appears in International Society for Music Information Retrieval (ISMIR), Utrecht, 2010

AUTOMATIC MUSIC TAGGING WITH TIME SERIES MODELS

Emanuele Coviello
University of California, San Diego
Dept. of Electrical and Computer Engineering
ecoviell@ucsd.edu

Luke Barrington
University of California, San Diego
Dept. of Electrical and Computer Engineering
lukeinusa@gmail.com

Antoni B. Chan
City University of Hong Kong
Dept. of Computer Science
abchan@cityu.edu.hk

Gert. R. G. Lanckriet
University of California, San Diego
Dept. of Electrical and Computer Engineering
gert@ece.ucsd.edu

ABSTRACT

State-of-the-art systems for automatic music tagging model music based on bag-of-feature representations which give little or no account of temporal dynamics, a key characteristic of the audio signal. We describe a novel approach to automatic music annotation and retrieval that captures temporal (e.g., rhythmical) aspects as well as timbral content. The proposed approach leverages a recently proposed song model that is based on a generative time series model of the musical content — the dynamic texture mixture (DTM) model — that treats fragments of audio as the output of a linear dynamical system. To model characteristic temporal dynamics and timbral content at the tag level, a novel, efficient hierarchical EM algorithm for DTM (HEM-DTM) is used to summarize the common information shared by DTMs modeling individual songs associated with a tag. Experiments show learning the semantics of music benefits from modeling temporal dynamics.

1. INTRODUCTION

This paper concerns *automatic tagging* of music with descriptive keywords (e.g., genres, emotions, instruments, usages, etc.), based on the content of the song. Music annotations can be used for a variety of purposes, such as searching for songs exhibiting specific qualities (e.g., "jazz songs with female vocals and saxophone"), or retrieval of semantically similar songs (e.g., generating playlists based on songs with similar annotations).

State-of-the-art music "auto-taggers" model a song as a "bag of audio features" [7, 9–11, 14]. The bag of features representation extracts audio features from the song at a regular time interval, but then treats these features independently, ignoring the temporal order or dynamics between them. Hence, this representation fails to account for the longer term musical dynamics (e.g. tempo and beat) or temporal structures (e.g. riffs and arpeggios), which are clearly important characteristics of a musical signal.

We address this limitation by adopting the dynamic texture (DT) model [6], a generative, *time-series model* of musical content that captures longer-term time dependencies. The DT model is similar to the Hidden Markov model (HMM) which has proven robust in music identification [12]. The difference is that HMMs require to quantize the audio signal into a fixed number of discrete "phonemes", while the DT has a continuous state space that is a more flexible model for music.

Musical time series often show significant structural changes within a single song and have dynamics that are only locally homogeneous. Hence, [1] proposes to model the audio fragments from a single song as a dynamic texture mixture (DTM) model [3], for the task of automatic music *segmentation*. These results demonstrated that the DTM provides an accurate segmentation of music into homogeneous, perceptually similar segments (corresponding to what a human listener would label as 'chorus', 'verse', 'bridge', etc.) by capturing *temporal* as well as *textural* aspects of the musical signal.

In this paper, we adopt the DTM model to propose a novel approach to the task of automatic music *annotation* that accounts for both the timbral content and the temporal dynamics that are predictive of a semantic tag. We first model all songs in a music database as DTMs, capturing longer-term time dependencies and instantaneous spectral content at the *song-level*. Second, the characteristic temporal and timbral aspects of musical content that are commonly associated with a semantic tag are identified by learning a *tag-level* DTM that summarizes the common features of a (potentially large) set of song-level DTMs for the tag. Given all song-level DTMs associated with a particular tag, the common information is summarized by clustering similar song-level DTs using a novel, efficient hierarchical EM (HEM-DTM) algorithm. This gives rise to a tag-level DTM with few mixture components (as opposed to tag-level Gaussian mixture models in [14], which do not capture temporal dynamics). Experimental results show that the proposed time-series model improves annotation and retrieval, in particular for tags with temporal dynamics that unfold in the time span of a few seconds.

The remainder of this paper is organized as follows. In Section 2, we present the annotation and retrieval system using time-series data, while in in Section 3, we present an efficient hierarchical EM algorithm for dynamic texture mixtures. Finally, in Sections 4 and 5, we present experiments using DTM for music annotation and retrieval.

2. ANNOTATION AND RETRIEVAL

In this section we formulate the tasks of annotation and retrieval of audio data as a semantic multi-class labeling (SML) problem [2] in the context of time-series models.

