QUERY BY MULTI-TAGS WITH MULTI-LEVEL PREFERENCES FOR CONTENT-BASED MUSIC RETRIEVAL

Ju-Chiang Wang[†*], *Meng-Sung Wu*[†], *Hsin-Min Wang*[†] *and Shyh-Kang Jeng*[*]

[†]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[*]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
E-mail: {asriver, wums, whm}@iis.sinica.edu.tw, skjeng@cc.ee.ntu.edu.tw

ABSTRACT

This paper presents a novel content-based music retrieval system that accepts a query containing multiple tags with multiple levels of preference (denoted as an MTML query) to retrieve music from an untagged music database. We select a limited number of popular music tags to form the tag space and design an interface for users to input queries by operating the scroll bars. To effect MTML content-based music retrieval, we introduce a tag-based music aspect model that jointly models the auditory features and tag-based text features of a song. Two indexing methods and their corresponding matching methods, namely pseudo song-based matching and tag co-occurrence pattern-based matching, are incorporated into the pre-learned tag-based music aspect model. Finally, we evaluate the proposed system on the MajorMiner dataset. The results demonstrate the potential of using MTML queries to retrieve music from an untagged music database.

Index Terms— Music retrieval system, tag-based music aspect model, query by multi-tags, social tags.

1. INTRODUCTION

With the explosive growth of music audio collections, methods for retrieving music tracks have generated a great deal of interest among researchers in recent years. There are many ways to retrieve pieces of desired music, e.g., query by humming/singing, query by example, query by meta-information, and query by music tag. In this paper, we study the query by music tag task.

Music tags are generated from a folksonomy, which is a full-scale taxonomy of music that reflects the current usage among Internet users [1]. For example, Last.fm is a collaborative social tagging network that collects information about music habits of users in terms of music tags. In addition, several web-based music tagging games have been created with the objective of collecting useful tags, e.g., MajorMiner.org [2], Tag A Tune [3], and the Listen Game developed by D. Turnbull [4]. These applications provide tagged music resources for researchers to investigate users' tagging behavior among different kinds of music tracks [5]; however, the collected music tags have only been assigned to existing music tracks, i.e., there are no tags available for new tracks. The so-called cold start issue has motivated research into a number of interesting topics, such as automatic music tag annotation and tag-based music retrieval from an untagged music database.

Music tags provide a natural way to describe general musical concepts because people tend to mentally tag a piece of music with specific words when they listen to it. The tags can include different types of musical information, such as genre, mood, instrumentation, personal preferences, original artist(s), and particular usages. A user can assign tags of same type or different types to a specific song, and this may lead to some specific tag co-occurrence patterns (denoted as *co-tag* patterns hereafter) among auditorily similar songs. For example, some instrumental or timbre tags, such as guitar, drum, rap, saxophone, piano, synth, and drum-machine, are inspired by auditory cues directly. These instrumental tags usually result in a series of consequent tags, e.g., electric guitar, distortion, and drum commonly result in rock, loud, and metal tags; saxophone and piano are often assigned together with jazz or soft tags; rap mostly co-occurs with the hip-hop tag; while synth and drum-machine are often with electronic and techno tags. Therefore, retrieving music with a certain co-tag pattern is more effective than retrieving music with a single tag. For example, if a person tags the song "Trouble" performed by "Coldplay" with "male, pop, piano, slow, soft, and British" tags, he may use those tags to search for other songs that are similar to "Trouble" later. A single tag query like "pop" is very ambiguous, but combining a number of tags provides a clearer description of the desired song.

The co-tag phenomenon motivated us to investigate retrieving songs from a music database by using multiple tags. In this paper, we propose a content-based music retrieval system that accepts a query comprised of multiple tags with multiple levels of preference (denoted as an MTML query

This work was supported in part by Taiwan e-Learning and Digital Archives Program (TELDAP) sponsored by the National Science Council of Taiwan under Grant: NSC99-2631-H-001-020.

hereafter), derived from a predefined music tag set, to search for music in an untagged music database. As shown in Fig. 1, users can input queries by simply operating the scroll bars designed according to a pre-defined tag set. It is believed that a tag will enter the common musical vocabulary in an organic fashion once it is adopted by a large number of users. Therefore, it is reasonable to select a limited number of the most popular music tags based on their frequencies to form the semantic space of possible tag queries for use on the MTML interface.

