Posterior Weighted Bernoulli Mixture Model for Music Tag Annotation and Retrieval

Ju-Chiang Wang[*†], Hung-Shin Lee[*†], Shyh-Kang Jeng[*] and Hsin-Min Wang[†]
[*]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[†]Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: asriver@iis.sinica.edu.tw, hungshinlee@gmail.com, skjeng@ew.ee.ntu.edu.tw, whm@iis.sinica.edu.tw

***Abstract*— Music tags provide different types of semantic information about music. Recently, automatic music tagging has generated a great deal of interest among researchers in the field of music information retrieval. In this paper, we propose a posterior weighted Bernoulli mixture model (PWBMM) that automatically annotates a song with tags, or retrieves relevant songs given a semantic tag. The PWBMM approach improves the codeword Bernoulli average (CBA) approach in two ways. First, it uses a Gaussian mixture model-based posterior representation of music feature vectors, instead of a vector-quantized representation; second, it incorporates the counts of tags assigned to training songs when training the Bernoulli mixture model. The results of music tag annotation and retrieval experiments on the MajorMiner and CAL-500 datasets demonstrate that the PWBMM approach outperforms the CBA approach in terms of different evaluation metrics..**

I. INTRODUCTION

The goal of automatic music tagging is to model the relations between text (tags) and music. There are two types of tasks: (1) annotating an unseen song with tags; and (2) retrieving songs from a music database by inputting a pre-defined tag.

In recent years, automatic music tagging has generated a great deal of interest among researchers in the field of music information retrieval. It has been one of the evaluation tasks at the annual Music Information Retrieval Evaluation eXchange (MIREX) since 2008[1]. The system that won the MIREX 2008 competition [1] models the feature distribution of each tag with a Gaussian mixture model (GMM) and estimates the model's parameters with a weighted mixture hierarchies expectation maximization algorithm. In contrast to using probability models, Eck et al. [2] use AdaBoost to automatically generate audio tags for music recommendation. In addition, Hoffman et al. [3] propose the codeword Bernoulli average (CBA) model, which is a probabilistic model that attempts to predict the probability that a tag applies to a song based on a vector-quantized (VQ) representation of the song's audio. The codewords perform well in terms of data representation if there are sufficient clusters. However, vector quantization based on the *K*-means derived codebook reveals hard-assignment of a vector in a test song to its closest codeword; hence, the differentiation and unseen-coverage ability is not good enough.

In this paper, we try to improve the CBA approach in two ways. First, we replace the K-means derived codebook with a GMM. As a result, the sequence of frame-based feature vectors of a song is represented by a fixed-dimensional vector of posterior probabilities of the Gaussian components in the GMM, instead of the normalized counts of the codewords in the *K*-means codebook. Second, the count of a tag assigned to a song is taken into account when training the Bernoulli mixture model (BMM). The results of music annotation and retrieval experiments on two music datasets demonstrate that the new approach outperforms the CBA approach.

The remainder of this paper is organized as follows. In Section 2, we describe the music features used in this work and how we exploit GMM for music data representation. In Section 3, we introduce the Bernoulli mixture model and explain how to apply it in music tagging. The experiments and results are detailed in Section 4. Finally, we summarize our conclusions and discuss our future work in Section 5.

II. MUSIC DATA REPRESENTATION

In this section, we describe the music features used in this work, and explain how we exploit GMM posterior probabilities to transform frame-based feature vectors into song-level data representations.

A. Music Feature Extraction

To apply our model to predict a song's tags, we must first index the features of the song as a sequence of fixed-dimensional vectors. We use low-level audio features because they contain musical information that can be measured by processing the audio signal directly. The features are computed within a fixed-size frame of the signal, and the values of different features are bundled as a high dimensional vector. We do not consider high-level music descriptions, such as pitch/melody, tempo/beat, chord/bass-line, key/harmonic, and structure, as they are related to musical concepts that are difficult to define with exact values, or the corresponding extractors may lead to errors and uncertainties. For example, the musical structure cannot be described by a value, while the pitch extractor might output double pitches. A detailed discussion of high-level music descriptions and low-level audio features can be found in [4].

