

CLUSTERING FOR MUSIC SEARCH RESULTS

Yi-Hsuan Yang, Yu-Ching Lin, and Homer Chen

National Taiwan University

ABSTRACT

Clustering for better representation of the diversity of text or image search results has been studied extensively. In this paper, we extend this methodology to the novel domain of music search. We conduct empirical evaluation of different clustering algorithms, audio feature representations, and the incorporation of lyrics for music clustering. Our evaluation shows the fusion of audio and text features yields the best clustering accuracy.

Index Terms— Music search, clustering, lyrics

1. INTRODUCTION

Because of the ambiguity of queries, the result of a text-based search system can be very diverse. For example, the search result of the query “tiger” may contain an animal of the Felidae family, a MAC OS, a baseball team, and a golf player. It would be time consuming for the user to scan through the list of retrieved items to find the desired ones. Representing the diversity of search result is vital for the success of a search system [1]. A typical solution studied extensively in the context of text search and image search is to apply a clustering algorithm to group the retrieved items [2]–[4]. For example, in [3], the retrieved images of a query are organized in clusters with semantic labels.

Likewise, the result of a text-based music search system (e.g. [5]) should also be clustered for better representation of the diversity. For example, the retrieved songs of the query “breakup” may contain sad, bittersweet, or even angry songs. Being vastly different in semantics and the audio content, these songs should be separated and assigned to different clusters to facilitate our interpretation of the search result. Clustering for music search result, however, has not been explored much in prior work.

In this paper, we present an empirical evaluation of clustering for music search result. Firstly, with features extracted from audio signals, we compare the performance of different clustering algorithms in terms of a clustering accuracy measure. Secondly, to account for the overall characteristics of a song, we evaluate the performance when audio similarity is measured based on the distribution of audio features within a song. Finally, as lyrics also carry rich semantic information of a song, we investigate the incorporation of lyrics for music clustering.

The paper is organized as follows. Section 2 reviews related work on text-based music search and clustering. Section 3 describes the system framework employed in this study. Section 4 reports the result of the empirical evaluation, and Section 5 concludes the paper.

2. RELATED WORK

In the face of the ever increasing amount of digital music, searching music through natural language queries emerges as a promising means for effective information access. In [5], a music search engine for arbitrary natural language queries is built. By mining relevant web pages from the Internet, the feature vector for a song or a query is constructed. By computing the Euclidean distance between the two associated feature vectors, the similarity between a query and a song can be measured, and the nearest songs in the database be retrieved. In [6], the authors propose to model each song by a semantic multinomial distribution estimated by a total of 135 musically-relevant concept detectors. In retrieval, the similarity between a query multinomial and a song is measured by Kullback-Leibler divergence. Content-based music retrieval from text queries has also been studied in [7]. However, little work if any has been done on clustering the music search result.

In the field of music signal processing, clustering has been applied to identify the musical elements of a song (e.g., in structural segmentation [8] and audio keyword discovery [9]). Clustering a collection of songs into different groups, nevertheless, has received less attention due to the difficulty of obtaining ground truth for evaluation [10]. In [11], the authors use web resources to evaluate an algorithm for artist clustering. Our work, on the other hand, aims at song clustering for better representation of the search result.

3. SYSTEM OVERVIEW

In this section we describe the system framework and the database employed in this study. As Figure 1 shows, we design a simple query-by-lyrics search system which returns songs that contain the query term in their lyrics. This search system matches our usual needs of finding songs related to a specific lyrics fragment, such as “happy birthday,” “new love,” and “breakup.” A clustering algorithm is then applied to the retrieved songs to identify potential clusters based on audio or text cues. We focus on the clustering part here.

