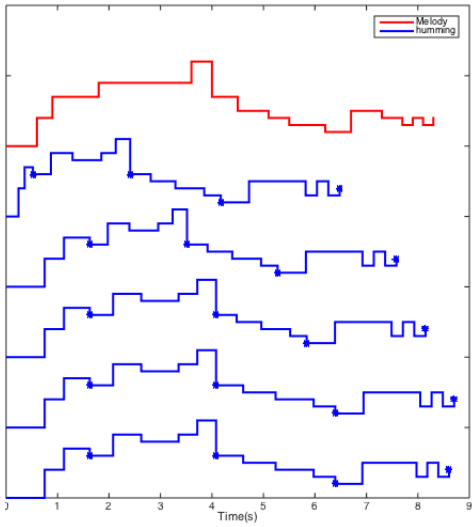

Figure 6: The procedure of NRA

The candidate songs are then ranked by their smallest distance with the query, with the song with the lowest distance ranked first. The retrieval method is used to generate a ranked list of top 20 songs most similar to the query.

5. Similar Artist Recommendation

We test the similar artist recommendation system on Bollywood artists mostly because Bollywood music provides us with a comparatively small set, which is easier to annotate and evaluate.

5.1. Dataset Building

Bollywood industry is a relatively small circle with a total number of 116 artists. There are three main websites that introduce and discuss Bollywood artists in both English and Hindi. They are NDTV[4], The Indian Express[5] and Wikipedia[6]. We have downloaded 3431 articles, 2622 from Indian Express and 809 from NDTV and Wikipedia. The articles are very comprehensive in the scope of the news they covered, including artists influences, collaborative efforts, gossip news, etc.

In order to evaluate our results, we need to manually build a standard related artist set for each artist. Regretfully, there is no acceptable gold standard online for Bollywood artists and the information available on AMG is very limited in recommending similar artists for lots of singers. We selected three Indian students with a strong Indian musical background as the annotators, and provided them with the full artists name list. They independently chose the similar artists for each target artist. We asked them to choose around 10 related artists for each candidate and pick the ones they all agree with in order to show a fair comparison to the baseline, Last.fm, which shows around 10 similar artists for each target artist. Last.fm[7], a popular internet radio, and online music service, uses classification of metadata tags to find similar artists. Since its data is relatively open, it is one of the standards that music information retrieval work compares their results to.

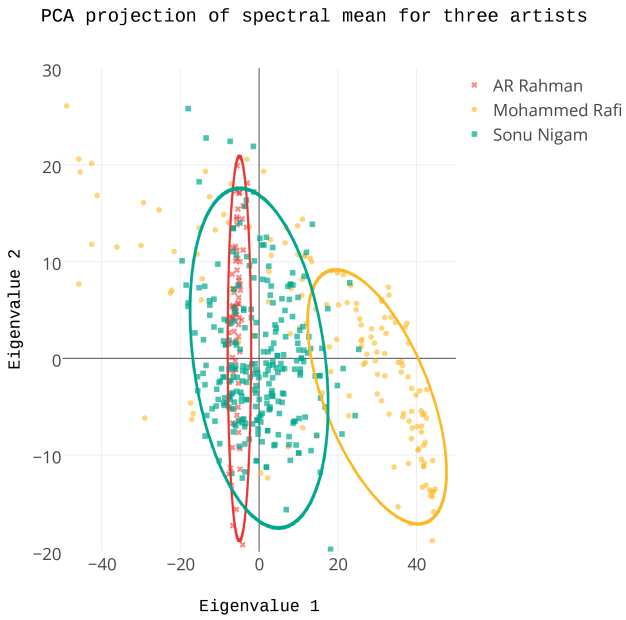

Figure 7: Spectral Mean Features with Principle Component Analysis

5.2. Spectral Mean Distance of Audio Features

Bollywood music is influenced by both classical Indian music and modern western music. A particular Bollywood artist is a composer or singer with his/her own style. For example, Rahat Fateh Ali Khan is known to fuse devotional Muslim Sufi music with other styles. We propose that musical characteristics of an artist can be represented as the

[4]http://www.ndtv.com
[5]http://www.indianexpress.com
[6]http://www.wikipedia.org
[7]http://www.last.fm

aggregated average acoustic features of all his/her songs. We can then measure the similarity between two artists using spectral distance measurements.