[1] http://www.music-ir.org/mirex/2008/

To extract music features, we use MIRToolbox 1.2.5[2], a free software that comprises approximately 50 audio/music feature extractors and statistical descriptors [5]. As shown in Table I, we consider four categories of features in this work, namely, dynamic, spectral, timbre, and tonal features. All the features are extracted in the same fixed-size short-time frame to ensure alignment and prevent the mismatch of different features in a vector. Therefore, we do not use some dynamic frame-length features (e.g., onset, event-density, pulse-clarity, attack-slop, and attack-time features), or long-term features (e.g., fluctuations and lower-energy features). A sequence of 70-dimensional feature vectors is extracted from a 16-bit, 22kHz, mono music waveform with a 50ms frame size and 0.5 hop shift.

TABLE I
THE MUSIC FEATURES USED IN THE FRAME-BASED VECTOR.

Category	Feature Description	Dim
dynamic	rms	1
spectral	centroid: mean	1
	spread: standard deviation	1
	skewness	1
	kurtosis	1
	entropy	1
	flatness	1
	rolloff at 85%	1
	rolloff at 95%	1
	brightness: high frequency energy	1
	roughness: sensory dissonance	1
	irregularity: the degree of variation of successive peaks	1
timbre	zero crossing rate	1
	spectral flux	1
	MFCC	13
	delta MFCC	13
	delta-delta MFCC	13
tonal	key clarity	1
	key mode possibility	1
	HCDF	1
	chroma peak	1
	chroma centroid	1
	chroma	12

B. GMM Posterior Representation

This step converts the frame-based feature vectors of a song, which can be of different lengths, into a fixed-dimensional vector representation of the song. In the CBA approach, this is achieved by assigning each feature vector to the closest cluster prototype (codeword), and then representing the song by a vector of the counts of pre-trained codewords, i.e., a bag of codewords representation. Our goal is to find a more accurate prototype for each cluster and estimate a better generative model to represent the observed music features from the real-valued space. The Gaussian mixture model (GMM) fits this requirement very well.

[2] http://www.jyu.fi/music/coe/materials/mirtoolbox

Hereafter, we assume that a collection of universal music feature vectors is modeled by a large GMM, called the Universal Music Features Model (UMFM). The likelihood of a music feature vector $\mathbf{x}$ is defined as

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{1}$$

where π_k , $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are, respectively, the mixture weight, mean vector, and covariance matrix of the k-th Gaussian mixture component. By considering a "latent music class" as a latent variable $z_k \in \{z_1, z_2, \ldots, z_K\}$ that corresponds to the k-th Gaussian mixture component, we obtain the conditional probability as follows:

$$p(\mathbf{x} \mid z_k = 1) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{2}$$

To train the UMFM, we first normalize the feature vectors in each dimension to mean 0 and standard deviation 1. Then, the UMFM is fitted by using the expectation and maximization (EM) algorithm to maximize the total likelihood of the feature vectors generated by the model with respect to the mixture weights, the mean vectors, and the covariance matrices.

With the UMFM, we can estimate the likelihood that a given feature vector belongs to a latent music class z_k by the posterior probability of the class:

$$p(z_k = 1 \mid \mathbf{x}) = \frac{\pi_k p(\mathbf{x} \mid z_k = 1)}{\sum_{i=1}^{K} \pi_i p(\mathbf{x} \mid z_i = 1)}. \tag{3}$$

By assuming that each frame of a given song, s_j, contributes equally to the song, the posterior probability of a certain latent music class can be computed by

$$p(z_k = 1 \mid s_j) = \frac{1}{N_j} \sum_{n=1}^{N_j} p(z_k = 1 \mid \mathbf{x}_{jn}), \tag{4}$$

where $\mathbf{x}_{jn}$ is the feature vector of the n-th frame of s_j and N_j is the number of frames in s_j. The posterior representation of a song in Eq. (4) can be modeled by the generative process of the Bernoulli mixture model, as will be described later.

