Colorizing Tags in Tag Cloud: A Novel Query-by-Tag Music Search System

Ju-Chiang Wang[1,2], Yu-Chin Shih[1,2], Meng-Sung Wu[2] , Hsin-Min Wang[2] and Shyh-Kang Jeng[1]
[1] Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan
[2] Institute of Information Science, Academia Sinica, Taipei 115, Taiwan
{asriver, ycshih, wums, whm}@iis.sinica.edu.tw, skjeng@cc.ee.ntu.edu.tw

ABSTRACT
This paper presents a novel content-based query-by-tag music search system for an untagged music database. We design a new tag query interface that allows users to input multiple tags with multiple levels of preference (denoted as an MTML query) by colorizing desired tags in a web-based tag cloud interface. When a user clicks and holds the left mouse button (or presses and holds his/her finger on a touch screen) on a desired tag, the color of the tag will change cyclically according to a color map (from dark blue to bright red), which represents the level of preference (from 0 to 1). In this way, the user can easily organize and check the query of multiple tags with multiple levels of preference through the colored tags. To effect the MTML content-based music retrieval, we introduce a probabilistic fusion model (denoted as GMFM), which consists of two mixture models, namely a Gaussian mixture model and a multinomial mixture model. GMFM can jointly model the auditory features and tag labels of a song. Two indexing methods and their corresponding matching methods, namely pseudo song-based matching and tag affinity-based matching, are incorporated into the pre-learned GMFM. We evaluate the proposed system on the MajorMiner and CAL-500 datasets. The experimental results demonstrate the effectiveness of GMFM and the potential of using MTML queries to search music from an untagged music database.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.3 [**Group and Organization Interfaces**]: Web-based interaction; H.5.5 [**Sound and Music Computing**]: System.

General Terms
Algorithms, Performance, Experimentation, Human Factors.

Keywords
Tag cloud-based music query interface, MTML query, content-based music information retrieval, probabilistic fusion model.

*Area Chair: Lexing Xie

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

1. INTRODUCTION
With the explosive growth of music collections, music information retrieval (MIR) has been extensively studied in recent years. There are many ways to retrieve pieces of desired music, e.g., query by humming/singing [1], query by example [2], query by meta-information, and query by tag [3]. In this paper, we study the query-by-tag MIR task. Specifically speaking, we propose a novel content-based MIR system, which allows users to input a query of multiple tags with multiple levels of preference by colorizing desired tags in a web-based tag cloud interface to search music from an untagged music database.

There are several tagged music resources for researchers to investigate users' tagging behavior among music tracks. For example, Last.fm is a collaborative social tagging network that collects information about users' music habits in terms of music tags. However, the tagging resources collected by Last.fm may encounter a problem called tagger bias [3], which is originated from the completely non-constrained tagging environment in Last.fm. Consequently, several web-based music tagging games have been created with the objective of collecting useful tags, e.g., MajorMiner.org [4], Tag A Tune [5], and the Listen Game developed by D. Turnbull [6]. In these tagging games, music clips are randomly assigned to taggers in order to reduce the tagger bias. However, the collected music tags have only been assigned to existing music tracks; in other words, 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.

Unlike the traditional meta-information (e.g., artist(s) name, track name, and album name) and well-defined categories (e.g., genre and emotion) annotated by domain experts, music tags are free text labels generated by common Internet users. Because these tags are annotated without constraint, they can be noisy (e.g., misspelled, redundant, irrelevant, and unlimited in term numbers). It is believed that a tag will enter the common musical vocabulary once it is adopted by a large number of users, and thereby extracting tags with high term frequencies can intuitively reduce the noisy factors. This leads gradually to the emergence of the so-called folksonomy, which is a full-scale taxonomy of music that reflects the current usage among Internet users [7]. The point of view about music tags motivates several current tag-based music search interfaces. For example, Last.fm[1] highlights or enlarges

[1] http://www.last.fm/charts/toptags

those commonly used tags for users to search music information in question, such as web pages, related artists, and music playlists. This kind of interface is also known as the tag cloud, which becomes one of the key visual elements in Web 2.0. The tags in a tag cloud are usually single words and are normally listed alphabetically, and the importance of each tag is shown with font size or color [8]. Tag cloud facilitates browsing and navigating all available tags alphabetically and by popularity. There are several developments of tag cloud interface in different visual layouts [9-11]. In sum, tag cloud provides an intuitive 2-dimensional layout that reflects aggregations of tag-usage statistics. However, a tag in the traditional layout of tag cloud is usually a hyperlink that directs to a collection of items that are associated with the tag, i.e., a user can only choose a single tag at once (e.g., Last.fm), which can not completely describe the information need. Although tag cloud reveals some tagging statistics derived from the data behind the interface, it does not provide users with any interaction. In light of the above discussions, as shown in Figure 1, we propose a novel tag-based music query interface that allows users to input a query comprised of multiple tags with multiple levels of preference (denoted as an MTML query hereafter) by colorizing desired tags in a tag cloud. As will be detailed in Section 3, the tag cloud in the interface not only provides the tagging statistics of the music content but also allows users to interact with it to organize the query by manipulating the colors of the desired tags.

We believe that the MTML query can help facilitate content-based music retrieval for two reasons.

1) Unlike images, which often contain only a few clearly identifiable objects, a piece of music can be described in nature by multiple music tags. 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 the same type or different types to a specific song, and this may lead to specific tag co-occurrence (denoted as *co-tag* hereafter) patterns among auditorily similar songs. For example, instrumental or timbre tags, such as guitar, drum, rap, saxophone, and piano, 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. 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, and slow" 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.
2) A song in a tagged music database has multiple tags with different counts. The tag count corresponds to the number of users who have annotated the song with the tag, i.e., it shows the tag popularity. Therefore, the MTML query can be directly matched with the *tag count distribution* of each song in a *tagged music database* to retrieve relevant music. The MTML query can also be matched with the *tag affinity distribution* of each song in an *untagged music database*. The tag affinity distribution of an untagged song consists of the confidence degree given by each tag predictor. The MTML query actually gives a more precise query than a single tag query and a simple binary multi-tag query.

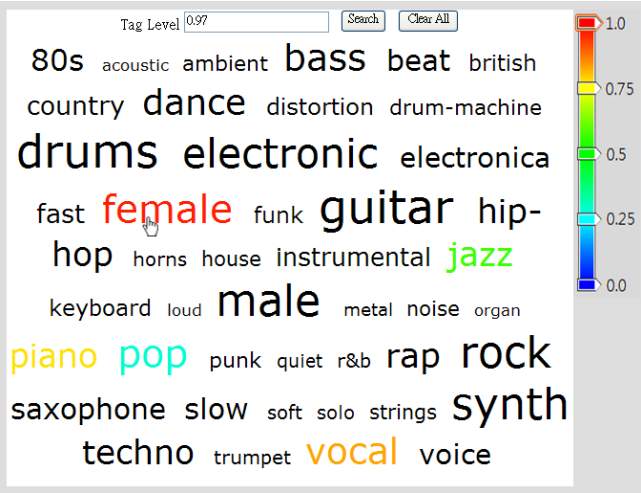


Figure 1. The screenshot of the tag cloud-based music search interface that allows a user to input a query consisting of multiple tags with multiple levels of preference by colorizing desired tags in the tag cloud. When moving the mouse pointer on a tag, the "Tag Level" (0.00~1.00) is shown in the value box in the top of the figure. The user can clicks and holds the left mouse button to change the color and level of the tag. After manipulating the colors of the desired tags, the user can click the search button to retrieve the desirable music. According to the color map in the rightmost of the figure, the user has inputted "female" with 0.97, "jazz" with 0.58, "piano" with 0.73, "pop" with 0.28 and "vocal" with 0.82.