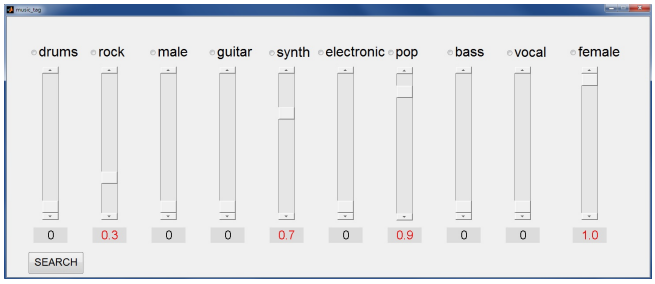

Fig. 1. The query interface of the proposed MTML content-based music retrieval system.

A number of approaches have been proposed for music retrieval using music tags. Turnbull et al. [6] model the feature distribution of each tag with a Gaussian mixture model (GMM) and estimate the model's parameters with a weighted mixture hierarchies expectation maximization algorithm. Lo et al. [7] apply a binary ensemble classifier comprised of SVM and AdaBoost to each tag to model its corresponding music features. The above systems can only retrieve music with a single tag due to the use of independent tag classifiers. In addition, Levy et al. [1] apply text-based information retrieval techniques to music collections. They represent a music track with a joint vocabulary made up of social tags and muswords, which are quantized terms that represent the auditory characteristics of a segment-based signal in a track. The authors utilize two IR models, the vector space model and the aspect model, to retrieve music in a *query-by-example* fashion. Each track in the music database is represented as a scaled concatenation of a bag-of-tags (BOW) vector and a bag-of-muswords (BOM) vector, denoted as BOW+M. Two aspect models are learned: the tag-based aspect model is learned for indexing the BOW feature of a track, and the musword-based aspect model is learned for indexing the BOM feature of a track. The BOW+M vector of a test audio query is compared to the BOW+M vectors of tracks in the music database. Because of tag sparsity, only a small number of tag labels are left unmasked in the database. The BOM+W representation and the aspect models improve the retrieval performance even when tags are scarce. In [1], the music database is not completely untagged because a certain amount of label information is available in the database. Recently, the latent aspect model of music tags has been extended to handle open vocabulary tags [8]. Although the approaches proposed in [1, 8] accept free text queries, which are equivalent to multi-tag queries, they do not consider the preference of each tag.

The remainder of this paper is organized as follows. Section 2 contains an overview of the proposed MTML content-based music retrieval system. In Section 3, we describe the audio feature extraction module and song-level representation. In Section 4, we introduce the tag-based music aspect model and explain how to apply it in music tag annotation and tag-based music retrieval. We discuss the evaluation of the proposed system on the MajorMiner dataset in Section 5. In Section 6, we summarize our conclusions.

2. SYSTEM OVERVIEW

The proposed MTML content-based music retrieval system is implemented in two phases; the feature indexing phase and the music retrieval phase, as shown in Fig.2. In the feature indexing phase, each music track in the untagged music database is indexed as a fixed-dimensional vector based on the track's audio features. We use two approaches to index track-level audio features: indexing based on a music feature reference or indexing based on automatic music tagging. In the retrieval phase, given an MTML query, the music retrieval system will return a ranked list of music tracks. We also apply two matching methods that correspond to the above indexing approaches, namely, the pseudo song-based matching method and the co-tag pattern-based matching method. To achieve content-based music retrieval using MTML queries, we jointly model the co-tag pattern (i.e., the normalized MTML distribution) and the auditory features of each music track.

