PLAYING WITH TAGGING: A REAL-TIME TAGGING MUSIC PLAYER

Ju-Chiang Wang[1,2]*, Hsin-Min Wang*[2] *and Shyh-Kang Jeng*[1]

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

ABSTRACT

Visualizing audio signals during playback has long been a fundamental function of music players. However, most visual effects are generated by audio signal processing directly and render meaningless or incomprehensible displays to users. In this paper, we present an intelligent music player called the Playing with Tagging (PWT) music player. By integrating a real-time music tagger, the PWT player can display dynamic tag distributions via a set of tag bars that move in sync with the music. To synchronize the tag distributions, the music tagger must be able to *online recognize* the music tags. We utilize a Gaussian mixture model (GMM) as an auditory feature encoding reference and a mixture of tag-based aspect models (TBAMs) to predict the tag distribution for a short sliding chunk of the music played. To evaluate the real-time tagging function, we simulate tag prediction on short music chunks. The results of experiments on the MajorMiner dataset demonstrate the potential and effectiveness of the proposed music tagging method.

Index Terms— Online music tag annotation, real-time music visualization, probabilistic tag-based aspect model.

1. INTRODUCTION

Visualizing audio signals during playback has long been a fundamental function of music players. For example, digital home stereo systems are usually equipped with a small screen that shows the bar chart of the spectrum; and Windows Media Player provides several visual effects, such as splashes of color, geometric shapes, and random graphs, for users to select when playing music. Most visual effects are generated directly by audio signal processing based on low-level features, such as the time domain waveform or the frequency domain spectrum. Such visualizations are usually meaningless or incomprehensible because a pure audio signal does not contain any semantics and the user cannot read it directly. It would be more interesting and entertaining if the music player could display the musical concepts in sync with the music being played. The information would also help general users understand the semantic meaning that the music track is intended to express. In this paper, we present an intelligent music player called the Playing with Tagging (PWT) music player, which incorporates a real-time music tagger that recognizes and displays the dynamic tag distribution of the current audio content during playback. As shown in Figure 1, the bars, which represent the current tag distribution, stretch and shrink with the music just like a moving spectrum. The music player can *online recognize* the music tags of a sliding chunk of the music and *synchronize* the display of the predicted tag distribution with the music being played. The music player can be used on any platform of computational devices.

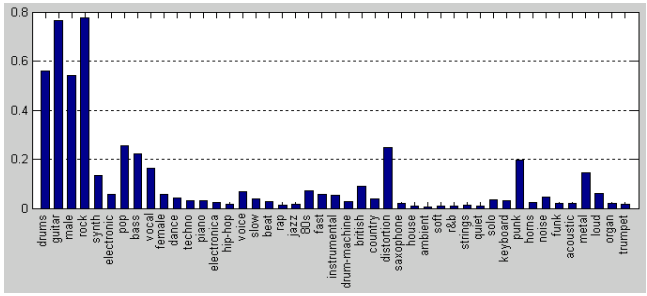


Figure 1. A screenshot of the PWT music player's visualization function at 00:50 for the first chorus of the song "Ripcord" from Radiohead's album "Pablo Honey." The bar chart shows the dynamic tag distribution and moves in sync with the music.

Music tags are generated from a folksonomy, which is a full-scale taxonomy of music that reflects the current usage among Internet users [1]. For example, Last.fm is a collaborative social tagging network that collects information about users' music habits in terms of music tags. In addition, several web-based music tagging games, such as MajorMiner [2] and Tag A Tune [3], have been created with the purpose of collecting useful tags. These applications provide tagged music resources for researchers to investigate users' tagging behavior. The tags may provide different facets of information about the music, such as genre, mood, instrumentation, and original artist(s). However, the collected music tags only relate to existing music tracks, i.e., tags are uneven or not available for new tracks. This so-called cold start issue has motivated research into a number of topics, such as automatic music tag annotation [3-7] and tag-based music retrieval [3-8] from an untagged music database.

This work was supported in part by the Taiwan e-Learning and Digital Archives Program (TELDAP) sponsored by the National Science Council of Taiwan under Grant: NSC 100-2631-H-001-013.