To effect the MTML content-based music retrieval, we propose a Gaussian Multinomial Fusion Model (GMFM), which consists of two mixture models, namely a Gaussian mixture model (GMM) and a multinomial mixture model (MMM). The GMFM can jointly model the auditory features and tag labels of a tagged song. The tag cloud for the target untagged music database can be generated by automatic music tagging based on the GMFM learned from a limited tagged training music database. We also propose two indexing methods and their corresponding matching methods, namely, pseudo song-based matching and tag affinity-based matching, based on the GMFM.

The major contribution of this paper is threefold.

1) We address a new query scenario for music information retrieval, i.e., query by multiple tags with multiple levels of preference (denoted as an MTML query).
2) We design a new tag query interface that allows users to input an MTML query by colorizing desired tags in a tag cloud.
3) We propose a novel probabilistic fusion model, i.e., GMFM, which jointly model the auditory features and tag labels of a tagged song, and two indexing/matching methods based on the GMFM to effect the MTML content-based music retrieval.

The remainder of this paper is organized as follow. In Section 2, we discuss related work. In Section 3, we present our tag cloud-based MIR interface. Section 4 contains an overview of the proposed MTML-based MIR system. In Section 5, we describe the audio signal processing part. In Section 6, we introduce the GMFM and our MTML-based MIR methods. We discuss the evaluation of the proposed system in Section 7. In Section 8, we summarize our conclusion and discuss feature work.

2. RELATED WORK

A number of interfaces and approaches have been proposed for music retrieval using music tags. In [12], the authors propose a similarity measure in the space of social audio features and demonstrate an advanced music retrieval interface to retrieve songs by selecting a tag in a tag cloud on a mobile device. The tag cloud is user dependent; in other words, it only displays tags corresponding to a user's music collection. The tag cloud-based retrieval scenario is basically the same as that in Last.fm. Turnbull et al. [13] model the feature distribution of each tag with a GMM and estimate the model's parameters with a weighted mixture hierarchies expectation maximization algorithm. Hoffman et al. [14] propose a Codeword Bernoulli Average (CBA) method to model a tag's probability based on a song-level vector quantized feature representation. The training efficiency and tag prediction performance highly depend on the quality of the pre-trained codebook. Our recent work [15] applies a binary ensemble classifier comprised of SVM and AdaBoost to each tag to model its corresponding music features. The extended work in [16] shows that the tag prediction performance can be improved by considering the tag count information in auto-tagger training. All the above systems only consider retrieving music with a single tag.

In [7], Levy et al. apply text-based information retrieval (IR) techniques to music collections. They represent a music track with a joint vocabulary made up of social tags and muswords, where muswords are the quantized terms that represent the auditory characteristics of a segment-based signal in a track. The authors utilize two IR models 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. Then, they apply the probabilistic latent semantic analysis (PLSA) model on the vector representations of songs to enhance the music retrieval effectiveness and efficiency. However, in their experiments, the music database is not completely untagged because a certain percentage of label information is used in the BOW+M representation. Recently, the latent aspect model of music tags has been extended to handle open vocabulary tags [17]. The open vocabulary tags, which contain many noisy labels, are reduced to a small set of topic labels using Latent Dirichlet Allocation (LDA), and then classifiers are trained on the transformed topic labels. Although they accept free text queries, which are equivalent to multi-tag queries, they do not consider the preference of each tag.

The Heard It system [18] creates an intelligent tag-based music retrieval system with three connected components: a social game, a learning machine, and a music understanding component. First, a social Internet game [19] is developed such that players can listen to music and share their opinions online. A player actually plays with a machine instead of other players but the player is not aware of this. The design increases the playability of the game. The objective is to have players verify whether a music clip should be associated with a pre-defined tag or not. Through the game, a large amount of reliable music tags with associated music clips have been collected and used to train automatic music tagging classifiers. These classifiers are then applied to predict new songs with pre-defined tags. Finally, a content-based music search system is established, and users can retrieve or generate a music playlist via multiple tags. The tags are categorized into 6 classes, namely, emotion, characteristics, genre, instrumentation, use of music, and vocal. Users can select desired tags from the 6 classes or type the tag words in the pre-defined tag set, as shown in Figure 2. The categorization of tags helps users organize their music needs. However, users can only enter multiple *binary* tags without preference levels, and the tags in the menu do not show the social information or data distribution behind the interface.

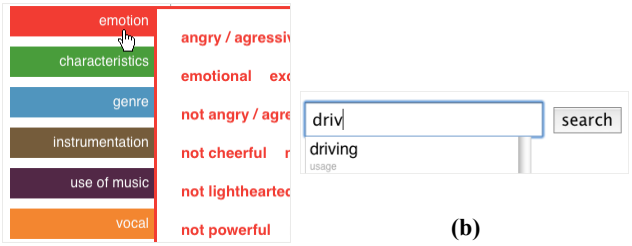


Figure 2. The tag query interface of Herd It. Users can (a) select tags from the pre-categorized menu or (b) enter the tag words in the search box.

In a word, the abovementioned tag-based MIR approaches or systems accept either single or multiple tags typed in or selected from a tag list or a tag cloud. None of them consider the preference level of a tag. However, in the proposed tag-based MIR system, the tag cloud in the interface not only provides the tagging statistics of the music content but also allows users to interact with it to organize the query by manipulating the colors of the desired tags. As will be detailed in Section 6, in the proposed GMFM, the GMM and MMM are learned in a single stage, while the models in [7, 14, 17] are learned in two stages. In this paper, we do not deal with the open vocabulary tags since the current MTML interface does not allow users to enter tags not covered by the tag cloud.

3. TAG CLOUD-BASED MIR INTERFACE

As the screenshot of the proposed web-based MIR interface shown in Figure 1, there is a music tag cloud where the font size of a tag corresponds to the popularity according to the tag frequency in the music database. Since the music database is originally untagged, the tag cloud is generated via automatic tagging, i.e., the font size of a tag is determined by the accumulated confidence degree among music clips given by the corresponding tag predictor. All tags are allocated alphabetically and initialized to black color. In the rightmost, there is a color map (from dark blue to bright red), in which each color represents a tag level ranging from 0.00 to 1.00. When a user clicks and holds the left mouse button on a desired tag, the color of the tag will change cyclically according to the color map, and the corresponding tag preference level will be synchronized with the tag color and shown in the value box of "Tag Level" in the top of the interface. Once the user releases the left mouse button, the current color and level of the selected tag will be stored. The color changing period is set to 3 seconds, i.e., the tag color changes gradually from dark blue (0.00) to bright red (1.00) in 3 seconds, and then retrogresses from bright red (1.00) to dark blue (0.00) in another 3 seconds. The user can move the mouse pointer back to a previously colored tag to modify its color and level. In this way, the user can easily organize and check the query of multiple tags with multiple levels of preference through the colored tags. When the user clicks the "search" button, the interface will submit the selected tags with corresponding levels of preference to the music search system, and then a ranked music list with related materials will be returned to the user.