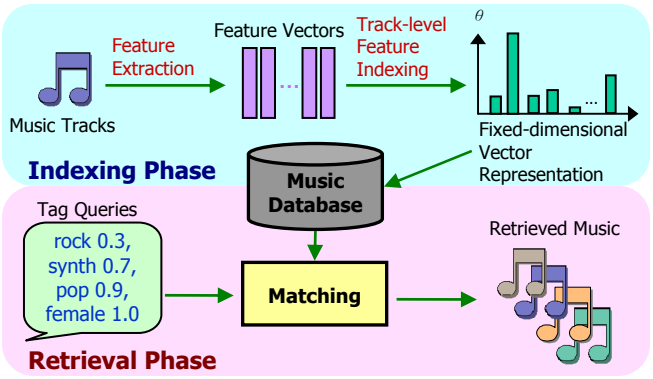

Fig. 2. Flowchart of the proposed MTML content-based music retrieval system.

2.1. The generative flow of the co-tag patterns

We assume that the human memory stores some latent co-tag patterns that are difficult to describe. When tagging a song, people usually choose one or more of those patterns according to the auditory musical characteristics of the song. Although we cannot describe the latent co-tag patterns and auditory musical characteristics exactly, we believe there is

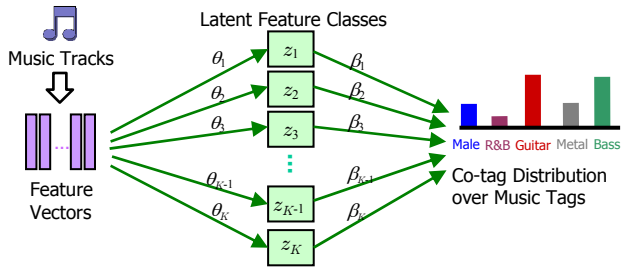

Fig. 3. The overall co-tag generative flow.

a strong link between them. Therefore, as shown in Fig. 3, we introduce a hidden layer of latent classes of music features (denoted as the latent feature class hereafter) into the co-tag generative flow to link the latent co-tag patterns and music features.

Assume there are K latent feature classes z_k, k=1,…,K. Each class z_k represents a group of music feature vectors, and its corresponding latent co-tag pattern is denoted as β_k. A song is first decomposed into a sequence of feature vectors. Then, the effective number (denoted as θ_k) of a latent feature class z_k of the song can be computed according to a pre-trained feature reference. Theoretically, all observed co-tag patterns can be generated by the convex combination of β_k and θ_k, k=1,…,K. With β_k, k=1,…,K, we can predict the co-tag pattern for an untagged song based on the value of θ_k, k=1,…,K. If the song's music features can be described completely by a certain latent feature class z_k, i.e., θ_k=1, and θ_i=0 for all $i \neq k$, then its co-tag pattern would exactly follow the latent co-tag pattern β_k. To implement the idea, we assume that the latent co-tag pattern can be modeled as a multinomial distribution, and the latent feature classes can be described by mixture components of a Gaussian mixture model (GMM). Then, all co-tag patterns can be generated by a mixture of the multinomial models, i.e., a mixture of latent co-tag patterns.

3. MUSIC FEATURE EXTRACTION AND SONG-LEVEL REPRESENTATION

In this section, we describe the music features used in this work, and explain how we convert the frame-based feature vectors of a song into a song-level fixed-dimensional representation through a music feature reference.

3.1. Music feature extraction

We use MIRToolbox 1.3 for music feature extraction [8]. As shown in Table 1, we consider four types of features, namely, dynamic, spectral, timbre, and tonal features. To ensure alignment and prevent mismatch of different features in a vector, all the features are extracted with the same fixed-sized short-time frame. Given a song, a sequence of 70-dimensional feature vectors is extracted with a 50ms frame size and 0.5 hop shift. Then, we normalize the 70-dimensional frame-based feature vectors in each dimension to mean 0 and standard deviation 1.

Table 1. The music features used in the 70-dimensional frame-based music feature vector.