2.1 Notation

A song s is represented as a collection of T overlapping time series $\mathcal{Y} = \{y^1_{1:\tau}, \ldots, y^T_{1:\tau}\}$, where each $y^t_{1:\tau}$ represents τ sequential audio feature vectors extracted by passing a short-time window over the audio signal (also called an audio *fragment*). The number of fragments, T, depends on the length of the song. The semantic content of a song with respect to a vocabulary $\mathcal{V}$ of size $|\mathcal{V}|$ is represented in an annotation vector $\mathbf{c} = [c_1, \ldots, c_{|\mathcal{V}|}]$, where $c_k > 0$ only if there is a positive association between the song and the word w_k, otherwise $c_k = 0$. Each *semantic weight*, c_k, represents the degree of association between the song and word w_k. The data set $\mathcal{D}$ is a collection of $|\mathcal{D}|$ song-annotation pairs $(\mathcal{Y}_d, \mathbf{c}_d)$.

2.2 Music Annotation

We treat annotation as a semantic multi-class problem [2, 14] in which each class is a word w, from a vocabulary $\mathcal{V}$ of unique tags (e.g., "bass guitar", "hip hop", "boring"). Each word w_k is modeled with a probability distribution over the space of audio fragments, $p(y^t_{1:\tau}|w_k)$. The annotation task is to find the subset $\mathcal{W} = \{w_1, \ldots, w_{\mathcal{A}}\} \subseteq \mathcal{V}$ of $\mathcal{A}$ words that best describe a novel song $\mathcal{Y}$.

Given the audio fragments of a novel song $\mathcal{Y}$, the most relevant words are the ones with highest posterior probability, computed using Bayes' rule:

$$p(w_k|\mathcal{Y}) = \frac{p(\mathcal{Y}|w_k)\, p(w_k)}{p(\mathcal{Y})}, \tag{1}$$

where $p(w_k)$ is the prior of the k^{th} word, and $p(\mathcal{Y}) = \sum_{k=1}^{|\mathcal{V}|} p(\mathcal{Y}|w_k)p(w_k)$ is the song prior. To promote annotation using a diverse set of words, we assume an uniform prior, $p(w_k) = 1/|\mathcal{V}|$. We follow [14] in estimating the likelihood term in (1) with the geometric average of the individual sequence likelihoods, $p(\mathcal{Y}|w_k) = \prod_{t=1}^{T} \left(p(y^t_{1:\tau}|w_k)\right)^{\frac{1}{T}}$. Note that, unlike bag-of-features models that discard any dependency between audio features vectors (each describing milliseconds of audio), we only assume independence between different *sequences* of audio feature vectors (describing seconds of audio). Correlations within a single sequence are accounted for by the model presented in Section 3.

The probability that the song $\mathcal{Y}$ can be described by word w_k is

$$p_k = p(w_k|\mathcal{Y}) = \frac{\prod_{t=1}^{T} \left(p(y^t_{1:\tau}|w_k)\right)^{\frac{1}{T}}}{\sum_{l=1}^{|\mathcal{V}|} \prod_{t=1}^{T} \left(p(y^t_{1:\tau}|w_l)\right)^{\frac{1}{T}}}. \tag{2}$$

Finally, the song can be represented as a semantic multinomial, $\mathbf{p} = [p_1, \ldots, p_{|\mathcal{V}|}]$, where each $p_k = p(w_k|\mathcal{Y})$ represents the relevance of the k^{th} word for the song, and $\sum_{i=1}^{|\mathcal{V}|} p_i = 1$. We annotate a song with the most likely tags according to $\mathbf{p}$, i.e., we select the tags with the words with the largest probability.

2.3 Music Retrieval

Given a query word, songs in the database can be retrieved based on their relevance to the semantic query word [1]. In particular, the song's relevance to the query word w_k is equivalent to the posterior probability of the word, $p(w_k|\mathcal{Y})$ in (2). Hence, retrieval involves rank-ordering the songs in the database based on the k-th entry (p_k) of the semantic multinomials $\mathbf{p}$.

2.4 Learning DTM tag models

In this paper, we model the tag-level distributions, $p(y^t_{1:\tau}|w_k)$, as dynamic texture mixture models. The tag-level distributions are estimated from the set of training songs associated with the particular tag. One approach is to extract all the audio fragments from the relevant training songs, and then run the EM algorithm [3] directly on this data to learn the tag-level DTM. This approach, however, requires storing large amounts of audio fragments in memory (RAM) for running the EM algorithm. For even modest-sized databases, the memory requirements can exceed the RAM capacity of most computers.