Most existing music tagging methods focus on *offline tag prediction* for music tag annotation and retrieval. In contrast, we propose an efficient generative approach that combines a Gaussian mixture model (GMM) and a mixture of tag-based aspect models (TBAMs) for *real-time tag prediction*. To synchronize the dynamic visual tag distributions with the music being played, tag prediction is performed on a fixed-length chunk of music with a small number of frames sliding at a down-sampled rate over the music being played. We utilize the GMM as an auditory feature reference to encode the frame vectors of an audio chunk in a fixed-dimensional feature vector. To online predict the tag distribution of a sliding chunk of an audio music track, the tag predictor must be very efficient. After the audio feature extraction and GMM-based auditory feature representation steps, the computation in TBAM-based online tag prediction is linear.

The remainder of this paper is organized as follows. Section 2 provides an overview of the proposed PWT music player. In Section 3, we describe the audio feature extraction and representation components as well as the synchronization schema. In Section 4, we introduce the TBAM-based music tag prediction. In Section 5, we discuss the simulations of online tag prediction and evaluate the performance on the MajorMiner dataset. Section 6 contains the conclusion and our future work.

2. SYSTEM OVERVIEW

The proposed PWT music player is implemented in two phases, the *model training phase* and the *music playing phase*, as shown in Figure 2. In the model training phase, each music clip in the tagged music database is extracted in a set of frame-based audio feature vectors; then a pre-trained GMM is applied to encode the frame vectors in a fixed-dimensional feature vector. Finally, the TBAMs are learned from the fixed-dimensional feature vectors and the associated tag labels. In the *music playing phase*, the music track to be played is extracted into frame vectors by a buffered audio feature extractor. Then, the frame vectors of a sliding chunk are encoded in a fixed-dimensional feature vector, and the GMM and TBAMs are used to generate the tag distribution of the chunk. A timer synchronizes the display of the tag distribution of each chunk and the audio playback. The tag distribution of a sliding chunk will be displayed at the end of the chunk, as shown in Figure 3. In other words, users can always see the tag distribution of the short chunk the music player has just played.

3. AUDIO FEATURE EXTRACTION AND PLAYBACK SYNCHRONIZATION

In this work, the audio feature extraction component and clip-level (or chunk-level) feature representation are modified from our previous work [8]. To satisfy the requirement for real-time processing, we only utilize the timbre features, i.e., Mel-Frequency Cepstral Coefficients (MFCCs), delta-MFCCs and delta-delta MFCCs, which can be extracted faster than the audio playback, to form a 39-dimensional

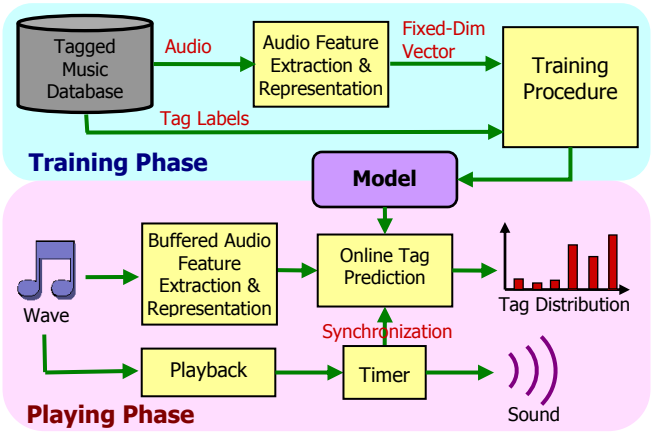


Figure 2. The flowchart of the proposed PWT music player.

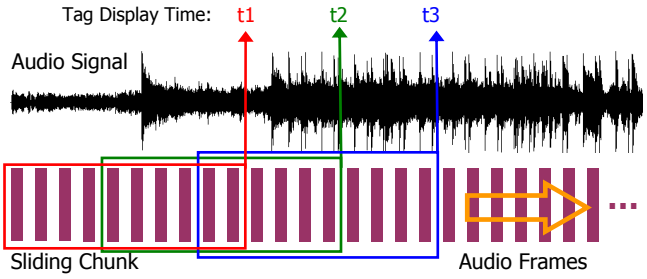


Figure 3. An illustration of the synchronization of the dynamic tag distribution display and music playback, where the tag distribution of the sliding chunk is displayed at the end of the chunk.

frame vector. Each frame is extracted with a window size d=0.05 seconds and a frame shifting rate h=0.5.