Types	Feature Description	Dim
dynamic	rms	1
spectral	centroid, spread, skewness, kurtosis, entropy, flatness, rolloff 85, rolloff 95, brightness, roughness, irregularity	11
timbre	zero crossing rate, spectral flux, MFCC, delta MFCC, delta-delta MFCC	41
tonal	key clarity, key mode possibility, HCDF, chroma , chroma peak, chroma centroid	17

3.2. The universal music feature model

To produce a standard music feature reference that can be used for any music database, we introduce the *Universal Music Feature Model* (UMFM), which uses a large GMM to model a universal collection of music feature vectors. In the UMFM, we define a "latent feature class" as a latent variable z_k, k=1,…,K, which corresponds to the k-th Gaussian component with mixture weight π_k, mean vector $\boldsymbol{\mu}_k$, and covariance matrix $\boldsymbol{\Sigma}_k$. The conventional EM algorithm is used to learn the UMFM.

Given the t-th feature vector $\mathbf{x}_{nt}$ of a song s_n, the UMFM computes the posterior probability of z_k as follows:

$$p(z_k|\mathbf{x}_{nt}) = \frac{p(z_k;\pi_k)p(\mathbf{x}_{nt}|z_k;\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} p(z_l;\pi_l)p(\mathbf{x}_{nt}|z_l;\boldsymbol{\mu}_l,\boldsymbol{\Sigma}_l)}. \quad (1)$$

By assuming that each frame contributes equally to the song, we obtain the UMFM-based representation $\boldsymbol{\theta}_n$ whose k-th component θ_{nk} is computed by

$$\theta_{nk} \leftarrow p(z_k|s_n) = \frac{1}{T_n}\sum_{t=1}^{T_n} p(z_k|\mathbf{x}_{nt}), \quad (2)$$

where T_n is the number of frames in the song s_n. Although the true universal music feature set that covers all frame-based feature vectors from any music database is not available in practice, we believe the UMFM can index any songs if it is trained on a large enough feature set. The song-level feature representation of a song can be modeled by the generative process of a multinomial mixture model, which we discuss in the next section.

4. THE TAG-BASED MUSIC ASPECT MODEL

The aspect model has been used for tag-based music retrieval in previous research [1, 8, 10]. In this work, our goal is to learn the tag-based music aspects from songs' auditory features via a tag-level multinomial mixture model (MMM) on a tagged music dataset. We treat the music tag labels as the text features of a song and represent them with a "bag-

of-tags" vector. Suppose we have a music dataset with N songs, each denoted by s_n, n=1,…,N; and let each song's tag count $c(n,m)$, m=1,…,M, be an non-negative integer representing the number of times that tag w_m has been assigned to song s_n. The co-tag pattern of a song is a sequence of M tags, denoted as $\mathbf{w}=(w_1,w_2,\ldots,w_M)$; and $p(\mathbf{w} \mid s_n; \mathbf{B})$ is the tag-level MMM of song s_n, where $\mathbf{B}$ is the parameter set of the MMM.

4.1. The generative process

Generating the co-tag pattern $\mathbf{w}$ of song s_n involves three steps. First, we choose a latent feature class z_k with the probability θ_{nk}:

$$p(z_k \mid s_n; \boldsymbol{\theta}_n) = \theta_{nk}. \tag{3}$$

The probability of z_k is viewed as a mixture prior learned in the feature representation stage. The prior acts as a constraint on the music features in the Bayesian learning framework. Second, given the latent feature class z_k, a tag w_m is generated with the conditional probability

$$p(w_m \mid z_k; \boldsymbol{\beta}_k) = \beta_{km}, \tag{4}$$

where β_{km} is the m-th parameter in the k-th vector $\boldsymbol{\beta}_k$ of parameter matrix $\mathbf{B}$. Finally, the co-tag pattern $\mathbf{w}$ of the song s_n is generated by marginalization over z_k, which yields

$$p(\mathbf{w} \mid s_n; \boldsymbol{\theta}_n, \mathbf{B}) = \prod_{m=1}^{M} \sum_{k=1}^{K} p(z_k \mid s_n; \boldsymbol{\theta}_n) p(w_m \mid z_k; \boldsymbol{\beta}_k) = \prod_{m=1}^{M} \sum_{k=1}^{K} \theta_{nk} \beta_{km}^{c(n,m)}. \tag{5}$$