We extracted musical (Tzanetakis and Cook, 2000), psychoacoustic (Cabrera, 1999) and speech features (Eyben et al., 2010) from Bollywood songs for each artist in a spectral vector representation. Su et al. (2013) used these features successfully in categorizing musical genres and moods.

The musical features include timbre, chroma, spectral flatness; psychological features include loudness, sharpness; sound features include frequency and speech characteristics. For each artist s_i, the feature dimension is 865. Each entry $v_i(k)$ is the mean value of the corresponding feature for artist s_i. The distance of two artists over the acoustic feature space is calculated as follows:

$$d(i,j) = \|\mathbf{v}_i - \mathbf{v}_j\| \tag{1}$$

where i, j stand for two artists, $d(i,j)$ is the distance of artist i and j over acoustic feature space, $\|\cdot\|$ is the L2-norm, $\mathbf{v}_i$ is the audio feature vector of artist i, dimension is 865.

Note that each feature in the acoustic space has been normalized by its mean and variance. Thus the closer the distance is, the more similar the styles of the songs of two artists are.

5.3. Co-occurrence in the texts

We extract the co-occurrence of two artists in the contexts. For two arbitrary artists, s_i and s_j, the co-occurrence is computed as follows:

$$co(i,j) = \frac{\mathbf{c}_i^T \mathbf{c}_j}{|\mathbf{c}_i||\mathbf{c}_j|} \tag{2}$$

where $co(i,j)$ is the co-occurrence score of artist i and j. $\mathbf{c}_i$ is the number of times artist i occurs in each window, $|\cdot|$ is the L1-norm. For simplicity, we set our window size to the length of each paragraph.

5.4. Degree of related artists

name	degree	rank	Listeners
Lata Mangeshkar	320	1	91.4k
A R Rahman	233	2	328.3k
Kishore Kumar	199	3	53.1k
...			
Vijay Benedict	0	115	1.25k
Vijay Yesudas	0	116	2.06k

Table 1: The rank of artists sorted by their degree. Listeners are the data from Last.fm

The co-occurrence score measures the closeness of two artists in the text. In this section, we propose a new feature called, degree. In Graph theory, the degree means the number of edges that are incident on the vertex. Analogous to this definition, we define the degree of an artist as the number of times that the other artists are "incident on" the artist. Given any artist as the vertex, we calculate all the times that the other artists co-occur in the same paragraph

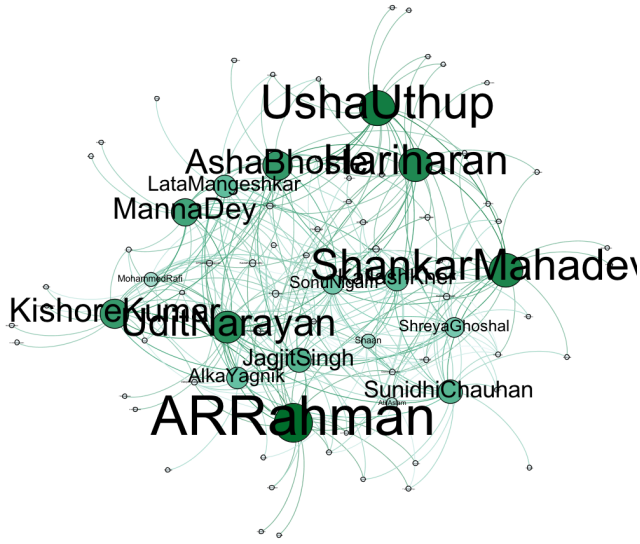

Figure 8: Degree: Link among Bollywood artists

when moving the window throughout the whole articles. The degree for the artist is calculated as follows:

$$r_i = \sum_{j \neq i;k} (c_k(i,j) > 0 \; ? \; 1 : 0) \tag{3}$$

where $c_k(i,j)$ indicate artist i and j co-occur in window k. In our experiment, artists with the highest degree and lowest degree can be viewed from Table 1. We have cross-referenced the results with Last.fm play counts, which are shown in Table 1 Listeners column, for the artists and we have found that artists with the higher degree have larger play counts, which means they are more popular. Generally speaking, the artist with a higher degree tends to be more influential. This feature helps us to re-rank the candidate list and balance the results with more popular and lesser-known artists.