As illustrated in Figure 3, to synchronize the display of dynamic tag distributions with music playback, tag prediction is performed on a chunk comprised of T frames sliding at a down-sampled rate R over the audio playback. When the chunk hop size HS is set at 4 frames, the down-sampled rate $R = SR / \{1 / (HS \cdot h \cdot d)\}$ is 441, where SR is the sampling rate of the audio playback, which is 44100 Hz, i.e., the player updates the tag distributions every 0.01 seconds. The playing time (e.g., t1, t2 and t3) corresponds to the last (rightmost) frame of the chunk (i.e., the red, green, and blue ones). In this way, users always see the tag distribution of the short chunk the music player has just played.

To produce a standard auditory feature reference that can be used for any music signal, we apply a global GMM trained on a collection of randomly selected feature vectors to encode an audio frame. In the GMM, we define a set of "latent feature classes", z_k, k=1,…,K, each of which corresponds to the k-th Gaussian component, denoted as $N_k(\cdot)$, with mixture weight π_k, mean vector $\boldsymbol{\mu}_k$, and covariance matrix $\boldsymbol{\Sigma}_k$. The t-th frame vector $\mathbf{x}_t$ of a song is encoded by a set of posterior probabilities over z_k, k=1,…K as follows:

$$p(z_k \mid \mathbf{x}_t) = \frac{N_k(\mathbf{x}_t \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{h=1}^{K} N_h(\mathbf{x}_t \mid \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_h)}. \quad (1)$$

We do not consider the mixture weight π_k because it was not useful in our previous work [9]. A chunk of frames is the

basic unit for online tag prediction. Suppose that chunk s_n starts at time t_n of the playback song. Then, its chunk-level audio posterior representation $\boldsymbol{\theta}_n$ can be computed by

$$\theta_{nk} \leftarrow p(z_k \mid s_n) = \frac{1}{T}\sum_{t=t_n}^{t_n+T-1} p(z_k \mid \mathbf{x}_t), \qquad (2)$$

where θ_{nk} is the k-th component of $\boldsymbol{\theta}_n$. Since adjacent chunks have many frames that overlap, the change in the audio posterior representations (i.e., $\boldsymbol{\theta}$) of adjacent chunks will not be salient, and the resulting dynamic tag distributions will change smoothly. Note that, like Internet video streaming, a short buffer is necessary to guarantee that the display of dynamic tag distributions and music playback is synchronized, even though the music player's online tag prediction method is efficient. If the timer (see Figure 2) finds that tag prediction cannot synchronize with the playback, it instructs the playback function to pause until enough predicted tag distributions are buffered.

4. REAL-TIME TAG PREDICTION

To realize real-time tag prediction, we apply TBAM [8], a probabilistic generative model that jointly models the audio posterior distribution derived by Eq. (2) and tag labels with counts of each clip in a training music database.

Suppose the training music database contains J audio clips h_j, j=1,…,J, and the corresponding tag labels with counts $c(j,m)$, which is a non-negative integer indicating the number of times that tag w_m from a predefined tag set $\mathbf{w}=\{w_1,\ldots,w_M\}$ has been assigned to h_j. Since the real-time tag prediction is applied on the chunk-level audio posterior distribution, each training audio clip is uniformly divided into n_j chunks of T frames. We assume that the tag labels of a training audio clip can be shared by its component chunks[1]. Therefore, the training music database contains $N = \sum_j n_j$ training chunks s_n, n=1,…,N, and their corresponding tag labels with counts $c(n,m)$. Given a latent feature class z_k, we define its corresponding TBAM as a multinomial distribution with parameter $\boldsymbol{\beta}_k$, whose m-th component β_{km} corresponds to tag w_m and is subject to $\sum_m \beta_{km} = 1$. $\boldsymbol{\beta}_k$ can be learned by maximizing the following log-likelihood via the expectation-maximization algorithm:

$$\boldsymbol{L} = \sum_{n=1}^{N}\sum_{m=1}^{M} c(n,m)\log\sum_{k=1}^{K}\theta_{nk}\beta_{km}, \qquad (3)$$

where θ_{nk} is the k-th component of $\boldsymbol{\theta}_n$ from Eq. (2), representing the probability that the audio content of s_n is similar to z_k, the k-th Gaussian component in the pre-trained GMM.

In the playing phase, given an audio posterior distribution $\boldsymbol{\theta}$ of a sliding chunk, the affinity of tag w_m for the chunk is computed by the linear combination of mixture probabilities, each with a pre-learned model parameter β_{km}:

$$p(w_m \mid \boldsymbol{\theta}) = \sum_{k=1}^{K}\theta_k\beta_{km}. \qquad (4)$$

[1] The assumption is valid if the training clip is short, since the tag labels are in general consistent over a short clip. In this work, the length of a training clip is 10 seconds.