III. BERNOULLI MIXTURE MODEL

The Bernoulli mixture model (BMM), also known as *latent class analysis* [6], is described in detail in [7]. Here, we explain how to apply BMM to the music tagging problem through a generative process and model inference. We also describe how to incorporate the counts of tags assigned to the songs in BMM training. Assume that we have a music corpus with J songs, each denoted by s_j, j=1,…,J; and let each song's tag count c_{jw}, w=1,…,W, be a positive integer representing the number of times that tag t_w has been assigned to song s_j. The binary random variable $\boldsymbol{y}$, with $y_{jw} \in \{0,1\}$, determines whether or not tag t_w applies to song s_j.

A. Generative Process

The generative process of BMM involves two steps. First, the probability of the latent class z_{kw}, k=1,…,K, is chosen from song s_j's class weight vector $\boldsymbol{\theta}_j$:

$$p(z_{kw}=1 \,|\, \boldsymbol{\theta}_j) = \theta_{jk}, \tag{5}$$

where θ_{jk} is the weight of the k-th latent class. Then, a case of the discrete variable y_{jw} is selected based on the following conditional probabilities:

$$\begin{aligned} p(y_{jw}=1 \,|\, z_{kw}=1, \boldsymbol{\beta}) &= \beta_{kw} \\ p(y_{jw}=0 \,|\, z_{kw}=1, \boldsymbol{\beta}) &= 1-\beta_{kw} \end{aligned}, \tag{6}$$

where $\boldsymbol{\beta}$ is a K-by-W parameter matrix recording β_{kw} indexed in the k-th row and the w-th column. The conditional probability, which models the probability that song s_j is annotated with tag t_w, is a Bernoulli distribution with input discrete variable y_{jw} and parameter $\boldsymbol{\beta}$ for the k-th class z_{kw}.

The complete joint distribution over $\boldsymbol{y}$ and $\boldsymbol{z}$ is described by the model parameter $\boldsymbol{\beta}$ and weight matrix $\boldsymbol{\Theta}$ whose row vector is $\boldsymbol{\theta}_j$ of song s_j:

$$p(\boldsymbol{y}, \boldsymbol{z} \,|\, \boldsymbol{\beta}, \boldsymbol{\Theta}) = \prod_{j=1}^{J} p(\boldsymbol{y}, z_{kw}=1 \,|\, \boldsymbol{\beta}, \boldsymbol{\theta}_j) = \prod_{j=1}^{J}\prod_{w=1}^{W} \theta_{jk}^{z_{kw}} \beta_{kw}. \tag{7}$$

The marginal log likelihood of the corpus can be expressed as:

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\Theta}) = \sum_{j=1}^{J}\sum_{w=1}^{W} \log\left\{ \sum_{k=1}^{K} \theta_{jk}^{z_{kw}} \beta_{kw} \right\}. \tag{8}$$

B. Model Inference by the EM Algorithm

The Bernoulli mixture model can be fitted with respect to parameter $\boldsymbol{\beta}$ and weight matrix $\boldsymbol{\Theta}$ by maximum-likelihood (ML) estimation. In this work, we assume that the latent class of BMM corresponds to the latent music class of UMFM described in Sec. II. In other words, the latent classes have been determined by GMM fitting in the music data representation stage. Each song that has been transformed into the posterior weights $\theta_{jk} = p(\, z_k=1 \,|\, s_j \,)$ by Eq. (4) provides observation information to BMM; therefore, we only need to estimate $\boldsymbol{\beta}$, which corresponds to the probability of a latent music class. We apply the EM algorithm to maximize the log-likelihood in Eq. (8) in the presence of a latent variable $\boldsymbol{z}$.