5.5. Learning Feature Weights

We now have three categories of features for our training system. They are co-occurrence of this artist with the target artist, degree of influence of this artist, and the spectral mean distance of the artist with target artist. We construct our training data set as follows. For the total number of 116 artists, we construct tuple sets. For artist s_i, we have $\hat{C}_i$ and C_i, where $\hat{C}_i$ means calculated candidate artists set that is related to s_i and C_i means standard candidate artists set that is related to s_i. Thus, for each $s_j \in \hat{C}_i$, if $s_j \notin C_i$, we can build a tuple with the flag "false". Otherwise, the flag is set to "true". Each tuple contains four elements, that include three features and one flag. Once we get all the tuples, we split it into training set and testing set with the ratio 9:1. We use 10-fold validation for average performance. Since it is an unbalanced tuple set with negative tuples being dominant, we select negative tuples uniformly at random to the

	candidate set	Last.fm			spectral			Co-occurrence			Spec+Co			Spec+Co+Degree		
Degree	aritist name	P	R	**F**	P	R	**F**	P	R	**F**	P	R	**F1**	P	R	**F**
High	Lata Mangeshkar	0.90	0.38	0.53	0.72	0.33	0.46	1	0.08	0.15	0.73	0.33	0.46	0.72	0.33	0.46
	A R Rahman	0.60	0.33	0.43	0.82	1.00	0.90	0.13	0.11	0.12	0.45	0.56	0.50	0.64	0.78	0.70
	Kishore Kumar	0.44	0.33	0.38	0.55	0.50	0.52	1.00	0.33	0.50	0.64	0.58	0.61	0.55	0.50	0.52
	Sonu Nigam	0.25	0.20	0.22	0.27	0.30	0.29	0.20	0.10	0.13	0.18	0.20	0.19	0.36	0.40	0.38
	...	...														
Low	Shubha Mudgal	0.44	0.44	0.44	0.73	0.89	0.80	0	0	0	0.73	0.89	0.80	0.73	0.89	0.80
	Shibani Kashyap	0	0	0	0.73	1.00	0.84	0	0	0	0.73	1.00	0.84	0.55	0.75	0.63
	Rajkumari	0.33	0.13	0.19	0.55	0.75	0.63	1	0.13	0.22	0.55	0.75	0.63	0.55	0.75	0.63
	average	0.25	0.14	**0.17**	0.57	0.63	**0.58**	0.31	0.06	**0.10**	0.55	0.61	**0.56**	0.53	0.60	**0.55**

Table 2: *Results (precision, recall, F-score) in percentage for comparing three features performances over the last.fm results. It is calculated by 10-fold validation. Each fold iterates for 100 times in logistic regression.*

Extracted Features	MRR
Simple autocorrelation based pitch extraction	0.487
Only "PitchYinfft" algorithm	0.69
PitchYinfft, Melodia and pYin algorithms	0.8071

Table 3: Comparison of results with pitch obtained using different pitch-extraction algorithms as features

Note Transcription Algorithm	MRR
HMM-GMM based acoustic model	0.7679
DNN-HMM based acoustic model	0.8071

Table 4: Comparison of results with different algorithms

same size of the number of positive tuples to make a balanced set. We use logistic regression to train the model and update the weights with the stochastic gradient descent.

6. Experiments and Results

6.1. Results for Humming Recognition

For training note models, we have used humming data from the IOACAS corpus [8] and TCS corpus [9] with additional humming data collected by us. We annotated the humming data manually using the Tony software [10].

For the evaluation of the overall query by humming system, we have used the standard set used by MIREX for this purpose. It uses the Roger Jang corpus [11], consisting of 4431 queries and 48 ground-truth MIDI files. So, for our experiments we first transcribe the notes in all the ground truth MIDI files. The queries are then each transcribed and passed onto our retrieval system, which generates a list of most likely candidate melodies. The system is evaluated using mean Reciprocal Ranking (MRR):

$$MRR = \frac{1}{|Q|} \sum_{(i=1)}^{|Q|} (1/rank_i) \quad (4)$$

We initially used different feature sets and the MRR obtained using the different features is shown in Table 3. It indicates that using a combination of pitch values as features give better results. We also first created a simple HMM-GMM based model and the results in Table 4 indicate that using Deep Neural Networks (DNN) with HMM improves the overall retrieval rate of the system. We have currently trained the transcription system on a relatively small dataset, and we believe that using additional training data can improve our overall transcription accuracy and the retrieval accuracy.