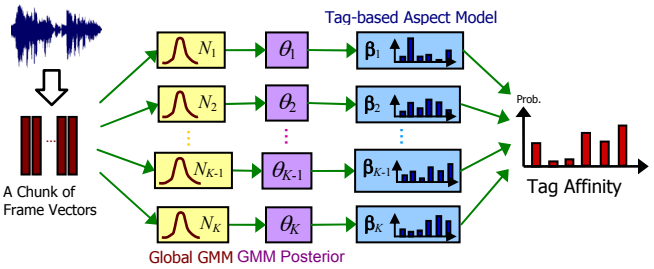


Figure 4. The prediction flow of tag affinity.

The intuition of tag prediction is illustrated in Figure 4. Suppose the human memory store K discrete timbre patterns described by a global GMM (i.e., N_k, k=1,…K), each of which is indexed by z_k and has a corresponding co-tag distribution $\boldsymbol{\beta}_k$. If a user has just listened to a chunk that can be described completely by N_1, (i.e., θ_1 =1, and θ_i=0 for all $i\neq1$), the tag affinities will exactly follow the distribution $\boldsymbol{\beta}_1$.

The computation in Eq. (4) is linear given $\boldsymbol{\theta}$. Therefore, the bottleneck of online tag prediction occurs when computing $\boldsymbol{\theta}$ in Eqs. (1) and (2), whose complexity is proportional to the number of latent feature classes K. Although reducing K could make the computation more efficient, it could also degrade the tag prediction performance dramatically because K corresponds to the resolution of the audio posterior representation. If the music player is implemented on a handheld device with less computing power, K (the major factor), HS and h should be chosen carefully.

5. EXPERIMENTS

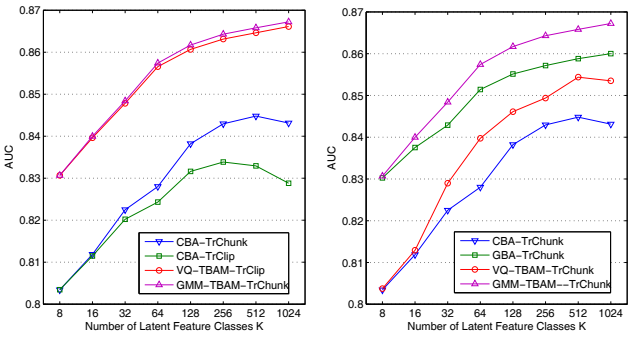
We evaluated the proposed system on the MajorMiner dataset [2]. The dataset is available from the MajorMiner website, which uses a game to gather informative free text labels for music. We downloaded all the music clips associated with the 45 most commonly used tags on the website. The resulting dataset contains 2,472 ten-second audio clips together with the tag labels. We perform three-fold cross-validation 10 times. In a set of randomly split three folds, 1,648 clips are used for training and 824 for testing. To simulate the real-time tagging scenario, each 10-second test clip is uniformly divided into non-overlapping 1-second (i.e., T =HS =40 frames) chunks. Since the test clips are short, the tag labels are generally consistent over a short clip, so each chunk inherits the tag labels of its source clip. In this paper, we only evaluate real-time tag prediction on short chunks. Currently the PWT music player is implemented with Matlab environment. We will demonstrate the PWT music player at the conference.

We consider two strategies for handling the training clips, namely, clip-level training (TrClip) and chunk-level training (TrChunk). In TrClip, we do not divide the training clips into chunks, i.e., the TBAM is trained on $\boldsymbol{\theta}_j$, j=1,…,1648. While, in TrChunk, like the test clips, each 10-second training clip is divided into 10 one-second chunks, and the TBAM is trained on $\boldsymbol{\theta}_n$, n=1,…,16480. To train the GMM, we randomly select 25% of the frame-based feature vectors (~235K frames) in the complete dataset. The GMM

and TBAM are trained with the stopping criterion that the objective function is increased by less than a ratio of 0.0001.

We compare our method with the Codeword Bernoulli Average (CBA) method [6], which uses a codebook to encode the frame-based feature vectors of a clip or a chunk. Our GMM-based TBAM method has two advantages over CBA. First, when both T and K are small, the GMM-based posterior representation provides a better model generalization than the vector quantization (VQ)-based histogram representation. Second, the employment of multinomial distributions in TBAM enables the modeling of tag co-occurrences and tag counts.