In the E-step, given the song-level weight matrix $\boldsymbol{\Theta}$ and the model parameter $\boldsymbol{\beta}$, the posterior probability of each latent variable z_{kw} can be computed by

$$\begin{aligned} \gamma_j(z_{kw}) &= p(z_{kw}=1 \,|\, \boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{y}) \\ &= \frac{p(y_{jw} \,|\, z_{kw}=1, \boldsymbol{\beta}) p(z_{kw}=1 \,|\, \boldsymbol{\theta}_j)}{p(y_{jw} \,|\, \boldsymbol{\beta}, \boldsymbol{\theta}_j)} \\ &= \begin{cases} \frac{\theta_{jk}\beta_{kw}}{\sum_{i=1}^{K}\theta_{ji}\beta_{iw}} & \text{for } y_{jw}=1 \\ \frac{\theta_{jk}(1-\beta_{kw})}{\sum_{i=1}^{K}\theta_{ji}(1-\beta_{iw})} & \text{for } y_{jw}=0. \end{cases} \end{aligned} \tag{9}$$

In the M-step, taking the expectation of the log likelihood in Eq. (7) with respect to the posterior probability of the latent variable in Eq. (9) yields

$$E_{\boldsymbol{z}}\left[\log p(\boldsymbol{y}, \boldsymbol{z} \,|\, \boldsymbol{\beta}, \boldsymbol{\Theta})\right] = \sum_{j=1}^{J}\sum_{k=1}^{K} \gamma_j(z_{kw}) \left\{ \log\theta_{jk} + \sum_{w=1}^{W} \left[y_{jw} \log\beta_{kw} + (1-y_{jw}) \log(1-\beta_{kw}) \right] \right\}. \tag{10}$$

The derivative of the expectation with respect to β_{kw} is

$$\begin{aligned} \frac{\partial}{\partial\beta_{kw}} E_{\boldsymbol{z}}\left[\log p(\boldsymbol{y}, \boldsymbol{z} \,|\, \boldsymbol{\beta}, \boldsymbol{\Theta})\right] &= \sum_{j=1}^{J} \gamma_j(z_{kw}) \left(\frac{y_{jw}}{\beta_{kw}} - \frac{1-y_{jw}}{1-\beta_{kw}} \right) \\ &= \frac{\sum_j \gamma_j(z_{kw}) y_{jw} - \sum_j \gamma_j(z_{kw}) \beta_{kw}}{\beta_{kw}(1-\beta_{kw})}. \end{aligned} \tag{11}$$

By setting the derivative to zero, we obtain the update rule for β_{kw} as follows,

$$\beta_{kw} \leftarrow \frac{\sum_j \gamma_j(z_{kw}) y_{jw}}{\sum_j \gamma_j(z_{kw})}. \tag{12}$$

From Eq. (12), it is clear that β_{kw} will increase if y_{jw} is equal to 1. In other words, y_{jw} =1 makes song s_j contribute a positive value to β_{kw} once in each EM iteration if the song is highly related to the latent music class z_{kw} (i.e., if θ_{jk} has a large probability). From the tag counts of the music corpus, we know that there are different levels of relations between a song and a tag. If song s_j has a more than one tag count c_{jw} for tag t_w, we can make song s_j contribute to β_{kw} c_{jw} times, rather than once, in each iteration of EM. This leads to a new update rule for β_{kw}:

$$\beta_{kw} \leftarrow \frac{\sum_j c_{jw} \gamma_j(z_{kw}) y_{jw}}{\sum_{j, y_{jw}=1} c_{jw} \gamma_j(z_{kw}) + \sum_{j, y_{jw}=0} \gamma_j(z_{kw})}. \tag{13}$$

The new update rule in Eq. (13) is equivalent to duplicating song s_j c_{jw} times in the training corpus.

