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
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.

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
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.

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

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 UFM, 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 $\pi_k$, mean vector $\mu_k$, and covariance matrix $\Sigma_k$. The conventional EM algorithm is used to learn the UFM.

Given the $i$-th feature vector $x_{ni}$ of a song $s_n$, the UFM computes the posterior probability of $z_k$ as follows:

$$p(z_k | x_{ni}) = \frac{p(z_k; \pi_k)p(x_{ni}|z_k; \mu_k, \Sigma_k)}{\sum_{k=1}^{K} p(z_k; \pi_k)p(x_{ni}|z_k; \mu_k, \Sigma_k)}.$$  \hfill (1)

By assuming that each frame contributes equally to the song, we obtain the UFM-based representation $\theta_n$, whose $k$-th component $\theta_{nk}$ is computed by

$$\theta_{nk} \leftarrow p(z_k \mid s_n) = \frac{1}{T_n} \sum_{i=1}^{T_n} p(z_k | x_{ni}),$$  \hfill (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 UFM 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 \( w=(w_1,w_2,...,w_M) \); and \( p(w | s_n; \Theta, B) \) is the tag-level MMM of song \( s_n \), where \( B \) is the parameter set of the MMM.

### 4.1. The generative process

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

\[
p(z_k | s_n; \theta_n) = \theta_{nk}.
\]

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:

\[
p(w_m | z_k; \beta_k) = \beta_{km},
\]

where \( \beta_{km} \) is the \( m \)-th parameter in the \( k \)-th vector \( \beta \) of parameter matrix \( B \). Finally, the co-tag pattern \( w \) of the song \( s_n \) is generated by marginalization over \( z_k \), which yields:

\[
p(w | s_n; \Theta_n, B) = \prod_{m=1}^{M} \sum_{k=1}^{K} p(z_k | s_n; \theta_n) p(w_m | z_k; \beta_k) = \prod_{m=1}^{M} \sum_{k=1}^{K} \theta_{nk} \beta_{km}^{c(n,m)}.
\]

Given the weight matrix \( \Theta \) whose \( n \)-th vector is \( \theta_n \) of song \( s_n \), the final log-likelihood of the training music dataset can be expressed as:

\[
L = \log p(w; \Theta, B) = \sum_{n=1}^{N} \log[p(s_n) + p(w | s_n; \Theta_n, B)]
\]

\[
= \sum_{n=1}^{N} \left[ \log p(s_n) + \sum_{m=1}^{M} c(n,m) \log \sum_{k=1}^{K} \theta_{nk} \beta_{km} \right],
\]

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

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

\[
p(z_k | s_n, w_m; \Theta, B) = \frac{p(z_k | s_n; \theta_n) p(w_m | z_k; \beta_k)}{\sum_{q=1}^{K} p(z_q | s_n; \theta_n) p(w_m | z_q; \beta_q)}.
\]

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

\[
\beta_{km} \leftarrow \frac{\sum_{n=1}^{N} c(n,m) p(z_k | s_n, w_m; \theta_n, \beta_k)}{\sum_{n=1}^{N} \sum_{m=1}^{M} c(n,m) p(z_k | s_n, w_m; \theta_n, \beta_k)}.
\]

\( \beta \) 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 co-tag pattern-based matching 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 \( \Theta \). Then, the affinity of tag \( w_m \) for the song is computed by the convex combination of mixture probabilities, each with parameter \( \beta \):

\[
p(w_m | s; \Theta, B) = \sum_{k=1}^{K} p(z_k | s; \Theta_n, B) p(w_m | z_k; \beta_k).
\]

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 UMF-based representations of the MTML query and each music track. Each song in the database is indexed by its UMF-based representation derived by Eq. (2). The MTML query represented by \( \tilde{c}(m), m=1,2,...,M \), 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 UMF-based representation. The E-step is:

\[
p(z_k | \tilde{s}, w_m; B) = \frac{p(z_k | \tilde{s}) p(w_m | z_k; \beta_k)}{\sum_{q=1}^{K} p(z_q | \tilde{s}) p(w_m | z_q; \beta_q)}.
\]

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

\[
p(z_k | \tilde{s}, w_m; B) = \frac{p(z_k | \tilde{s}; \theta_n) p(w_m | z_k; \beta_k)}{\sum_{q=1}^{K} p(z_q | \tilde{s}; \theta_n) p(w_m | z_q; \beta_q)}.
\]
\[
\tilde{\theta}_k \leftarrow p(z_k \mid \tilde{s}; \beta) = \frac{\sum_{m=1}^{M} \tilde{c}(m)p(z_k \mid s, w_m; \beta_k)}{\sum_{r=1}^{M} \tilde{c}(r)}.
\]

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., \(c(n,m)\), \(m=1,2,\ldots,M\), 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 calculated as 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 \(P\) is calculated as follows:

\[
\text{NDCG@}P = Z_P \left\{ R(1) + \sum_{i=1}^{P} \frac{R(i)}{\log_2(i)} \right\},
\]

where \(R(i)\) is the relevance score of the \(i\)-th track on the ranked list, \(i=1,\ldots,P\), and \(Z_P\) is a normalization factor that guarantees the ideal NDCG@\(P\) 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 phenomenon

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>piano</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>electronic</td>
<td>guitar</td>
<td>slow</td>
<td>female</td>
<td>drum</td>
<td>strings</td>
<td>vocal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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