The clustering problem can be formulated as follows. Given a query $q^{(i)}$ and a set of songs $D^{(i)}$ retrieved by a text-based search engine, a clustering algorithm is employed to divide $D^{(i)}$ into K clusters $X_1^{(i)}, X_2^{(i)}, \dots, X_K^{(i)}$. To evaluate the clustering result quantitatively, we ask subjects to divide $D^{(i)}$ into K clusters through a subjective test. This results in K sets of songs $Y_1^{(i)}, Y_2^{(i)}, \dots, Y_K^{(i)}$ and a “don’t care” set of songs whose cluster assignments do not reach a consensus among subjects. We can then measure the accuracy of a clustering algorithm as follows,

$$\max_{\pi, \pi'} \frac{\sum_{k=1}^K |X_{\pi_k}^{(i)} \cap Y_{\pi'_k}^{(i)}|}{\sum_{k=1}^K |Y_{\pi'_k}^{(i)}|}, \quad (1)$$

where π is a permutation of $\{1, 2, \dots, K\}$ and $|X|$ denotes the cardinality of a set. In other words, we look for two permutations of $X^{(i)}$ and $Y^{(i)}$ that bring about the maximal intersections between the $2K$ sets of songs. The clustering result of songs belong to the don’t care set is disregarded. The performance of a clustering algorithm is measured by the average clustering accuracy across queries $q^{(1)}, q^{(2)}, \dots, q^{(|Q|)}$, where $|Q|$ denotes the number of queries.

Our music database consists of 1240 Chinese pop songs obtained from personal collections. The lyrics are retrieved from the Internet by a web crawler. We define 9 queries and ask 3 subjects to group the retrieved songs of each query into $K=2$ clusters. Note we have fixed K to 2 in this study to simplify the annotation and evaluation processes. Table I shows the employed queries, the number of retrieved songs, the number of songs assigned to each cluster by the subjects, and the general properties of the songs in each cluster. The baseline method in our evaluation groups all songs to form a single cluster. According to (1), the baseline accuracy for query $q^{(i)}$ is computed as $|Y_1^{(i)}| / (|Y_1^{(i)}| + |Y_2^{(i)}|)$. The average accuracy of the baseline method is 58.8%.

4. EMPIRICAL EVALUATION

We carry out empirical evaluations to answer the following three questions regarding clustering for music search results.

1. Which clustering algorithm brings about best clustering accuracy? We focus on audio features in this evaluation.
2. Does measuring similarity based on the distribution of audio features in a song improve clustering?
3. Does the incorporation of features extracted from the lyrics improve clustering?

4.1. Evaluation of different clustering algorithms based on audio features

We use the music software Marsyas [12] to extract audio features from music signals and construct a 30-dimensional

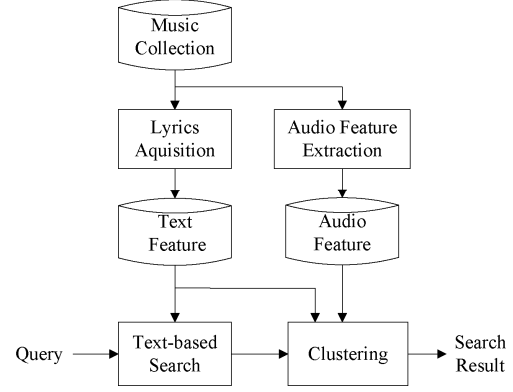


Figure 1. A schematic diagram of the lyrics-based search system employed in this study.

Table I
Queries and the clusters annotated by subjects

query $q^{(i)*}$	# retrieved songs $ D^{(i)} $	# songs in 1 st cluster	# songs in 2 nd cluster	properties of the songs in 1 st /2 nd cluster
hero	10	3	2	exciting / not exciting
surprise	10	3	3	exciting / not exciting
sunset	11	7	4	sad / relaxing
lie	11	5	4	angry / sad
forgive	11	6	4	sad / not sad
breakup	12	5	5	sad / not sad
tempted	18	8	5	exciting / not exciting
sad	19	8	6	sad / not sad
moon	22	10	4	relaxing / not relaxing