6.2. Results for Related Artist Recommendation

Once we have done the logistic regression, we can apply the trained weights to the test set. We evaluate each artist s_i and calculate the related artists set $\hat{C}_i$. Then we compare it to the standard candidate set C_i and calculate the precision, recall and F-score.

Table 5 shows parts of our results. There are five columns. The first column is the baseline of Last.fm's results. The other four are the results from combination of features. Section 5.4. shows that artists with higher degree contain more correlation links to other artists, which partly reflect their influences. We show artists with the highest degree and lowest degree. From Table 5, we can see that Last.fm does not work very well for artists with low degree. Actually, we can not find similar artists information for artists with low degree on the last.fm's website. On the contrary, our method compensates this shortage. We can see that our method performs smoothly when dealing with both high degree and low degree artists and performs 40% better on average in F-measure. Actually, the sole spectral mean distance feature has already reached a pretty good precision and recall for some artists. Combined with co-occurrence features, we can see that precision and recall increased for the high degree artists whereas they did not decrease on the low degree artists. However, the co-occurrence feature alone does not perform so well. This may be due to the fact that our corpus is not large enough, so we will continue to collect data in order to improve our results in the future.

7. Conclusion

In this paper, we present a music retrieval and recommendation system using machine learning techniques. We propose a Deep Neural Network (DNN) based note transcription method and create a complete query by humming music retrieval system, which we test using the standard MIREX Query by humming data set. We show that the

[8] http://www.music-ir.org/mirex/wiki/

[9] http://www.ailab.hcmus.edu.vn/slp/download/TCSCorpus/

[10] https://code.soundsoftware.ac.uk/projects/tony

[11] http://www.music-ir.org/mirex/wiki/

	candidate set	Last.fm			spectral			Co-occurrence			Spec+Co			Spec+Co+Degree		
Degree	aritist name	P	R	**F**	P	R	**F**	P	R	**F**	P	R	**F1**	P	R	**F**
High	Lata Mangeshkar	0.90	0.38	0.53	0.72	0.33	0.46	1	0.08	0.15	0.73	0.33	0.46	0.72	0.33	0.46
	A R Rahman	0.60	0.33	0.43	0.82	1.00	0.90	0.13	0.11	0.12	0.45	0.56	0.50	0.64	0.78	0.70
	Kishore Kumar	0.44	0.33	0.38	0.55	0.50	0.52	1.00	0.33	0.50	0.64	0.58	0.61	0.55	0.50	0.52
	Sonu Nigam	0.25	0.20	0.22	0.27	0.30	0.29	0.20	0.10	0.13	0.18	0.20	0.19	0.36	0.40	0.38
	...	...														
Low	Shubha Mudgal	0.44	0.44	0.44	0.73	0.89	0.80	0	0	0	0.73	0.89	0.80	0.73	0.89	0.80
	Shibani Kashyap	0	0	0	0.73	1.00	0.84	0	0	0	0.73	1.00	0.84	0.55	0.75	0.63
	Rajkumari	0.33	0.13	0.19	0.55	0.75	0.63	1	0.13	0.22	0.55	0.75	0.63	0.55	0.75	0.63
	average	0.25	0.14	**0.17**	0.57	0.63	**0.58**	0.31	0.06	**0.10**	0.55	0.61	**0.56**	0.53	0.60	**0.55**

Table 5: *Results (precision, recall, F-score) in percentage for comparing three features performances over the last.fm results. It is calculated by 10-fold validation. Each fold iterates for 100 times in logistic regression.*

QBH system overall shows encouraging results and can be improved with additional data. We also propose a similar artist recommendation system and experiment the system on an exhaustive list of 116 Bollywood artists and show that the recommendations based on spectral distance, co-occurrence and degree measures give better results on average for all artists compared to popular similar artist recommendation website. We plan to collect more data in the future and test the system on a larger dataset.

8. Acknowledgements

This research was partially supported by Grant Number 16214415 of the Hong Kong Research Grant Council and partially supported by Bai Xian Asian Institute. We would like to thank our colleagues from Human Language Technology Center of The Hong Kong University of Science and Technology, who provided expertise that greatly assisted the research. We thank Anik Dey for assistance with his Bollywood knowledge which improved the precision of our work and Ricky Chan for assistance with the acoustic modeling of note transcription system. We would also like to thank the three annotators who helped us label our dataset.