Since users can obtain tagging statistics behind the MIR system from the tag cloud display, we believe that the tag cloud-based query interface can help users organize their desired musical concepts. For example, users can readily know the popularity of a tag through the font size. They may compromise their information needs by selecting some popular tags in order to retrieve more music tracks. Besides, there are infinite kinds of possible MTML queries. Slight changes to the MTML query may lead to different music ranking results. In other words, users can interact with the music query interface to further discover music in the database.

Currently, the tag-colorizing interface is implemented as a desktop interface. However, it can be directly applied to a smart mobile device where users can simply press and hold on the desired tags through its touch screen. The tag cloud layout can fit the pinch-to-zoom mechanism and be clearly displayed on the small touch screen. With such an interface, users can easily search music by inputting an MTML query through a *mouse* or a *finger* without the need of typing any words or values. We can also implement the desktop interface by using a scrolling bar for each tag. Users can increase or decrease the preference for a particular tag more instantaneously by using a mouse to control the scrolling function. However, the scrolling function-based interface may not be able to show tagging statistics of the music database.

4. SYSTEM OVERVIEW

In Section 3, we have described how users can enter the MTML query through the tag cloud-based interface to retrieve music. In this section, we will give an overview on the proposed MTML content-based music search system.

As shown in Figure 3, the proposed MTML content-based music search system is implemented in two phases: *feature indexing phase* and *music retrieval phase*. In the feature indexing phase, each music clip in the untagged music database is indexed as a fixed-dimensional vector based on the clip's audio features. We use two indexing approaches: indexing with the *auditory posterior distribution* of a music clip by using an auditory feature reference, or indexing with the *tag affinity distribution* of a music clip given by automatic music tagging. In the retrieval phase, given an MTML query from the proposed tag-colorizing interface, the content-based music search system will return a ranked list of relevant music clips by vector matching. We apply two matching methods, namely *pseudo song-based matching* and *tag affinity-based matching*; each corresponding to one of the two indexing approaches. In pseudo song-based matching, an MTML query is first transformed into a pseudo song (i.e., the predicted auditory posterior distribution of the MTML query), and then matched with the music clips in the database in the space of auditory posterior distribution. In tag affinity-based matching, an MTML query is directly used to match with the music clips in the database in the space of tag affinity distribution. Table 1 summarizes the two implementations of the MTML music search system.

4.1 Tag Affinity Prediction

We assume that human memory stores a series of latent co-tag patterns that are difficult to describe. When tagging a song, people usually choose one or more of these patterns according to the auditory characteristics of the song. Although we cannot describe the latent co-tag patterns and auditory characteristics exactly, we believe that there is a strong link between them. Therefore, as shown in Figure 4, we introduce a hidden layer of latent classes of music features (denoted as the *latent feature class* hereafter) into

Table 1. The two implementations of the MTML music search system.

Name	Indexing Vector	Matching Method
System 1	Auditory Posterior Distribution	Pseudo Song-based Matching
System 2	Tag Affinity Distribution	Tag Affinity-based Matching

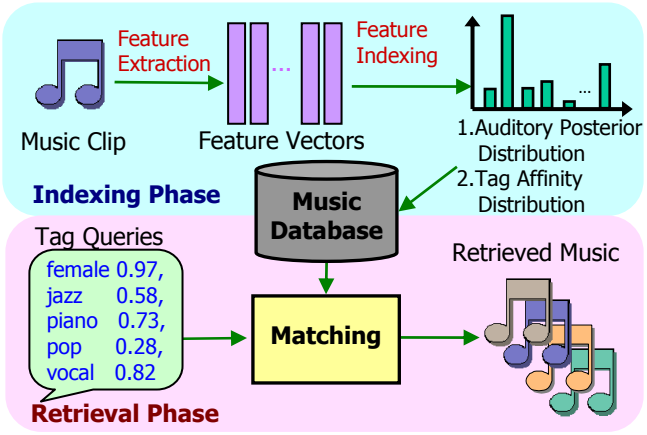


Figure 3. The flowchart of the proposed MTML content-based music search system.

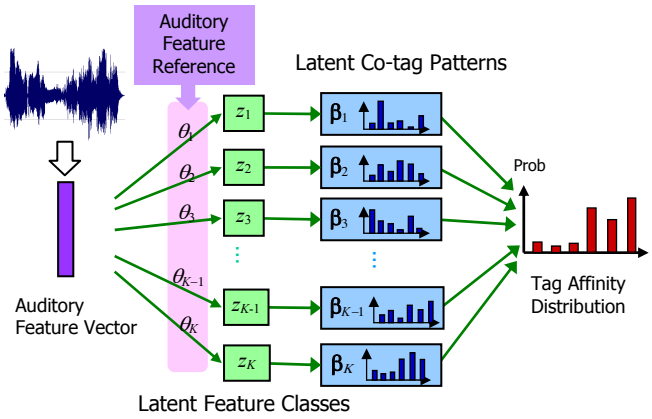


Figure 4. Indexing a music clip by automatic tagging: the prediction flow of tag affinity.

the prediction flow of tag affinity distribution to link the latent co-tag patterns and auditory features.

Assume there are K latent feature classes z_k, k=1,…,K. Each class z_k represents a group of auditory feature vectors, and its corresponding latent co-tag pattern is denoted as $\boldsymbol{\beta}_k$. A music clip is first extracted into an auditory feature vector. Then, the posterior probability (denoted as θ_k) of latent feature class z_k of the clip can be computed according to a pre-trained auditory feature reference. With $\boldsymbol{\beta}_k$, k=1,…,K, we can predict the tag affinity distribution for an untagged clip based on the value of θ_k, k=1,…,K. For example, if a clip's auditory features can be described completely by a certain latent feature class z_1, i.e., θ_1=1, and θ_i=0 for all i≠1, then its tag affinity distribution would exactly follow $\boldsymbol{\beta}_1$. To implement the idea, we assume that the tag labels with counts of a tagged song can be modeled by an MMM, where each latent co-tag pattern corresponds to a component multinomial distribution with parameter $\boldsymbol{\beta}_k$, k=1,…,K, and the auditory feature vector of the

song can be modeled by a GMM, which corresponds to the aforementioned auditory feature reference. The two mixture models (MMM and GMM) condition on the same set of latent feature classes, i.e., a mixture component in MMM corresponds to a specific mixture component of GMM. The fusion of MMM and GMM leads to the GMFM.

4.2 Pseudo Song Estimation

In System 1 in Table 1, by using the GMFM, the MTML query is folded in into the latent co-tag patterns to estimate a pseudo song in the auditory posterior distribution representation. As shown in Figure 5, an MTML query is transformed into a posterior probability vector $\boldsymbol{\lambda}$ whose k-th component is the posterior probability λ_k of latent feature class z_k, k=1,…,K. We assume that λ_k, k=1,…,K, derived from the MTML query have a similar property with the auditory posterior distribution θ_k, k=1,…,K, that is used to index music clips in the untagged music database. For example, if an MTML query is extremely like $\boldsymbol{\beta}_1$, the folding-in process will yield a dominative posterior λ_1 on z_1, i.e., λ_1=1, and λ_i=0 for all $i\neq1$, which means that the MTML query is extremely relevant to the song whose auditory posterior distribution is dominated by θ_1. Therefore, the pseudo song derived from the MTML query can be used to match with the clips in the untagged music database.