C. Predicting Tags of New Songs

An untagged song s is first transformed into a sequence of music feature vectors, after which the posterior weight vector $\boldsymbol{\theta}$ is computed from the pre-trained UMFM. Finally, the score for song s to be annotated with tag t_w is computed as the

conditional probability of y_w =1 given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, which is the model parameter of BMM:

$$p(y_w = 1|\boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k p(y_w = 1|\beta_{kw}) = \sum_{k=1}^{K} \theta_k \beta_{kw}. \quad (14)$$

The scores associated with the pre-defined tags provide a reference for annotating a new song by following the ranking of the conditional probabilities of all tags, or by retrieving songs in decreasing order of the conditional probabilities of the songs associated with a particular tag.

For the annotation task, the M most likely tags are selected by ranking the tag scores for a new song, where M is rounded off based on the average number of tags for a song in the training corpus. In the retrieval task, the relations between the scores and the binary classes (either relevant or irrelevant to a tag) are expressed by a quadratic decision function with the Mahalanobis metric [8]. The corresponding coefficients can be determined by performing an inside validation test on the training music corpus. However, we found that, if a song has several tags, the scores of the top tags are usually lower than those of a song with a few tags. Therefore, the scores of a particular tag across different songs might not be comparable. To solve this problem, we normalize the scores of different tags for a song by subtracting d times their mean, where d denotes the "diversity factor" [3]. However, the normalization process may yield negative values for some tags. In such cases, at the end of the normalization process, we set the negative values to positive values smaller than the smallest positive value without changing their order. Then, it is straightforward to set a pruning threshold on the normalized scores for a specific tag across different songs in the retrieval process. Since this method maintains the ranking order of tags for a song, it does not affect the tagging result of a song.

Although the GMM posterior representation of a song may loose some information about the song, the entire approach is very efficient. The computational cost only depends on the number of latent classes K and the length of the given song. If we ignore the feature extraction part, the tag-prediction procedure is almost performed on the fly. Therefore, our approach provides an effective solution for an on-line music tagging system.

IV. Experiments

A. Datasets

We evaluate the proposed approach on the MajorMiner and CAL-500 datasets.

The MajorMiner dataset is taken from the MajorMiner website[3], 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 particular music clip, the label is adopted by the system. Hence, each music clip's tag count is at least 2. Following to the audio tag classification task in the MIREX competition, 45 popular tags are considered in the tag set. We download all the music clips associated with those 45 tags from the MajorMiner website. The resulting dataset contains 2,441 music clips. In the dataset, the count of a tag given to a music clip is at most 12. The 45 tags in the MajorMiner dataset are listed in TABLE II.

TABLE II
THE 45 TAGS IN THE MAJORMINER DATASET

metal	instrumental	horns	piano	guitar
ambient	saxophone	house	loud	bass
fast	keyboard	electronic	noise	british
solo	electronica	beat	80s	dance
jazz	drum machine	strings	pop	r&b
female	rock	voice	rap	male
slow	vocal	quiet	techno	drum
funk	acoustic	distortion	organ	soft
country	hip hop	synth	trumpet	punk

CAL-500 is a music corpus comprised of 500 clips of popular Western songs [1]. The length of the clips ranges from 3 seconds to more than 22 minutes. Each clip has been manually labeled by at least three humans based on 135 predefined musically-relevant concepts, resulting in 174 distinct text labels. In this work, the "soft-assignment" annotation labels with a multi-level degree in the range [0, 1] are transformed into positive integer counts ranging from 1 to 14. We extract our own music features from the clips instead of using the MFCC features provided in the corpus.

B. Model Training

For each music clip in the training data, we extract a sequence of 70-dimensional feature vectors from a 16-bit, 22kHz, mono music waveform with a 50ms frame size and 0.5 hop shift. To train the UMFM used in the proposed approach and the codebook used in the CBA approach, we randomly select 25% of the feature vectors from each clip. There are approximately 235,000 vectors for the MajorMiner dataset and 821,000 vectors for the CAL-500 dataset. The mixture number and the codebook size are set between K=128 and K=2,048 for the MajorMiner dataset and between K=128 and K=1,024 for the CAL-500 dataset. We do not consider the K=2,048 case for the CAL-500 dataset due to the high computational overhead. To train the 70-dimensional UMFM with 20 iterations on the K-means-derived seeds, we utilize the HTK 3.4 toolkit [9]. The codebook is trained by using the Matlab software with the stopping criterion that the total distance is reduced by less than 0.0001 compared to that of the previous iteration. With the UMFM and the codebook, we can estimate two kinds of music data representation for a song, namely GMM posterior representation and vector quantized representation.