Figure 5 shows the average performance of 10-times three-fold cross-validation in terms of the area under the receiver operating characteristic (ROC) curve per chunk (AUC). In Figure 5(a), we compare chunk-level training (TrChunk) with clip-level training (TrClip). Clearly, TrChunk outperforms TrClip under different K (the number of mixture components in GMM-TBAM or the number of codewords in CBA). For the GMM-TBAM method, there is no significant difference between the two training strategies (GMM-TBAM-TrChunk vs. GMM-TBAM-TrClip). In contrast, for the CBA method, clip-level training yields a poor performance when K is large (CBA-TrChunk vs. CBA-TrClip). This could be because the VQ-based histogram representation has insufficient modeling ability for a small chunk. Figure 5(b) shows the performance of different methods using chunk-level training. Here, we compare two additional methods, namely, the Gaussian Bernoulli average (GBA) and the VQ-histogram-based TBAM (VQ-TBAM). GBA is modified from CBA by replacing the VQ-based histogram with a GMM-based posterior representation; while VQ-TBAM uses the VQ-based histogram to represent a chunk, instead of the GMM-based posterior to encode a chunk. We observe that, under different K, the proposed GMM-TBAM method outperforms VQ-TBAM, CBA and GBA; and GBA outperforms CBA. The results demonstrate that GMM outperforms VQ in encoding a short chunk (1 second in this paper) and the tag-based aspect model can generalize tag modeling better than a set of independent Bernoulli models. The GMM-TBAM method is efficient enough to be applied in the PWT music player. When K=32, it outperforms CBA with K=512 and also runs 16 times faster than CBA with K=512. The performance improves as K increases, but it tends to saturate if K becomes too large. Considering the efficiency of the real-time tagging system, we recommend using K=64, which reached almost 0.86 in AUC according the slope of performance increase. However, the selection of K depends on the computing power of the device. The best AUC performance tested on 10-second clips of the MajorMiner dataset in the tag classification task at MIREX 2010 was 0.8828 [10]. Our GMM-TBAM system, which applies a very simple and efficient model with basic timbre features (MFCCs) to predict tags for a short audio chunk (1 second), yields a comparable performance (0.8675 when K=1024). Note that our MajorMiner dataset was collected during March 2011, so it might be slightly different from the one used at MIREX 2010.



(a) **(b)**

Figure 5. Tag prediction performance: (a) comparison between CBA and TBAM using chunk-level training (TrChunk) and clip-level training (TrClip); (b) comparison of the four methods using chunk-level training (TrChunk).

6. CONCLUSION AND FUTURE WORK

In this paper, we have presented the PWT music player, which uses an online music tagger to visualize dynamic tag distributions with music playback in real-time. Our experimental results demonstrate the potential and effectiveness of the real-time auto-tagging method. In our future work, we will conduct a subjective evaluation of the PWT player's usability. Moreover, to improve the real-time music tagging algorithm, we will develop other visualization mechanisms, such as an active tag cloud layout or a 3D-based display, to present dynamic tag distributions.

7. REFERENCES

[1] P. Lamere, "Social tagging and music information retrieval," *JNMR*, 37(2), pp. 101-114, 2008.

[2] M. Mandel and D. Ellis: "A web-based game for collecting music metadata," *JNMR*, 37(2), pp. 151–165, 2008.

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

[4] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation", *NIPS*, 2007.

[5] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE TASLP*, 16(2), pp. 467–476, 2008.

[6] M. Hoffman, D. Blei, and P. Cook, "Easy as CBA: a simple probabilistic model for tagging music," *ISMIR*, 2009.

[7] H.-Y. Lo, J.-C. Wang, H.-M. Wang and S.-D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE TMM*, 13(3), pp. 518-529, 2011.

[8] J.-C. Wang, M.-S. Wu, H.-M. Wang and S.-K. Jeng, "Query by multi-tags with multi-level preferences for content-based music retrieval," *IEEE ICME*, 2011.

[9] J.-C. Wang, H.-S. Lee, H.-M. Wang and S.-K. Jeng, "Learning the similarity of audio music in bag-of-frames representation from tagged music data," *ISMIR*, 2011.

[10] http://nema.lis.illinois.edu/nema_out/mirex2010/results/atg/subtask1_report/aff/