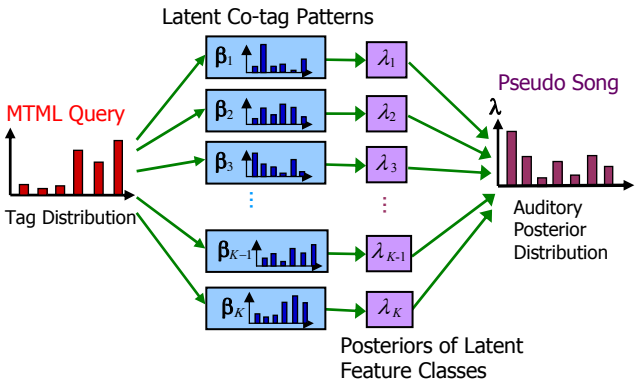


Figure 5. The estimation of the pseudo song (in the auditory posterior distribution representation) from an MTML query (in the tag distribution representation).

5. AUDIO SIGNAL PROCESSING

In this section, we describe the audio signal processing part in this work. To enhance the training efficiency and conform to the property of our music tagging modeling method, we adopt the segment-based audio feature representation instead of the frame-based one. A music clip in the database will be divided into segments and each segment is represented by a fixed-dimensional feature vector consisting of short-term temporal features and long-term perceptive features.

5.1 Audio Segmentation

Our audio segmentation is based on a measure of audio novelty proposed in [20]. An example segmentation result is shown in the bottom panel of Figure 6. We first compute the cosine measure of Mel-frequency cepstral coefficient (MFCC) vectors between any pairs of two frames (non-overlapping with size of 50ms) in a music clip, and build a self-similarity matrix, which can be visualized as a square image in the top panel of Figure 6. The color scale of a pixel in the image is proportional to the similarity. Then, we can

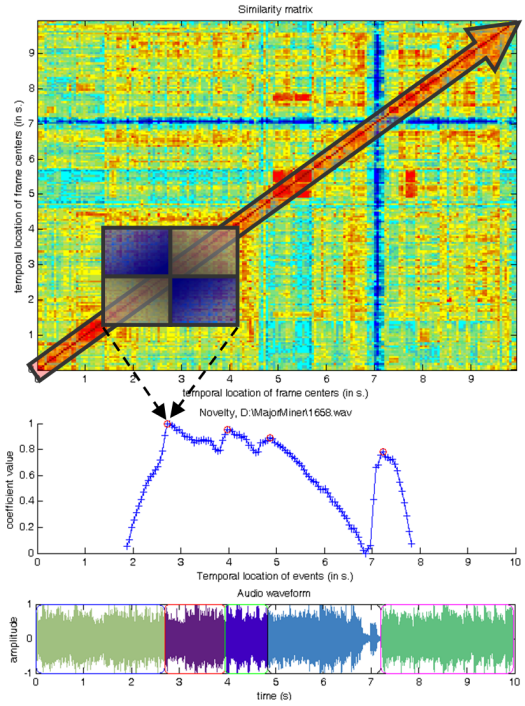


Figure 6. An illustration of audio segmentation.

obtain a time-aligned novelty curve, as shown in the middle panel of Figure 6, by convolving a checkerboard kernel with a radial Gaussian taper along the diagonal of the self-similarity matrix. The radial Gaussian taper of width H is defined as:

$$G(a,b)=\exp\left\{-4\times\left[\left(\frac{a-H/2}{H/2}\right)^2+\left(\frac{b-H/2}{H/2}\right)^2\right]\right\}, \quad (1)$$

where a, b = 1,…,H, are the horizontal and vertical indexes of the Gaussian taper, respectively. Therefore, we only need to calculate a diagonal strip of width H when constructing the self-similarity matrix. In this work, H is set to 64 (3.2sec). Finally, the local peaks of the novelty curve, as marked by circles in the middle panel of Figure 6, are selected as segment boundaries. To prevent feature extraction failures caused by insufficient data, we require the length of each segment to be at least 1 second.

Audio segmentation will divide a music clip into a dynamic number of segments with dynamic lengths. For the convenience of implementation, all the segments of a music clip are extracted into a segment-based feature vector and treated equally in training and testing. We will describe in detail the processing of segments in a music clip in Section 6.

5.2 Music Feature Extraction

To extract music features, we utilize MIRToolbox 1.3 [21], a free software that comprises approximately 50 audio/music feature extractors and statistical descriptors. As shown in Table 2, we consider seven categories of features in this work, namely, dynamic, fluctuation, rhythm, spectral, timbre, pitch, and tonal features. We set default values for parameters in MIRToolbox, such as the length of frame and hop size. Short-term (frame-based) features are represented by their mean and standard deviation calculated over the segment. After feature extraction, each segment is represented by a 180-dimensional feature vector.

Table 2. The music features and the corresponding dimension used in the segment-based feature vector.

Type	Feature Description	Dim
dynamic	rms	2
fluctuation	peak, centroid	3
rhythm	event density, pulse clarity, tempo, attack time, attack slope	7
spectral	centroid, spread, skewness, kurtosis, entropy, flatness, rolloff at 85%, rolloff at 95%, brightness, roughness, irregularity	22
timbre	zero crossing rate, spectral flux, low energy, MFCC, delta MFCC, delta-delta MFCC, zero crossing rate	78
pitch	pitch value, inharmonicity	4
tonal	key strength, key clarity, key mode possibility, HCDF, chroma peak, chroma centroid, chroma	61

6. METHODOLOGY

This section presents the methodology of the MTML music search system, which is developed based on observations from the tagged music resources. We take several music clips from the MajorMiner dataset, as shown in Table 3, as examples. Each music clip in 10 seconds long has its tag labels with counts. These music clips can be grouped into two sets according to their tags, i.e., the clips in a set have a similar tag count distribution. However, the clips in a set in Table 3 may not be grouped together based on the auditory features since there exists a gap between the similarity of auditory features and the similarity of tag count distributions. The gap may come from insufficient auditory feature extraction or representation. It is believed that the tag count distribution of a music clip represents human perception about the clip. If we group music according to the auditory features only, the grouping results may be beyond human expectation. Therefore, certain supervisions based on tag labels with counts should be considered when estimating the distribution of auditory features. However, it is impossible to collect infinite music data with label information. In this work, our objective is to jointly learn a probabilistic model that considers both the auditory features and the tag labels with counts from a limited training dataset. We hope that the model can approximately cover the mostly realistic situations. In our model, we treat the tag labels with counts of a music clip as a text feature. Therefore, a music clip contains two types of features, i.e., the auditory features and the tag features. Both of them will contribute in model learning, but the contributions may not be equal.