To allow efficient training in both computation time and memory requirements, we propose to break the learning procedure into two steps. First, a DTM is estimated for each song using the standard EM algorithm [3]. Next, each tag-level model is estimated using the hierarchical EM algorithm on all the song-level DTMs associated with the particular tag. Because the database is first processed at the song-level, the computation can be easily done in parallel (over the songs) and the memory requirement is greatly reduced to that of processing a single song. The memory requirements for computing the tag-level models is also reduced, since each song is succinctly modeled by the parameters of a DTM.

Such a reduction in computational complexity also ensures that the tag-level models can be learned from cheaper, weakly-labeled data (i.e., missing labels, labels without segmentation data, etc.) by pooling over large amounts of audio data to amplify the appropriate attributes. In summary, adopting DTM, or time-series models in general, as a tag-model for SML annotation requires an appropriate HEM algorithm for efficiently learning the tag-level models from the song-level models. In the next section, we review the DTM and present the HEM algorithm for DTM.

3. HIERARCHICAL EM FOR DTMS

In this section, we first review the dynamic texture (DT) and dynamic texture mixture (DTM) models for modeling musical time-series. We then present the hierarchical EM algorithm for efficiently learning a tag-level DTM from a set of song-level DTMs.

3.1 The Dynamic Texture Model

A dynamic texture [6] (DT) is a generative model that takes into account both the acoustics and the dynamics of audio sequences [1]. The model consists of two random variables, y_t, which encodes the acoustic component (audio

[1] Note that although this work focuses on single-word queries, our representation easily extends to multiple-word queries [13].

feature vector) at time t, and x_t, which encodes the dynamics (evolution) of the acoustic component over time. The two variables are modeled as a *linear dynamical system*,

$$x_t = Ax_{t-1} + v_t, \quad (3)$$

$$y_t = Cx_t + w_t + \bar{y}, \quad (4)$$

where $x_t \in \mathbb{R}^n$ and $y_t \in \mathbb{R}^m$ are real vectors (typically $n \ll m$). Using such a model, we assume that the dynamics of the audio can be summarized by a more parsimonious ($n < m$) *hidden state process* x_t, which evolves as a first order Gauss-Markov process, and each *observation variable* y_t, which encodes the acoustical component (audio feature vector at time t) is dependent only on the current hidden state x_t.

The matrix $A \in \mathbb{R}^{n\times n}$ is a *state transition matrix*, which encodes the dynamics or evolution of the hidden state variable (e.g., the evolution of the audio track), and the matrix $C \in \mathbb{R}^{m\times n}$ is an *observation matrix*, which encodes the basis functions for representing the audio sequence. The vector $\bar{y} \in \mathbb{R}^n$ is the mean of the dynamic texture (i.e. the mean audio feature vector). v_t is a *driving noise process*, and is zero-mean Gaussian distributed , e.g., $v_t \sim \mathcal{N}(0, Q)$, where $Q \in \mathbb{R}^{n\times n}$ is a covariance matrix. w_t is the *observation noise* and is also zero-mean Gaussian, e.g., $w_t \sim \mathcal{N}(0, R)$, where $R \in \mathbb{R}^{m\times m}$ is a covariance matrix. Finally, the *initial condition* is specified as $x_1 \sim \mathcal{N}(\mu, S)$, where $\mu \in \mathbb{R}^n$ is the mean of the initial state, and $S \in \mathbb{R}^{n\times n}$ is the covariance. The dynamic texture is specified by the parameters $\Theta = \{A, Q, C, R, \mu, S, \bar{y}\}$.

Intuitively, the columns of C can be interpreted as the principal components (or basis functions) of the audio features vectors over time. Hence, each audio feature vector y_t can be represented as a linear combination of principal components, with corresponding weights given by the current hidden state x_t. In this way, the DT can be interpreted as a time-varying PCA representation of an audio feature vector time-series.

3.2 The Dynamic Texture Mixture Model

A song is a combination of heterogeneous sequences with significant structural variations, and hence is not well represented as a single DT model. To address this lack of global homogeneity, [1] proposed to represent audio fragments, extracted from a song, as samples from a dynamic texture mixture (DTM) [3], effectively modeling local structure of the song. The DTM model [3] introduces an assignment random variable $z \sim \text{multinomial}(\pi_1, \cdots, \pi_K)$, which selects one of the K dynamic texture components as the source of the audio fragment. Each mixture component is parameterized by $\Theta_z = \{A_z, C_z, Q_z, R_z, \mu_z, S_z, \bar{y}_z\}$, and the DTM model is parameterized by $\Theta = \{\pi_z, \Theta_z\}_{z=1}^K$.

Given a set of audio samples, the maximum-likelihood parameters of the DTM can be estimated with recourse to the expectation-maximization (EM) algorithm [3], which is an iterative optimization method that alternates between estimating the hidden variables with the current parameters, and computing new parameters given the estimated hidden variables (the "complete data"). The EM algorithm for DTM alternates between estimating second-order statistics of the hidden-states, conditioned on each audio sequence, with the Kalman smoothing filter (E-step), and computing new parameters given these statistics (M-step).