* The queries have been translated from Chinese to English

feature space. Marsyas generates 19 timbral texture features (spectral centroid, spectral rolloff, spectral flux, time domain zero-crossing and MFCC), 6 rhythmic content features (by beat and tempo detection) and 5 pitch content features (by multi-pitch detection). MFCC (Mel-frequency cepstral coefficient), the most commonly used feature representation for audio signal processing, is computed by taking the cosine transform of the short-term log power spectrum expressed on a non-linear perceptual-related mel-frequency scale. 5 MFCCs are computed for each short frame of 23 ms, and the resulting feature vectors are collapsed into a single vector by taking mean and standard deviation. Note that the feature dimension is kept small to avoid the curse-of-dimensionality problem that plagues the measurement of similarity in high dimensions [13]. After z-score normalization, the dissimilarity between two songs can be measured by the Euclidean distance.

We compare the performance of the following three clustering algorithms: kmeans clustering [14], spectral clustering [15], and affinity propagation [16]. Being one of the most classical methods, kmeans clustering partitions the data points by assigning each point to the nearest cluster centroids. It iteratively determines the cluster centroids and which cluster an item belongs to by the EM (expectation-

maximization) algorithm. As suggested in [9], we select the initial cluster centroids by making them as orthogonal to each other as possible. A well known problem of kmeans clustering is its simplifying assumption that the density of each cluster is Gaussian-like. To mitigate this problem, spectral clustering [15] performs clustering on the top K eigenvectors of an affinity matrix derives from the distance between items, rather than on the original feature space. The affinity matrix F is defined by $F_{ij}=\exp(-d(u_i, u_j)^2/2\sigma^2)$ when $i\neq j$, and otherwise $F_{ii}=1$. Here $d(u_i, u_j)=\|u_i-u_j\|$ is the Euclidean distance between the feature vectors u_i and u_j , and σ is the scaling factor set by the average Euclidean distance in the data [9]. Different from the above methods, the recently proposed method affinity propagation [16] is free from the need to specify the number of clusters. It has been successfully applied to cluster images of faces, detect genes, and identify representative sentences.

In rows 1–4 of Table II, we see that the average clustering accuracy is improved to 80–85% by the above algorithms. Spectral clustering, among the three, achieves the best accuracy of 85.4%. As shown in Figure 2, a consistent improvement of clustering accuracy for each query is observed. Further test (not reported here due to space limitation) using different audio features shows that spectral clustering consistently outperforms the other two methods, and shows that the use of Marsyas features and spectral clustering gives rise to best performance.

4.2. Evaluation of modeling the distribution of features

We then evaluate the clustering result by modeling the distribution of the MFCCs over all the frames of a song using a Gaussian mixture model, instead of collapsing them into a single vector. A Gaussian mixture model (GMM) estimates a probability density as the weighted sum of M Gaussian densities. The parameters of the GMM are estimated by the EM algorithm. We fit a GMM of 3 Gaussian densities ($M=3$) for the MFCCs of a song, and adopt Monte Carlo sampling [13] to compute the distance between two GMMs as follows. We sample a large number N_s of points S from a model A , and compute the likelihood of these points given the other model B . That is,

$$d(A, B) = \sum_{i=1}^{N_s} \log P(S_i^A | A) + \sum_{i=1}^{N_s} \log P(S_i^B | B) - \sum_{i=1}^{N_s} \log P(S_i^A | B) - \sum_{i=1}^{N_s} \log P(S_i^B | A) \quad (2)$$

This distance measure is used in the construction of affinity matrix for spectral clustering. Interestingly, we find empirically that normalizing the value of (2) to $[0, 1]$ by a sigmoid function slightly improves the clustering accuracy.