Given the weight matrix $\boldsymbol{\Theta}$ whose n-th vector is $\boldsymbol{\theta}_n$ of song s_n, the final log-likelihood of the training music dataset can be expressed as

$$\begin{aligned} \boldsymbol{L} &= \log p(\mathbf{w}; \boldsymbol{\Theta}, \mathbf{B}) = \sum_{n=1}^{N} \log\left[p(s_n) + p(\mathbf{w} \mid s_n; \boldsymbol{\theta}_n, \mathbf{B})\right] \\ &= \sum_{n=1}^{N} \left[\log p(s_n) + \sum_{m=1}^{M} c(n,m) \log \sum_{k=1}^{K} \theta_{nk} \beta_{km} \right], \end{aligned} \tag{6}$$

where $p(s_n)$ is assumed to be uniformly distributed.

4.2. Model inference with the EM algorithm

The MMM can be fitted with respect to $\mathbf{B}$ and $\boldsymbol{\Theta}$ with maximum-likelihood (ML) estimation. However, in the generative process, $\boldsymbol{\Theta}$ is determined in the song-level feature representation stage in Eq. (2). Therefore, we only need to estimate $\mathbf{B}$. We apply the EM algorithm to maximize the log-likelihood in Eq. (6) with respect to $\mathbf{B}$ in the presence of latent variable $\boldsymbol{z}$.

In the E-step, given a song s_n and tag w_m, the posterior probability of z_k is

$$p(z_k \mid s_n, w_m; \boldsymbol{\Theta}, \mathbf{B}) = \frac{p(z_k \mid s_n; \boldsymbol{\theta}_n) p(w_m \mid z_k; \boldsymbol{\beta}_k)}{\sum_{q=1}^{K} p(z_q \mid s_n; \boldsymbol{\theta}_n) p(w_m \mid z_q; \boldsymbol{\beta}_q)}. \tag{7}$$

In the M-step, $\mathbf{B}$ is updated based on the expected complete data log-likelihood over the posterior probabilities computed in the E-step. The update rule for β_{km} is

$$\beta_{km} \leftarrow p(w_m \mid z_k; \boldsymbol{\Theta}, \mathbf{B}) = \frac{\sum_{n=1}^{N} c(n,m) p(z_k \mid s_n, w_m; \boldsymbol{\theta}_n, \boldsymbol{\beta}_k)}{\sum_{n=1}^{N} \sum_{r=1}^{M} c(n,r) p(z_k \mid s_n, w_r; \boldsymbol{\theta}_n, \boldsymbol{\beta}_k)}. \tag{8}$$

$\boldsymbol{\beta}_k$ actually represents the latent co-tag pattern of z_k and gives a tag-based music aspect to the latent feature class. We can apply the model in tag annotation, i.e., predicting tags for a new song, or in tag-based music retrieval.

4.3. Music retrieval with MTML queries

There are two ways to apply the aspect model in tag-based music retrieval. The first predicts the co-tag distribution of each song in the music database as its index and applies the well-known vector space model to compute the cosine distance between the MTML query and each music track. To index a song s via the auto-tagger, the song is first transformed into the song-level feature representation $\boldsymbol{\theta}$. Then, the affinity of tag w_m for the song is computed by the convex combination of mixture probabilities, each with parameter $\boldsymbol{\beta}_k$:

$$p(w_m \mid s; \boldsymbol{\theta}, \mathbf{B}) = \sum_{k=1}^{K} p(z_k \mid s; \boldsymbol{\theta}) p(w_m \mid z_k; \boldsymbol{\beta}_k) = \sum_{k=1}^{K} \theta_k \beta_{km}. \tag{9}$$

The predicted co-tag affinity is then normalized to sum to unity. After indexing each song in the music database, given an MTML query, we use a standard matching function, i.e., the cosine distance, to compare the query with the predicted co-tag distribution of each song in the database. The retrieval method that applies co-tag pattern-based matching is called the Auto-tagging method.