Using the posterior weights estimated by Eq. (4) for each training clip, we train the BMM with an initial matrix $\boldsymbol{\beta}$, in which all entries are set to 0.5. By iteratively updating β_{kw} using Eq. (13), the training process stops when the increase

[3] http://majorminer.org/

ratio of the log-likelihood in Eq. (8) is less than 0.0001. This method is denoted as PWBMM-II. We can apply the same posterior weights in the CBA approach. The BMM is trained by using Eq. (12) without considering the tag counts. This method is called PWBMM-I.

C. Experimental Setup

The experiments evaluate the proposed approach in terms of two tasks: annotation and retrieval. We perform three-fold evaluation on the MajorMiner dataset based on the MIREX audio tag classification evaluation setup, and ten-fold evaluation on the CAL-500 dataset following the work in [3]. In other words, there are three runs for the MajorMiner dataset, each using two folds for training and one fold for testing; and ten runs for the CAL-500 dataset, each using nine folds for training and one fold for testing. Each dataset is divided into folds randomly.

In the annotation task, the number of tags assigned to a clip is the same as the average number of tags assigned to the clip in the training data. Under our implementation, the system selects the 5 top tags for a clip in the MajorMiner dataset and the top 34 tags for a clip in the CAL-500 dataset. The performance is evaluated in terms of the precision rate, recall rate, and F-measure, as well as the area under the ROC curve per clip (Clip AUC).

In the retrieval task, given a tag, we first apply heuristic normalization (described in Sec. III) to the scores of each clip. Here, d is set at 1.5. Then, we apply the pre-trained quadratic decision function with the Mahalanobis metric on the normalized scores to retrieve the relevant clips. In the CBA method, given a tag t_w, the test clips are ranked according to $p(y_{jw}=1 \mid \boldsymbol{\beta}, \boldsymbol{\theta}_j)$ directly. The performance is evaluated in terms of F-measure and AUC per tag (Tag AUC). Note that, in the CAL-500 dataset, sometimes no clips should be retrieved for a tag according to the ground truth label (i.e., no clip in the test fold is associated with the tag) due to the large and diverse nature of the tag set. We exclude these tags because we cannot calculate the AUC for them.

D. Experimental Results and Discussions

We evaluate three approaches in the experiments: the CBA approach, the PWBMM-I approach and the PWBMM-II approach. The CBA approach is taken as the baseline. The results of experiments on the MajorMiner and CAL-500 datasets in terms of the mean and standard deviation (in parentheses) of the performance over multiple runs are summarized in TABLE III and TABLE IV, respectively. The values in bold font represent the best performance of each evaluation metric.

The experiment results suggest that the proposed PWBMM-based methods tend to perform better in annotation tasks than in retrieval tasks. This is because the methods were originally designed for annotation only. To apply them in the retrieval task, we adopt a simple normalization strategy to manipulate the scores and a simple decision function for thresholding. Although the two methods are not good enough, their performance is comparable to that of the methods evaluated on the MajorMiner dataset at MIREX 2009[4].

For the annotation task, the PWGMM-I approach, which uses the GMM posterior weights, outperforms the CBA approach, while PWGMM-II, which also considers the tag counts in training, outperforms PWGMM-I. The results verify that using GMM posterior representation and considering the tag counts of the training clips in PWBMM training improve the performance.

For the PWBMM-I and PWBMM-II approaches, increasing K does not yield a significant improvement, in contrast to the CBA approach. This is because GMM-based music data representation is more capable of representing the music feature space than codebook-based music data representation. As a result, the feature modeling ability of a 128-mixture UMFM is as good that of the 1,024-centroid codebook in terms of the Clip AUC metric.