Since GMM disregards the temporal order of MFCCs, we further employ Hidden Markov Model (HMM) to

Table II
Results of the empirical evaluation

	clustering algorithm	feature (# feature)	accuracy
1	baseline	–	58.8%
2	kmeans clustering	Marsyas (30)	83.2%
3	spectral clustering	Marsyas (30)	85.4%
4	affinity propagation	Marsyas (30)	82.1%
5	spectral clustering	MFCC (10)	73.4%
6	spectral clustering	MFCC + GMM	81.9%
7	spectral clustering	MFCC + HMM	75.9%
8	spectral clustering	tfidf (5000)	67.0%
9	spectral clustering	plsa (40)	77.0%
10	isoperimetric co-clustering	Marsyas + plsa (70)	86.9%

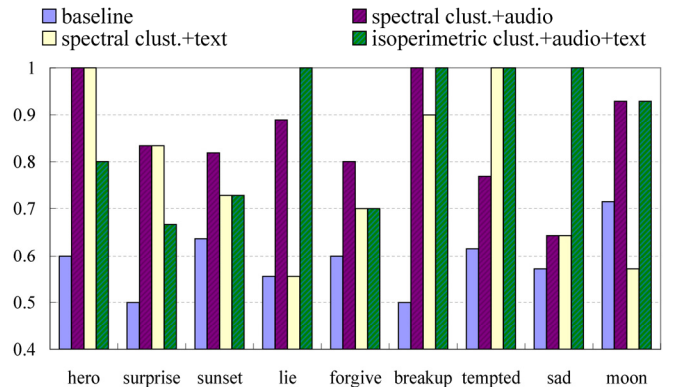


Figure 2. Clustering accuracy of the baseline method and various clustering algorithms. Isoperimetric clustering, which utilizes both audio and text features, achieves the best average accuracy of 87%.

incorporate temporal context. A HMM is a set of hidden states which are linked with transition probabilities from one state to another [17]. Modeling each state with 3 Gaussian densities, we fit each song with a 4 state HMM using the Baum-Welsh algorithm [18]. To compute the distance between two HMMs, we sample N_s sequences of length N_F from a model and compute the likelihood of these sequences given the other model by the Viterbi algorithm [18]. Refer to [13] for more details.

In rows 5–7 of Table II, we see modeling the MFCCs by GMM or HMM indeed outperform the single vector method when only 10 MFCC features are used. However, since we are able to achieve a higher performance with the single vector method by incorporating more musical features (row 3), the additional complexity introduced by generative models seems not to provide substantial benefits. It is also found that HMM does not perform better than GMM. A similar observation has been made in [17] in the context of music genre classification.

4.3. Evaluation of the incorporation of lyrics

Finally, we evaluate the incorporation of lyrics for music clustering. Lyrics are normally available on the web and downloadable with a simple crawler [19]. The retrieved lyrics are preprocessed with traditional information retrieval

operations such as stopword removal, stemming, and tokenization. A standard text feature representation, called the bag-of-words model, can then be constructed by counting the occurrence of words in each lyric, weighted by a tfidf technique. However, as the total number of unique words in a text corpus can be exceedingly large (more than 5000 for our database), we apply the probabilistic latent semantic analysis (PLSA) [20] to reduce the bag-of-words model into a 40-dimensional latent vector space. As shown in rows 8 and 9 of Table II, better accuracy is achieved by PLSA.

As shown in Figure 2, the performance of text-based clustering is generally worse than the audio-based one. This is reasonable since melody dominates our perception of music, according to a psychological study [21].

We also notice that for some queries such as “tempted,” text-based clustering achieves remarkably high accuracies. Therefore, we further investigate a novel clustering algorithm, called consistent isoperimetric high-order co-clustering (CIHC) [4], to utilize both audio and text features. By formulating clustering as a graph partition problem, CIHC simultaneously integrates features from different modalities under a graph theoretical framework. As shown in Table II, CIHC achieves the highest accuracy of 86.9% among all the clustering algorithms. In addition, it is interesting to note that CIHC does not perform better than spectral clustering for queries whose number of retrieved songs is relative small (e.g., queries “hero,” “surprise,” and “sunset”). This observation may reveal that it is possible to develop a more effective clustering algorithm.