6.1 The Gaussian Multinomial Fusion Model

As described in Sections 4.1 and 4.2, we would like to model the auditory feature vectors by a GMM and model the tag labels with counts by an MMM. We propose a fusion model called Gaussian Multinomial Fusion Model (GMFM) to combine the GMM and MMM. We assume that both GMM and MMM have the same number of mixtures, each mixture-component-pair of GMM and MMM conditions on the same latent feature class z_k, k=1,…,K, and the corresponding Gaussian and multinomial distributions have the same mixture prior π_k. All available tags are represented as a sequence of M tags, denoted as $\mathbf{w}=(w_1,w_2,\ldots,w_M)$. Suppose we have a tagged music dataset in which the music clips are divided into a total of N segments, and each segment, denoted by s_i, i=1,…,N, is extracted into a segment-based feature vector. A segment s_i is represented by an {auditory, tag-counts} feature pair denoted by $\{\mathbf{x}_i, \mathbf{c}_i\}$, where $\mathbf{x}_i$ is the auditory feature vector of s_i ; $\mathbf{c}_i$ is the vector of tag counts (i.e., a bag-of-tags vector) of the music clip from which s_i is originated (i.e., the tag labels with counts of a music clip are shared by all its component segments); and $c(i, j)$, j=1,…,M, denotes the count of the j-th tag w_j in $\mathbf{c}_i$. The likelihood functions of GMM and MMM for segment s_i are formulated in Eqs. (2) and (3), respectively,

$$p(\mathbf{x}_i \mid s_i, \Lambda) = \sum_k \pi_k N(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

$$p(\mathbf{w} \mid s_i, \Lambda) = \sum_k \pi_k MN(\mathbf{w} \mid \mathbf{c}_i, \boldsymbol{\beta}_k) = \sum_k \pi_k \prod_j \beta_{kj}^{c(i,j)}, \tag{3}$$

where $N(\cdot)$ is a Gaussian distribution with parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$; $MN(\cdot)$ is a multinomial distribution with the parameter $\boldsymbol{\beta}_k$, where its j-th component β_{kj}, j=1,…,M, represents the probability of the j-th tag in the k-th latent co-tag pattern; Λ represents the parameter set of the GMFM. Given the training music dataset, the joint log-likelihood L over all segments is defined as follows:

$$\begin{aligned} L &= \sum_i \log p(\mathbf{x}, \mathbf{w} \mid s_i, \Lambda) = \sum_i \log\left[p(\mathbf{x} \mid s_i, \Lambda)^{\rho}\, p(\mathbf{w} \mid s_i, \Lambda)^{(1-\rho)}\right] \\ &= \sum_i \rho \log p(\mathbf{x} \mid s_i, \Lambda) + (1-\rho)\log p(\mathbf{w} \mid s_i, \Lambda) \\ &= \sum_i \left\{ \rho \log \sum_k \pi_k N(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + (1-\rho)\log \sum_k \pi_k \prod_j \beta_{kj}^{c(i,j)} \right\}. \end{aligned} \tag{4}$$

where the power weight ρ, which ranges between 0 and 1, is used because the log-likelihoods of GMM and MMM are not in a comparable scale. The role of ρ will be discussed later.

6.2 Model Inference with the EM Algorithm

The GMFM can be fitted by maximizing the joint log-likelihood L in Eq. (4) with respect to the mixture weight π_k, Gaussian parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, and multinomial parameters $\boldsymbol{\beta}_k$, k=1,…,K. We apply the Expectation Maximization (EM) algorithm to estimate the model parameters. Both types of features, namely the auditory features and the tag features, of a segment should contribute to the posterior probability of a latent feature class z_k. Therefore, in the E-step, the posterior probability of z_k given s_i is the weighted combination of the corresponding posterior probability of the

Table 3. An example 2-group clustering result of music clips from the MajorMiner dataset based on their human labeled tags.

Group	Artist Name – Song Name – Clip Starting Time	Associated Music Tags (Tag Counts)
1	Cast - Two of a Kind - 2:20	bass(3),drum(5),electric guitar(2),guitar(5),male(5),pop(3),rock(5),vocal(2)
	The Kinks - Dedicated Follower of Fashion - 1:30	bass(2),drum(4),guitar(5),male(4),pop(2),rock(4),vocal(2)
	Suede - You Belong to Me - 1:30	bass(2),drum(3),guitar(4),male(4),pop(2),rock(5)
	New Order - Turn My Way - 2:40	bass(2),drum(3),guitar(3),male(3),pop(2),rock(4),vocal(2)
2	Intermix - Sonic Ritual - 4:20	beat(2),dance(2),electronic(3),electronica(5),house(2),synth(4),techno(4)
	Underworld - Air Towel - 1:00	beat(2),dance(3),electronic(4),electronica(4),house(2),synth(5),techno(3)

GMM, i.e., $p(z_k \mid \mathbf{x}_i, s_i)$, and that of the MMM, i.e., $p(z_k \mid \mathbf{w}, s_i)$, as expressed in the following:

$$p(z_k \mid s_i) \equiv \alpha \cdot p(z_k \mid \mathbf{x}_i, s_i) + (1-\alpha) \cdot p(z_k \mid \mathbf{w}, s_i)$$
$$= \alpha \cdot \frac{\pi_k N(\mathbf{x}_i \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_q \pi_q N(\mathbf{x}_i \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)} + (1-\alpha) \cdot \frac{\pi_k \prod_j \beta_{kj}^{c(i,j)}}{\sum_q \pi_q \prod_j \beta_{qj}^{c(i,j)}}, \quad (5)$$

where α is a *leverage factor* for adjusting the contributions of the auditory features and the co-tag distribution.

In the M-step, we maximize the expected log-likelihood over the posterior probability $p(z_k \mid s_i)$ with respect to the model parameters. The update rules for the model parameters are as follows:

$$\pi_k' \leftarrow \frac{1}{N} \sum_i p(z_k \mid s_i), \quad (6)$$

$$\boldsymbol{\mu}_k' \leftarrow \frac{\sum_i p(z_k \mid s_i) \mathbf{x}_i}{\sum_i p(z_k \mid s_i)}, \quad (7)$$

$$\boldsymbol{\Sigma}_k' \leftarrow \frac{\sum_i p(z_k \mid s_i)(\mathbf{x}_i - \boldsymbol{\mu}_k')(\mathbf{x}_i - \boldsymbol{\mu}_k')^T}{\sum_i p(z_k \mid s_i)}, \quad (8)$$

$$\beta_{kj}' \leftarrow \frac{\sum_i c(i,j) p(z_k \mid s_i)}{\sum_q \sum_i c(i,q) p(z_k \mid s_i)}. \quad (9)$$

The posterior probability of a latent feature class z_k in Eq. (5) plays a linkage between the GMM and the MMM, and determines the identical mixture weight π_k of the two models in Eq. (6).

The reason why we use the addition of the two mixture models' posterior probabilities rather than the product to estimate the posterior probability of z_k in Eq. (5) is that they have very different scales. From our experiences, the likelihood of the Gaussian PDF could be extremely small (could even become 0 due to the limited computing power) owing to the high dimension of the auditory feature vector. If product is used, the learning process could be overwhelmingly dominated by the auditory features, and the learning of MMM would highly depends on the feature aggregation of the GMM. Although the GMM learned in this way may be tight, the MMM could be very loose. This does not fit our goal of learning a GMFM with properly balanced feature aggregation that leads to good performance.

The leverage factor α is designed to be tuned and validated in the experiments. Note that in Eq. (4), ρ will not affect the estimation of model parameters in the joint log-likelihood maximization process because the log-likelihood terms of the two mixture models in Eq. (4) can be maximized independently. In practice, ρ plays a role in balancing and generating a reasonable L if we require a stopping criterion based on L during the iterative model fitting process. In this work, we readily set ρ to be equal to α. The learning process of GMFM is summarized in Algorithm 1.