9. Bibliographical References

Brossier, P. M. (2006). *Automatic annotation of musical audio for interactive applications*. Ph.D. thesis, Queen Mary, University of London.

Cabrera, D. (1999). Psysound: A computer program for psychoacoustical analysis. In *Proceedings of the Australian Acoustical Society Conference*, volume 24, pages 47–54.

Cano, P., Batlle, E., Kalker, T., and Haitsma, J. (2005). A review of audio fingerprinting. *Journal of VLSI signal processing systems for signal, image and video technology*, 41(3):271–284.

Cohen, W. W. and Fan, W. (2000). Web-collaborative filtering: Recommending music by crawling the web. *Computer Networks*, 33(1):685–698.

Dahl, G. E., Yu, D., Deng, L., and Acero, A. (2012). Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):30–42.

Dannenberg, R. B., Birmingham, W. P., Pardo, B., Hu, N., Meek, C., and Tzanetakis, G. (2007). A comparative evaluation of search techniques for query-by-humming using the musart testbed. *Journal of the American Society for Information Science and Technology*, 58(5):687–701.

Eyben, F., Wllmer, M., and Schuller, B. (2010). Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the international conference on Multimedia*, pages 1459–1462. ACM.

Ghias, A., Logan, J., Chamberlin, D., and Smith, B. C. (1995). Query by humming: musical information retrieval in an audio database. In *Proceedings of the third ACM international conference on Multimedia*, pages 231–236. ACM.

Guthrie, J. A., Guthrie, L., Wilks, Y., and Aidinejad, H. (1991). Subject-dependent co-occurrence and word sense disambiguation. In *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pages 146–152. Association for Computational Linguistics.

Hong, J., Deng, H., and Yan, Q. (2008). Tag-based artist similarity and genre classification. In *Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008. IEEE International Symposium on*, pages 628–631. IEEE.

Kharat, V., Thakare, K., and Sadafale, K. (2015). A survey on query by singing/humming. *International Journal of Computer Applications*, 111(14).

Mauch, M. and Dixon, S. (2014). pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 659–663. IEEE.

McDermott, J. H. and Oxenham, A. J. (2008). Music perception, pitch, and the auditory system. *Current opinion in neurobiology*, 18(4):452–463.

Mermelstein, P. (1976). Distance measures for speech recognition, psychological and instrumental. *Pattern recognition and artificial intelligence*, 116:374–388.

Molina, E., Tardón, L. J., Barbancho, I., and Barbancho, A. M. (2014). The importance of f0 tracking in query-by-singing-humming.

Pachet, F., Westermann, G., and Laigre, D. (2001). Musical data mining for electronic music distribution. In *Web Delivering of Music, 2001. Proceedings. First International Conference on*, pages 101–106. IEEE.

Ryynänen, M. P. and Klapuri, A. P. (2004). Modelling of note events for singing transcription. In *ISCA Tutorial*

and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing.

Salamon, J., Gomez, E., Ellis, D. P., and Richard, G. (2014). Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *Signal Processing Magazine, IEEE*, 31(2):118–134.

Shifrin, J., Pardo, B., Meek, C., and Birmingham, W. (2002). Hmm-based musical query retrieval. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 295–300. ACM.

Shih, H.-H., Narayanan, S. S., and Kuo, C. J. (2002). An hmm-based approach to humming transcription. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, volume 1, pages 337–340. IEEE.

Shih, H.-H., Narayanan, S. S., and Kuo, C. J. (2003). A statistical multidimensional humming transcription using phone level hidden markov models for query by humming systems. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 1, pages I–61. IEEE.

Su, D., Fung, P., and Auguin, N. (2013). Multimodal music emotion classification using adaboost with decision stumps. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3447–3451. IEEE.

Tzanetakis, G. and Cook, P. (2000). Marsyas: A framework for audio analysis. *Organised sound*, 4(03):169–175.

Wang, L., Huang, S., Hu, S., Liang, J., and Xu, B. (2008). An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, pages 471–475. IEEE.

Wold, E., Blum, T., Keislar, D., and Wheaten, J. (1996). Content-based classification, search, and retrieval of audio. *MultiMedia, IEEE*, 3(3):27–36.

Yang, J., Liu, J., and Zhang, W. (2010). A fast query by humming system based on notes. In *INTERSPEECH*, pages 2898–2901.