The second way is to compute the cosine distance of the UMFM-based representations of the MTML query and each music track. Each song in the database is indexed by its UMFM-based representation derived by Eq. (2). The MTML query represented by $\tilde{c}(m)$, $m=1,2,\ldots,M$, is first normalized to sum to one, and then folded in to the tag-based music aspect model using the EM algorithm to generate its UMFM-based representation. The E-step is

$$p(z_k \mid \tilde{s}, w_m; \mathbf{B}) = \frac{p(z_k \mid \tilde{s}) p(w_m \mid z_k; \boldsymbol{\beta}_k)}{\sum_{q=1}^{K} p(z_q \mid \tilde{s}) p(w_m \mid z_q; \boldsymbol{\beta}_q)}, \tag{10}$$

where $\tilde{s}$, which can be initialized from a uniform vector, is the pseudo song corresponding to the MTML query. The update rule for $\tilde{\theta}_k$ of the pseudo song in the M-step is

$$\widetilde{\theta}_k \leftarrow p(z_k \mid \widetilde{s}; \mathbf{B}) = \frac{\sum_{m=1}^{M} \widetilde{c}(m) p(z_k \mid \widetilde{s}, w_m; \boldsymbol{\beta}_k)}{\sum_{r=1}^{M} \widetilde{c}(r)}. \quad (11)$$

Given an MTML query, we use the same standard matching function to compare the UMFM-based representation of the pseudo song predicted from the MTML query and the UMFM-based representation of each song in the database. The method that applies pseudo song-based matching is called the Fold-in method.

5. EVALUATION ON THE MAJORMINER DATASET

The proposed tag-based music retrieval system is designed for use with any music database as long as there is a pre-trained UMFM and a tagged music dataset for defining the tag space and learning the music aspect model. We evaluate the proposed system on the MajorMiner dataset [2]. The dataset is obtained from the MajorMiner website, which uses a game to gather informative free text labels for music. Each player labels randomly given music clips (each about 10 seconds long) by listening to them without any meta-information. If two players assign the same text label to a music clip, the label is adopted by the system. Hence, each music clip's tag count is at least 2. We download all the music clips associated with the most commonly used 45 tags from the MajorMiner website. The resulting dataset contains 2,472 music clips. In the dataset, the count of a tag given to a music clip is at most 12.

5.1. Evaluation setup

To train the UMFM, we randomly select 25% of the feature vectors in the dataset, which yields approximately 235,000 vectors. The GMM and MMM are trained by the Matlab software with the stopping criterion that the objective function is increased by less than 0.0001. For online fold-in estimation, the stopping ratio is 0.001. We use the top 6 tags in a co-tag pattern when K=16, as shown in Table 2, to demonstrate the tag-based music aspects. The top 6 tags in a column seem to match some specific music aspect.

To evaluate the proposed music retrieval system, we need a set of MTML queries; and for each query, we need the relevant/irrelevant labels of tracks in the music dataset. Recall that each music track in the MajorMiner dataset is associated with tags and their counts, i.e., $c(n,m)$, m=1,2,…,M, thus, the tag label of each track can be used as a test MTML query. Then, given an MTML query, the proposed content-based music retrieval system retrieves the track itself as well as other tracks that are auditorily similar to it. However, since relevance information is not available and manual labeling is not feasible, we generate the relevance information based on the tag labels of tracks in the music dataset. We assume that two music tracks will be considered auditorily relevant by a user if they have exactly the same tag label distribution, i.e., the cosine measure between their tag labels is 1. Therefore, the relevance score between two tracks can be measured by the cosine similarity (in the range 0 to 1) between their tag labels. If we take the tag label of a track $t1$ as an MTML query, the relevance score of track $t2$ can be calculated as the cosine similarity between the tag labels of $t1$ and $t2$. In this way, we can generate the relevance information for each MTML query.