5. CONCLUSION

In this paper, we have presented an empirical evaluation of clustering the retrieved songs of a text-based music search system. Our major findings are as follows. First, spectral clustering performs better than the classical kmeans clustering and the recently-proposed affinity propagation. Second, modeling the distribution of audio features gives no salient improvement. Third, audio features generally generate better clustering results than text features extracted from the lyrics. Finally, fusing audio and text features by consistent isoperimetric high-order co-clustering achieves the highest accuracy of 86.9%. Due to the difficulty of obtaining ground truth, the scale of the dataset utilized here may be not large enough. We hope this work inspires more research works on clustering for music search results.

6. ACKNOWLEDGMENTS

This work is supported by a grant from Chunghwa Telecom under the contract RAC970362 and a grant from the National Science Council of Taiwan under the contract NSC 97-2221-E-002-111-MY3.

7. REFERENCES

- [1] C. Clark et al, “Novelty and diversity in information retrieval evaluation,” *Proc. SIGIR*, pp. 659–666, 2008.
- [2] H.-J. Zeng et al, “Learning to cluster web search results,” *Proc. SIGIR*, pp. 210–217, 2004.
- [3] F. Jing et al, “IGroup: web image search results clustering,” *Proc. ACM MM*, pp. 377–384, 2006.
- [4] M. Rege, M. Dong, and J. Hua, “Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering,” *Proc. WWW*, pp. 317–326, 2008.
- [5] P. Knees, T. Pole, M. Schedl, and G. Widmer, “A music search engine built upon audio-based and web-based similarity measures,” *Proc. SIGIR*, pp. 447–454, 2007.
- [6] D. Turnbull et al, “Towards musical query-by-semantic description using the CAL500 data set,” *Proc. SIGIR*, pp. 439–446, 2007.
- [7] C. Chechik et al, “Large-scale content-based audio retrieval from text queries,” *Proc. ACM MIR*, pp. 105–112, 2008.
- [8] M. Levy and M. Sandler, “Structural segmentation of musical audio by constrained clustering,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, 2008.
- [9] L. Lu and A. Hanjalic, “Audio keyword discovery for text-like audio content analysis and retrieval,” *IEEE Trans. Multimedia*, vol. 10, no. 1, pp. 74–85, 2008.
- [10] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, “A large-scale evaluation of acoustic and subjective music similarity measures,” *Proc. ISMIR*, pp. 99–105, 2003.
- [11] W. Peng, T. Li, and M. Ogihara, “Music clustering with constraints,” *Proc. ISMIR*, pp. 27–32, 2007.
- [12] G. Tzanetakis and P. Cook, “Marsyas: a framework for audio analysis,” *Organized Sound*, vol. 4, no. 3, pp. 169–175, 2000. [online] <http://marsyas.sness.net/>
- [13] J.-J. Aucouturier and F. Pachet, “Improving timbre similarity: how high’s the sky,” *Journal of Negative Results in Speech and Audio Sciences*, vol. 1, no. 1, 2004.
- [14] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Pearson Education, 2007.
- [15] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” *Proc. NIPS*, pp. 849–856, 2001.
- [16] B. J. Frey and D. Dueck, “Clustering by passing messages between data points,” *Science*, vol. 315, pp. 972–977, 2007.
- [17] A. Flexer, E. Pampalk, and G. Widmer, “Hidden Markov Models for spectral similarity of songs,” *Proc. DAFx*, 2005.
- [18] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, 1999.
- [19] G. Geleijnse and J. Korst, “Efficient lyrics extraction from the web,” *Proc. ISMIR*, 2006.
- [20] T. Hofmann, “Probabilistic latent semantic analysis,” *Proc. UAI*, pp. 289–296, 1999.
- [21] S. O. Ali and Z. F. Peynircioglu, “Songs and emotions: are lyrics and melodies equal partners,” *Psychology of Music*, vol. 34, no. 4, pp. 511–534, 2006.