Algorithm 1. The learning process of GMFM

Input: Initial model parameters $\{\boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}, \boldsymbol{\beta}_k^{(0)}\}$, k=1,…,K; Training segments $\{\mathbf{x}_i, \mathbf{c}_i\}$, i=1,…,N; Leverage factor α; Stopping ratio r;

Output: Learned model parameters $\{\hat{\pi}_k, \hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k, \hat{\boldsymbol{\beta}}_k\}$;

1: Initialize a set of equal $\pi_k^{(0)}$ that sum to 1;
2: Iteration index $t \leftarrow 0$;
3: Compute $L(t)$ with Eq. (4) using $\{\pi_k^{(0)}, \boldsymbol{\mu}_k^{(0)}, \boldsymbol{\Sigma}_k^{(0)}, \boldsymbol{\beta}_k^{(0)}\}$;
4: **while** $(L(t)\text{-}L(t\text{-}1))/L(t) > r$ or $t = 0$ **do**
5: Compute the posterior probability using Eq. (5);
6: $t \leftarrow t+1$;
7: Update $\{\pi_k^{(t)}, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}, \boldsymbol{\beta}_k^{(t)}\}$ using Eqs. (6) ~ (9);
8: Compute $L(t)$ with Eq. (4) using $\{\pi_k^{(t)}, \boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Sigma}_k^{(t)}, \boldsymbol{\beta}_k^{(t)}\}$;
9: **end while**

6.3 Music Retrieval with MTML Queries

As summarized in Table 1, there are two ways to apply the GMFM to the MTML content-based MIR system. Both Systems 1 and 2 need to first convert each music clip in the database into a set of auditory posterior distribution vectors. Given a music clip S, it is first extracted into a set of segment-based feature vectors denoted as $\mathbf{x}_t$, t=1,…,T. Then, the auditory posterior distribution of segment $\mathbf{x}_t$, denoted as vector $\boldsymbol{\theta}_t$, is computed with the GMM in the pre-learned GMFM:

$$\theta_{tk} \leftarrow p(\mathbf{x}_t \mid z_k, \Lambda) = \frac{\pi_k N(\mathbf{x}_t \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_q \pi_q N(\mathbf{x}_t \mid \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)}. \quad (10)$$

where θ_{tk} is the k-th component in $\boldsymbol{\theta}_t$.

In System 1, each music segment in the database is indexed by its $\boldsymbol{\theta}$ computed by Eq. (10). Then, a given MTML query $\tilde{\mathbf{c}}$, whose component is denoted as $\tilde{c}(j)$, j=1,2…,M, is folded in into the MMM to estimate a pseudo song $\boldsymbol{\lambda}$ whose component λ_k, k=1,…,K, is estimated by

$$\lambda_k \leftarrow p(z_k \mid \mathbf{w}, \tilde{\mathbf{c}}, \boldsymbol{\beta}_k) = \frac{\pi_k \prod_j \beta_{kj}^{\tilde{c}(j)}}{\sum_q \pi_q \prod_j \beta_{qj}^{\tilde{c}(j)}}. \quad (11)$$

The pseudo song has a similar property with the auditory posterior distribution $\boldsymbol{\theta}$ as explained in Section 4.2. Therefore, we apply the well-known vector space model to compute the cosine similarity between the pseudo song and each music segment in the database. The music segments in the database are ranked by sorting the similarities from high to low. The ranking position of a music clip in the database is the *averaged* ranking positions of its component segments.

In System 2, the GMFM-based automatic tagger predicts the tag affinity distribution of each music clip in the untagged music database. Given a music clip S, we first predict the tag affinity of S's component segment $\mathbf{x}_t$, t=1,…,T. The affinity of tag w_j for segment $\mathbf{x}_t$ is computed by

$$p(w_j \mid \mathbf{x}_t) = \sum_k \theta_{tk} \beta_{kj}. \quad (12)$$

Assume each segment in a music clip is treated equally, the affinity of tag w_j for S is computed by

$$p(w_j \mid S) = \frac{1}{T} \sum_{t=1}^{T} p(w_j \mid \mathbf{x}_t). \quad (13)$$

Given an MTML query, we use the same standard matching function, i.e., the cosine similarity, to compare an MTML query $\tilde{\mathbf{c}}$, and the tag affinity distribution of each music clip in the database.

Since the two systems may complement each other, we can perform a ranking ensemble on their ranking results. The system is denoted as System 3 in the experiments.

7. EVALUATION AND RESULTS

In this section, we first examine the convergence of GMFM learning and the parameter α. Then, we evaluate the MTML content-based MIR system on the MajorMiner [4] and CAL-500 datasets.

The MajorMiner 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 crawled all the music clips associated with the most commonly used 76 tags from the MajorMiner website in March 2011. 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.

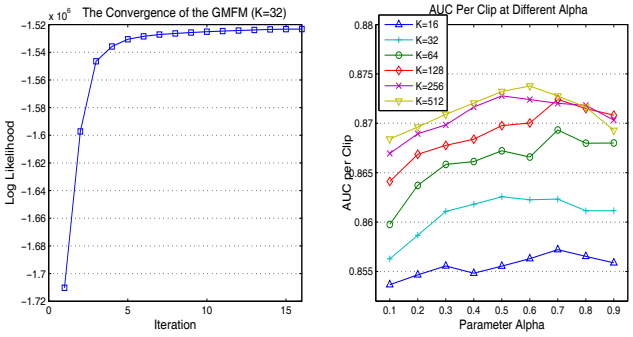
The CAL-500 dataset consists of 500 clips of western popular songs [14]. The length of a clip ranges from 3 seconds to more than 22 minutes. Each clip has been manually labeled by at least three humans following 174 pre-defined text labels. We select a subset of 111 tags, which cover categories of genre, instrumentation, usages, and vocals. In this work, the "soft-assignment" scores of tags between 0 and 1 are transformed into positive integer counts ranging from 1 to 14 for model training.

7.1 Examination of GMFM Learning

We fit the GMFM on the MajorMiner dataset to examine the convergence of the training log-likelihood. The initial GMM is derived by applying the K-means algorithm on a small portion of available segment-based vectors. The initial MMM is randomly generated and the parameters of each mixture component are normalized to sum to unity. From Figure 7(a), we can see that the log-likelihood of the GMFM (K=32) increases monotonically till the stopping criterion that the log-likelihood is increased by less than 0.0001 is reached. Note that K represents the number of latent feature classes. We also perform an automatic tagging experiment with the GMFM to validate the parameter α. The experiment is executed in a three-fold cross-validation manner, i.e., two-thirds of music clips for training and the remaining for testing. We repeat 20 runs to get the average tag prediction performance in terms of AUC (area under the ROC curve) per clip at different K and α. In each run, the dataset is divided into three folds at random. As shown in Figure 7(b), the GMFM achieves better tagging performance when α is between 0.5 and 0.7. The results indicate that the auditory features should contribute more in GMFM learning. In the following MIR experiments, α is set to the value that gives the best performance in Figure 7(b). Although the performance curves in Figure 7(b) vary slightly with K, they have a similar concave tendency. The performance is improved as K increases since a larger K yields a higher resolution of the latent feature class. But the improvement is not linearly proportional to the increase in K, and it starts to saturate at around K=256 as will be described in detail later.

7.2 Evaluation of the MTML MIR System

To evaluate the proposed MIR system, we need a set of MTML queries; and for each query, we need the relevant/irrelevant labels of music clips in the dataset. Recall that each music clip in the dataset is associated with tags and their counts, i.e., $c(i, j)$, j=1,…,M; thus, the tag labels with counts of each clip can be used as a test MTML query. Then, given an MTML query, the proposed content-based MIR system searches the music clip itself as well as other music clips that are perceptually 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 with counts of music clips in the dataset. We assume that two music clips will be considered relevant by a user if they have a highly similar tag count distribution, i.e., the cosine similarity between their tag count distributions is close to 1. For example, the music clips in a group in Table 3 are considered mutually relevant. Therefore, the ground-truth relevance R between two clips can be defined as the cosine similarity (in the range 0 to 1) between their tag count distributions. If we take the tag count distribution of a clip S_1 as an MTML query, the ground-truth relevance for clip S_2 can be calculated as the cosine similarity between the tag count distributions of S_1 and S_2. In this way, we can generate the relevance information for each MTML query.