We repeat three-fold cross-validation 20 times on the MajorMiner dataset, which is divided into three folds at random. In each run, 1,648 tracks are used for training and 824 tracks for testing. The tag label of each track in the test set is taken as an MTML query; hence, there are 824 MTML queries. Given a query, the retrieval system will rank the 824 music tracks based on their audio content. The ranked results are then compared with the relevance scores estimated based on the tag labels mentioned above. To evaluate the retrieval performance, we apply the normalized discounted cumulative gain (NDCG) [11]. The value of NDCG at n is calculated as follows:

$$\mathrm{NDCG}@n = Z_n \sum_{j=1}^{n} \frac{2^{R(j)} - 1}{\log(1 + j)}, \quad (12)$$

where $R(j)$ is the relevance score of the j-th track on the ranked list, and Z_n is a normalization factor that guarantees the ideal NDCG@n value will be 1.

5.2. Evaluation results and discussion

In the experiments, the number of latent feature classes is set between 16 and 2,048. The evaluation results in terms of NDCG@5 and @10 are shown in Fig. 4. We evaluate the Auto-tagging method, which applies co-tag pattern-based matching, and the Fold-in method, which applies pseudo song-based matching. As our baseline, we implement the Random method, which randomly generates the pseudo song for an MTML query.

The results in Fig. 4 show that the Auto-tagging approach outperforms the Fold-in approach. The latter seems to saturate at around K=64. The early saturation phenome-

Table 2. The latent co-tag patterns (tag-based music aspects) described by the top 6 tags when K=16.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
piano	electronic	male	drums	guitar	synth	synth	quiet	synth	guitar	guitar	piano	rap	saxophone	jazz	rock
guitar	synth	synth	electronic	synth	electronic	pop	ambient	guitar	rock	male	jazz	hip-hop	synth	saxophone	guitar
slow	dance	drums	female	drums	drums	drums	synth	drums	drums	drums	synth	male	electronic	female	drums
female	drums	guitar	dance	electronic	male	male	electronic	bass	pop	synth	electronic	funk	drums	piano	male
strings	beat	vocal	techno	bass	bass	dance	guitar	electronic	male	country	ambient	female	jazz	trumpet	vocal
vocal	techno	female	synth	punk	techno	guitar	noise	slow	bass	rock	quiet	beat	guitar	vocal	punk

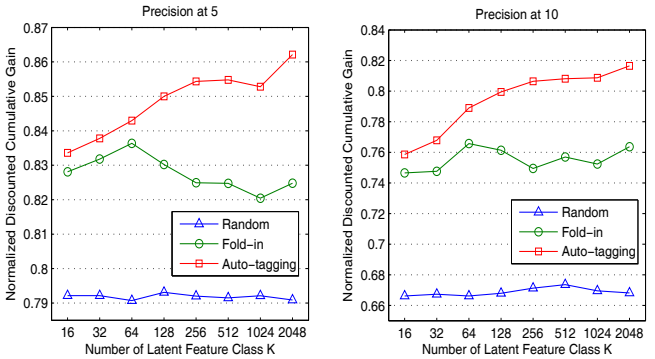

Fig. 4. The evaluation results in terms of NDCG@5 and @10.