The CBA approach outperforms the PWBMM-I approach in terms of tag precision on the MajorMiner dataset. A clip in the MajorMiner dataset (~10 seconds) is much shorter than a clip in the CAL-500 dataset (usually >1 minute). Since short clips can not express many scenarios and they are usually consistent in only a few musical concepts, their music features may show strong and obvious characteristics that really match the capability of vector quantization. In other words, sometimes a short clip can be well represented by only few codewords. However, the good performance of the PWBMM-I approach on the CAL-500 dataset indicates that a long clip prefers a more generative model. This is because a long clip contains abundant information that needs a softer representation to maintain its entropy within a fixed-dimensional vector. We have not observed the convergence of performance with respect to an increase in K, which means the performance may continue to improve as K increases.

V. CONCLUSIONS

In this paper, we have proposed a new music tag annotation and retrieval approach based on the posterior weighted Bernoulli mixture model. The approach uses GMM posterior representation of music feature vectors and incorporates the tag counts of training songs when training the Bernoulli mixture model. The results of experiments on the MajorMiner and CAL-500 datasets demonstrate that our approach outperforms the CBA approach in terms of different evaluation metrics.

The current implementation gives equal weight to each frame in a song when constructing the song-level GMM posterior representation. We believe that short-time fixed-length frames do not contribute equally to a specific tag. For long music clips in particular, different segments, such as the intro, verse, chorus, and bridge, may provide different information to trigger an individual's perception of the music. Through automatic music structure analysis and segmentation, we may find a better way to represent a song by giving higher weights to those music hooks. In our future work, we will also

[4] http://www.music-ir.org/mirex/wiki/2009:MIREX2009_Results

TABLE III
THE RESULTS OF EXPERIMENTS ON THE MAJORMINER DATASET.

		Annotation (M = 5)				Retrieval	
Model Name	***K***	**Clip F-Measure**	**Precision**	**Recall**	**Clip AUC**	**Tag F-Measure**	**Tag AUC**
CBA	128	0.4415 (0.0052)	0.4391	0.4458	0.8476 (0.0005)	0.1450 (0.0023)	0.7452 (0.0043)
	256	0.4516 (0.0113)	0.4498	0.4557	0.8532 (0.0038)	0.1586 (0.0068)	0.7491 (0.0143)
	512	0.4596 (0.0159)	0.4555	0.4600	0.8532 (0.0070)	0.1741 (0.0088)	0.7558 (0.0065)
	1,024	0.4660 (0.0116)	0.4646	0.4705	0.8597 (0.0017)	0.1734 (0.0051)	0.7555 (0.0061)
	2,048	0.4668 (0.0072)	**0.4662**	0.4708	0.8606 (0.0039)	0.1806 (0.0098)	0.7536 (0.0033)
PWBMM-I	128	0.4576 (0.0054)	0.4116	0.5174	0.8609 (0.0014)	0.2703 (0.0069)	0.7494 (0.0038)
	256	0.4614 (0.0078)	0.4153	0.5210	0.8629 (0.0097)	0.2780 (0.0049)	0.7501 (0.0076)
	512	0.4663 (0.0075)	0.4205	0.5271	0.8635 (0.0073)	0.2785 (0.0048)	0.7505 (0.0097)
	1,024	0.4690 (0.0095)	0.4230	0.5294	0.8662 (0.0050)	0.2838 (0.0035)	0.7601 (0.0035)
	2,048	0.4767 (0.0070)	0.4292	0.5414	0.8680 (0.0043)	0.2869 (0.0055)	0.7637 (0.0087)
PWBMM-II	128	0.4772 (0.0131)	0.4340	0.5265	0.8652 (0.0016)	0.2725 (0.0024)	0.7653 (0.0106)
	256	0.4805 (0.0131)	0.4305	0.5305	0.8717 (0.0035)	0.2762 (0.0052)	0.7640 (0.0097)
	512	0.4868 (0.0168)	0.4399	0.5292	0.8752 (0.0044)	0.2803 (0.0029)	0.7674 (0.0056)
	1,024	0.4891 (0.0117)	0.4435	0.5339	0.8720 (0.0052)	0.2828 (0.0022)	0.7736 (0.0048)
	2,048	**0.4945** (0.0148)	0.4493	**0.5402**	**0.8782** (0.0060)	**0.2900** (0.0067)	**0.7810** (0.0070)

TABLE IV
THE RESULTS OF EXPERIMENTS ON THE CAL-500 DATASET.