(a) (b)

Figure 7. (a) The log-likelihood computed in Eq. (4) at each iteration of GMFM learning; (b) the automatic tagging performance in terms of AUC per clip with different K and α.

We repeat three-fold cross-validation 20 times on the MajorMiner dataset. In each run, the dataset is divided into three folds at random. We use 1,648 clips for training the GMFM and 824 clips for MIR 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 in turn, the retrieval system will rank the 824 music clips based on their audio content. The same manner is applied on the CAL-500 dataset except that we perform five-fold cross-validation. The ranked results are compared with the ground-truth relevance. To evaluate the retrieval performance, we apply the discounted cumulative gain (DCG) and normalized DCG (NDCG) [22]. DCG is formulated as follows:

$$\mathrm{DCG}@P = R(1) + \sum_{i=2}^{P} \frac{R(i)}{\log_2 i}, \tag{14}$$

where $R(i)$ is the ground-truth relevance of the i-th music clip on the ranked list. The ground-truth relevance of a retrieved music clip will contribute to DCG a non-negative value discounted logarithmically proportional to its ranked position. The DCG at P is proportional to how relevant the top P retrieved music clips are to the query. If a system retrieves more relevant clips and highly relevant clips have higher positions, it will obtain a higher DCG. If we only consider the ranking of the retrieved clips and disregard the relevance degree with respect to the whole database, we can use the normalized DCG, which is formulated as follows:

$$\mathrm{NDCG}@P = \frac{\mathrm{DCG}@P}{\mathrm{IDCG}@P}, \tag{15}$$

where IDCG@P, which guarantees the ideal NDCG@P value will be 1, is the best DCG@P that can be obtained given a set of re-

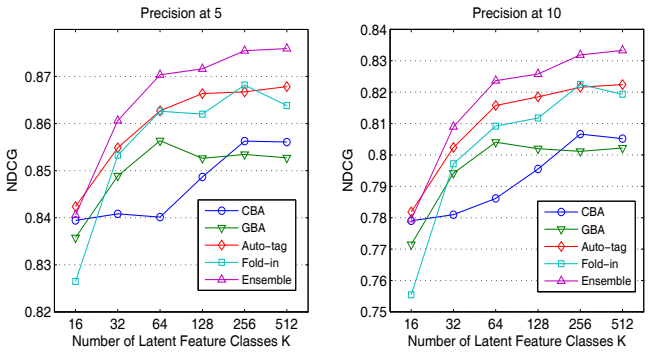


Figure 8. The NDCG results for the MajorMiner dataset.

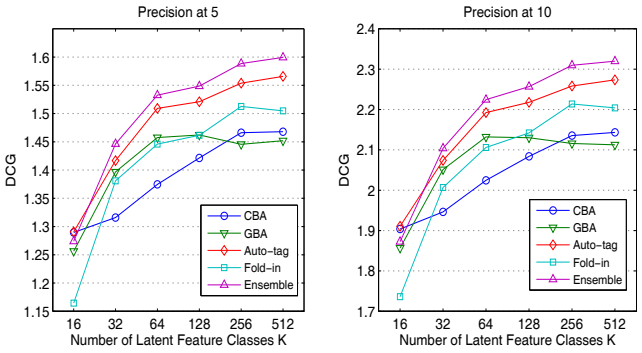


Figure 9. The DCG results for the MajorMiner dataset.

trieved clips. NDCG is practically meaningful because most users only care about the ranking of the top ranked results.

7.2.1 MIR Methods Compared

We evaluate the Fold-in method (i.e., System 1 in Table 1), the Auto-tag method (i.e., System 2 in Table 1), and the Ensemble method (i.e., System 3 mentioned in Section 6.3). The number of latent feature classes is set between 16 and 512. We also implement three methods as the baselines, namely, the codebook Bernoulli Average (CBA) method [14], the Gaussian Bernoulli Average (GBA) method, and the Random method. The CBA method starts with an unsupervised training of a codebook on available auditory feature vectors, and then each music clip is encoded as a bag-of-codewords vector. Next, a binary classifier for each tag is learned based on estimating a set of Bernoulli distributions, each corresponding to a codeword. There are several improvements with GMFM versus CBA. First, CBA is performed in a two-step learning manner in which the label information is not considered in codebook training and the learning of Bernoulli distributions fully depends on the vector grouping of the codebook. Second, CBA is not designed to model the co-tag phenomenon and tag counts since it employs an independent classifier for each tag and a Bernoulli distribution for each codeword. Third, the use of vector quantization for encoding a song limits the generalization of CBA so that it usually requires a large codebook to achieve good performance and model adaptation is not feasible when there are new songs or tag labels. In the GBA method, the codebook is replaced by a GMM. Note that K represents the size of codebook in CBA and the number of Gaussian components in GBA. The Random method is implemented in two ways: 1) randomly generating a pseudo song for an MTML query, and 2) indexing a clip with a randomly generated tag affinity. The performance of the Random method is the average performance of the two implementation ways. We also investigate the performance of the query-by-example (QBE) method. Compared with the Fold-in method, which uses the pseudo song estimated from an MTML query to search music, the QBE method uses the auditory posterior distribution of the associated music clip.

7.2.2 Results of MIR Experiments

The results of MIR experiments on the MajorMiner dataset in terms of mean NDCG@5, NDCG@10, DCG@5, and DCG@10 are shown in Figures 8 and 9, respectively. It can be found that the performance in general monotonically increases with K, but the improvements are gradually reduced. For the MajorMiner dataset, we suggest to use K=256 because of its performance and computational efficiency. From the figures, it is obvious that our methods, in particular System 2 (Auto-tag) and System 3 (Ensemble), out-

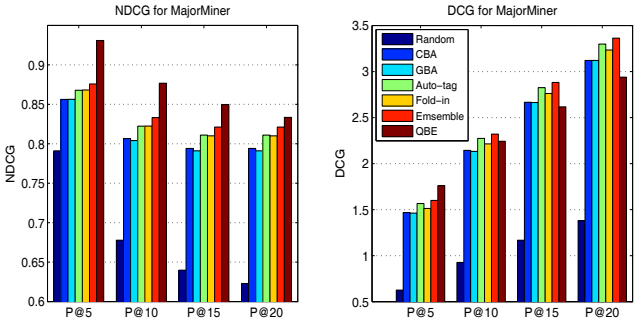


Figure 10. The comparison of the best performance of all methods on the MajorMiner dataset.