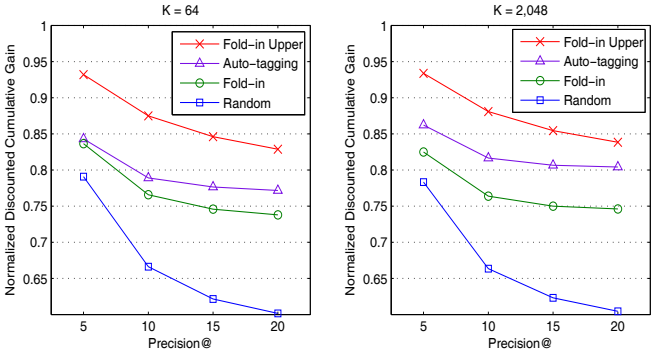

Fig. 5. The results derived by the proposed retrieval system with K=64 and K=2048.

non may be because we only use 45 tags. As the number of latent co-tag patterns is much larger than 45, some of the patterns may be redundant, i.e., some tag-based music aspects may be very similar. This may result in some randomness when estimating the pseudo song for a given MTML query; however, we believe the situation will improve if larger tag sets are considered. In contrast, the performance of the Auto-tagging approach continues to improve with the number of latent feature classes K. This is as expected because K represents the resolution of the latent feature classes. Even though some latent co-tag patterns may be redundant when K is large, the music features can be better modeled by the UMFM with a larger K. This facilitates automatic tagging, and thereby improves the performance of the Auto-tagging retrieval approach.

Figure 5 shows the results in terms of NDCG under different n. We also investigated the upper bound performance (UBP) of the Fold-in approach, denoted as "Fold-in Upper" in the figure. The approach is implemented in a query-by-example manner, i.e., the test query is the UMFM-based representation of the corresponding track instead of the UMFM-based representation estimated from its tag label. In other words, the fold-in procedure is omitted when evaluating the UBP. We observe that the UBP of NDCG@5 is about 0.93 instead of 1. The gap indicates that the objective similarity measures among tracks in fixed-dimensional feature representation are not completely consistent with the human judgments (i.e., the human-labeled tags). This may be because the music features used in this work still remain insufficient to represent music, or because the human-labeled tags contain errors or inconsistencies. We leave improving the UBP of the Fold-in method to our future work.

6. CONCLUSIONS

We have proposed a novel content-based music retrieval system that accepts multi-tags with multi-level preferences as queries, i.e., MTML queries. The limited number of popular tag choices and the preference scroll bars enable users to specify their favors of music in terms of some common musical concepts. In addition, we have presented two indexing methods and their corresponding matching methods, namely pseudo song-based matching and co-tag pattern-based matching. The tag-based music aspect models, which jointly model the auditory features and the tag-based text features of a song, are learned in a new manner based on the posterior of a pre-trained UMFM. We demonstrated the learned tag-based music aspects and evaluated the proposed MTML content-based music retrieval system on the MajorMiner dataset. The results demonstrate the potential of the system, i.e., with a feature extractor, a UMFM, and the corresponding tag-based aspect model, the system can be applied to any untagged music database.

8. REFERENCES

[1] M. Levy and M. Sandler, "Music Information Retrieval Using Social Tags and Audio," *IEEE Trans. on Multimedia*, 11(3), pp. 383-395, 2009.

[2] M. Mandel and D. Ellis, "A web-based game for collecting music metadata," *ISMIR*, 2007.

[3] E. Law, L. von Ahn, R. Dannenberg, and M. Crawford, "Tagatune: a game for music and sound annotation," *ISMIR*, 2007.

[4] D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet, "A game-based approach for collecting semantic annotations of music," *ISMIR*, 2007.

[5] P. Lamere, "Social tagging and music information retrieval," *Journal of New Music Research*, 37(2), pp. 101-114, 2008.

[6] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. on Audio, Speech and Language Processing*, 16(2), pp. 467–476, 2008.

[7] H.-Y. Lo, J.-C. Wang, and H.-M. Wang, "Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval," *ICME*, 2010.

[8] E. Law, B. Settles, and T. Mitchell, "Learning to tag from open vocabulary labels," *ECML*, 2010.

[9] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," *DAFx*, 2007.

[10] M. Levy and M. Sandler, "Learning latent semantic models for music from social tags," *Journal of New Music Research*, 37(2), pp. 137-150, 2008.

[11] K. Jarvelin and J. Kekalainen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. on Information Systems*, 20(4), pp. 422–446, 2002.