		Annotation (M = 34)				Retrieval	
Model Name	***K***	**Clip F-Measure**	**Precision**	**Recall**	**Clip AUC**	**Tag F-Measure**	**Tag AUC**
CBA	128	0.5664 (0.0042)	0.5706	0.5730	0.8554 (0.0051)	0.2686 (0.0079)	0.6796 (0.0086)
	256	0.5689 (0.0013)	0.5735	0.5753	0.8568 (0.0054)	0.2682 (0.0040)	0.6823 (0.0111)
	512	0.5710 (0.0154)	0.5733	0.5796	0.8579 (0.0071)	0.2747 (0.0076)	0.6919 (0.0158)
	1,024	0.5738 (0.0153)	0.5787	0.5800	0.8600 (0.0076)	0.2743 (0.0160)	0.6979 (0.0101)
PWBMM-I	128	0.5716 (0.0150)	0.5730	0.5751	0.8575 (0.0085)	0.3255 (0.0069)	0.6995 (0.0187)
	256	0.5723 (0.0152)	0.5744	0.5760	0.8596 (0.0087)	0.3271 (0.0170)	0.6992 (0.0270)
	512	0.5770 (0.0172)	0.5746	0.5810	0.8601 (0.0076)	0.3309 (0.0138)	0.7007 (0.0171)
	1,024	0.5789 (0.0137)	0.5783	0.5845	0.8651 (0.0036)	0.3328 (0.0098)	0.7004 (0.0184)
PWBMM-II	128	0.5782 (0.0179)	0.5796	0.5755	0.8605 (0.0090)	0.3284 (0.0183)	0.7045 (0.0179)
	256	0.5774 (0.0195)	0.5782	0.5774	0.8621 (0.0068)	**0.3336** (0.0132)	0.7071 (0.0151)
	512	0.5804 (0.0195)	0.5811	0.5794	0.8621 (0.0058)	**0.3336** (0.0099)	0.7099 (0.0171)
	1,024	**0.5853** (0.0190)	**0.5829**	**0.5856**	**0.8695** (0.0035)	0.3329 (0.0167)	**0.7128** (0.0087)

consider more temporal music characteristics to improve the GMM posterior representation.

ACKNOWLEDGEMENTS

This work was partially supported by the Taiwan e-Learning and Digital Archives Program (TELDAP), which is sponsored by the National Science Council of Taiwan under Grant: NSC99-2631-H-001-020. The authors wish to thank the anonymous reviewers for their valuable comments and suggestions, which helped improve the quality of the paper.

REFERENCES

[1] 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*, vol. 16, no. 2, pp. 467–476, 2008.

[2] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation", *Advances in Neural Information Processing System 21*, MIT Pressed, 2007.

[3] M. Hoffman, D. Blei, and P. Cook, "Easy as CBA: A simple probabilistic model for tagging music," *International Society for Music Information Retrieval Conference*, Japan, 2009.

[4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, Vol. 96, No.4, April 2008.

[5] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio", *International Conference on Digital Audio Effects (DAFx)*, Bordeaux, 2007.

[6] G. J. McLachlan and D. Peel, *Finite Mixture Models*, Wiley, 2000.

[7] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[8] W. J. Krzanowski, *Principles of Multivariate Analysis: A User's Perspective*, New York: Oxford University Press, 1988.

[9] S. Young et al., *The HTK Book*, Version 3.4.