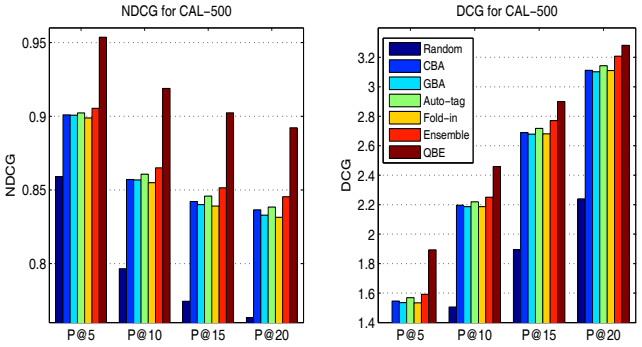


Figure 11. The comparison of the best performance of all methods on the CAL-500 dataset. The DCG@5 of the Random method is not shown because it is worse than 1.4.

perform the baselines in all cases. There could be two reasons for the superiority of System 2 (Auto-tag) over System 1 (Fold-in). First, System 2 matches the MTML query with the music clips in the database in the tag label space, based on which the ground-truth relevance R is estimated, while System 1 does matching in the auditory posterior space. Second, the GMM is fitted better than the MMM in GMFM learning because the GMFM favors the GMM and auditory features more according to α. However, the two systems can be combined as System 3 (Ensemble) to further improve the performance. The results verify that System 1 and System 2 complement each other. Although the ensemble method performs well, for the online computational efficiency, we suggest using System 2 since the folding-in procedure in System 1 may suffer from an additional step of estimating the pseudo song from the input MTML query online.

The comparison of the best performance of all methods evaluated on the MajorMiner and CAL-500 datasets are shown in Figures 10 and 11, respectively. From the figures, it is obvious that all methods outperform the Random method significantly. The QBE method is originally designed to evaluate the upper bound performance of the Fold-in method. However, it is interesting to find that QBE does not always outperform Fold-in. The reason could be as follows. The pseudo song transformed from a music clip *S*'s tag labels should have a near identical auditory posterior distribution with *S* ideally. However, the MMM learned from the latent feature classes and tag labels with counts of the training music clips may transform *S*'s tag labels with counts (i.e., the MTML query) into a pseudo song that may benefit from the ground-truth relevance estimated from the tag labels with counts in the music database, although it does not share a similar auditory posterior distribution with *S*,. This is reasonable if tag bias or noisy tags exist in the training dataset. Compared with MajorMiner, CAL-500 is a more well-labeled dataset, in which all the pre-defined tags are verified to the music clips (the tags in MajorMiner are not, due to the nature property of social tagging resources). As can be seen in Figure 11, the performance of CBA and GBA gets closer to that of Auto-tag, and CBA and GBA even outperform Fold-in in some cases, in contrast to the results for MajorMiner in Figure 10. Since the noisy factors in realistic tagged music resources can be dealt by topic modeling as discussed in [7, 17], the GMFM, which employs the concepts of topic modeling, has the advantage as well. The observation that the QBE method performs an outstanding result in terms of the NDCG metric also indicates that the similarity from auditory posterior distribution θ encoded by the GMM does properly match human's perceptual similarity.

8. CONCLUSION AND FUTURE WORK

In this paper, we have addressed a new tag-based query scenario for music information retrieval, i.e., query by multiple tags with multiple levels of preference. The MTML query scenario is accomplished by a query-by-tag MIR system with a novel tag query interface that allows users to search music by colorizing desired tags in a tag cloud. In addition to music, the tag-colorizing interface can also be applied to other multimedia documents, e.g., images and videos. To effect the content-based music retrieval, we have introduced a novel probabilistic fusion model GMFM, which consists of a GMM and an MMM, to jointly model the auditory features and tag labels of a song. Two indexing methods and their corresponding matching methods, namely pseudo song-based matching and tag affinity-based matching, are incorporated into the pre-learned GMFM. The experimental results have demonstrated the effectiveness of GMFM and the potential of using MTML queries to search music from an untagged music database.

Our future work is fivefold. First, with the tag-colorizing interface, users have the opportunity to interact with the interface and discover music in the database. We can also apply a user feedback scenario. In this way, we can collect the tags associated with a set of originally untagged music. The newly collected tagged music can be used to adapt the GMFM by using the Maximum a Posteriori (MAP) algorithm. Second, the GMFM can be extended to fit out a self-organizing map (SOM) of music tags. In this way, the tag cloud will have a more intelligent layout, which shows more information about tags to attract people to play and discover music with the system. Third, the usability of the proposed tag cloud-based interface should be further evaluated on a large untagged music database. In this paper, we only evaluate the technical method that realizes the interface on two small tagged music databases. Hopefully, we can cooperate with a music website or Internet radio station such that the user experience of the system could be practically evaluated. Fourth, we will further consider the usability of the tag cloud-based interface and the scrolling function-based interface. The integration of these two interfaces is also worthy of study. Fifth, we want to extend our method to deal with the open vocabulary tags. We do not consider the open vocabulary tags currently since the MTML interface does not allow users to enter tags not covered by the tag cloud. However, the interface will be more flexible if it also allows users to key in tags.

9. ACKOWLEDGEMENTS

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: NSC 100-2631-H-001-013. We thank the anonymous reviewers and our shepherd Dr. Lexing Xie for their helpful comments.

10. REFERENCES

[1] N. Kosugi, et al. A practical query-by-humming system for a large music database. In *Proc. of ACM MM*, 2000.

[2] W.-H. Tsai, H.-M. Yu, and H.-M. Wang. A query-by-example technique for retrieving cover versions of popular songs with similar melodies. In *Proc. of ISMIR*, 2005.

[3] P. Lamere. Social tagging and music information retrieval. *J. New Music Res.*. 37(2), pp. 101-114, 2008.

[4] M. I. Mandel and D. P. W. Ellis. A web-based game for collecting music metadata. *J. New Music Res.*. 37(2), pp. 151–165, 2008.

[5] E. Law and L. von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *Proc. of ACM CHI*. pp. 1197-1206, 2009

[6] D. Turnbull, et al. A game-based approach for collecting semantic annotations of music. In *Proc. of ISMIR*, 2007.

[7] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *IEEE Trans. on Multimedia*. 11(3), 383-395, 2009.

[8] http://en.wikipedia.org/wiki/Tag_cloud

[9] Owen Kaser and Daniel Lemire. Tag-cloud drawing: Algorithms for cloud visualization. In *Proc. of WWW*, 2007.

[10] S. Lohmann, J. Ziegler, and L. Tetzlaff. Comparison of tag cloud layouts: Task-related performance and visual exploration. T. Gross et al. (Eds.). *INTERACT 2009*, Part I, pp. 392–404, 2009.

[11] K. Knautz, S. Soubusta, and W.G. Stock. Tag clusters as information retrieval interfaces. In *Proc. of the 43rd Annual Hawaii International Conference on System Sciences (HICSS-43)*, 2010.

[12] M. Kuhn, R. Wattenhofer and S. Welten. Social audio features for advanced music retrieval interfaces. In *Proc. of ACM MM*, 2010.

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

[14] M. Hoffman, D. Blei, and P. Cook. Easy as CBA: A simple probabilistic model for tagging music. In *Proc. of ISMIR*, 2009.

[15] H.-Y. Lo, J.-C. Wang, and H.-M. Wang. Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval. In *Proc. of IEEE ICME*, 2010.

[16] H.-Y. Lo, S.-D. Lin, and H.-M. Wang. Audio tag annotation and retrieval using tag count information. In *Proc. of MMM*, 2011.

[17] E. Law, B. Settles, and T. Mitchell. Learning to tag from open vocabulary labels. In *Proc. of ECML*, 2010.

[18] http://herdit.org/music/

[19] http://apps.facebook.com/herd-it/

[20] J. Foote and M. Cooper. Media segmentation using self-similarity decomposition," In *Proc. of SPIE Storage and Retrieval for Multimedia Databases*, 2003.

[21] O. Lartillot and P. Toiviainen. A Matlab toolbox for musical feature extraction from audio. In *Proc. of DAFx*, 2007.

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