Previous work in [1] has successfully used the DTM for the task of segmenting the structure of a song into acoustically similar sections (e.g., intro, verse, chorus, bridge, solo, outro). In this work, we demonstrate that the DTM can also be used as a tag-level annotation model for music annotation and retrieval. We next present a hierarchical EM algorithm for efficiently estimating these tag-level DTMs from large sets of song-level DTMs, previously estimated for the set of training songs associated with a tag.

3.3 Hierarchical EM for learning DTM hierarchies

Given a DTM model of each training song as learned in the previous section, the goal now is to learn a tag-level DTM model that summarizes the common features of the corresponding song-level DTMs. First, all song-level DTMs with a particular tag are pooled together into a single, large DTM. Next, the common information is summarized by clustering similar DT components together, forming a new *tag-level* DTM with fewer mixture components.

The DT components are clustered using the hierarchical expectation-maximization (HEM) algorithm [15]. At a high level, this is done by generating *virtual samples* from each of the song-level component models, merging all the samples, and then running the standard EM algorithm on the merged samples to form the reduced tag-level mixture. Mathematically, however, using the virtual samples is equivalent to marginalizing over the distribution of song-level models. Hence, the tag model can be learned directly and efficiently from the parameters of the song-level models, without generating any virtual samples.

The HEM algorithm was originally proposed in [15] to reduce a Gaussian mixture model (GMM) with many components to a representative GMM with fewer components and has been successful in learning GMMs from large datasets for the annotation and retrieval of images [2] and music [14]. We next present an HEM algorithm for mixtures with components that are *dynamic textures* [4].

3.3.1 HEM Formulation

Formally, let $\Theta^{(s)} = \{\pi_i^{(s)}, \Theta_i^{(s)}\}_{i=1}^{K^{(s)}}$ denote the combined song-level DTM with $K^{(s)}$ components, where $\Theta_i^{(s)}$ are the parameters for the i^{th} DT component. The likelihood of observing an audio sequence $y_{1:\tau}$ with length τ from the combined song-level DTM $\Theta^{(s)}$ is given by

$$p(y_{1:\tau}|\Theta^{(s)}) = \sum_{i=1}^{K^{(s)}} \pi_i^{(s)} p(y_{1:\tau}|z^{(s)} = i, \Theta^{(s)}), \quad (5)$$

where $z \sim \text{multinomial}(\pi_1^{(s)}, \cdots \pi_{K^{(s)}}^{(s)})$ is the hidden variable that indexes the mixture components. $p(y_{1:\tau}|z^{(s)} = i, \Theta^{(s)})$ is the likelihood of the audio $y_{1:\tau}$ under the i^{th} DT mixture component, and $\pi_i^{(s)}$ is the prior weight for the i^{th} component. The goal is to find a tag-level annotation DTM, $\Theta^{(a)} = \{\pi_j^{(a)}, \Theta_j^{(a)}\}_{j=1}^{K^{(a)}}$, which

represents (5) using fewer number of mixture components, $K^{(a)}$, (i.e., $K^{(a)} < K^{(s)}$). The likelihood of observing an audio sequence $y_{1:\tau}$ from the tag-level DTM $\Theta^{(a)}$ is

$$p(y_{1:\tau}|\Theta^{(a)}) = \sum_{j=1}^{K^{(a)}} \pi_j^{(a)} p(y_{1:\tau}|z^{(a)} = j, \Theta^{(a)}), \qquad (6)$$

where $z^{(a)} \sim \text{multinomial}(\pi_1^{(a)}, \cdots, \pi_{K^{(a)}}^{(a)})$ is the hidden variable for indexing components in $\Theta^{(a)}$. Note that we will always use i and j to index the components of the song-level model, $\Theta^{(s)}$, and the tag-level model, $\Theta^{(a)}$, respectively. To reduce clutter, we will also use the shorthand $\Theta_i^{(s)}$ and $\Theta_j^{(a)}$ to denote the i^{th} component of $\Theta^{(s)}$ and the j^{th} component of $\Theta^{(a)}$, respectively. For example, we denote $p(y_{1:\tau}|z^{(s)} = i, \Theta^{(s)}) = p(y_{1:\tau}|\Theta_i^{(s)})$.

3.3.2 Parameter estimation

To obtain the tag-level model, HEM [15] considers a set of N virtual observations drawn from the song-level model $\Theta^{(s)}$, such that $N_i = N\pi_i^{(s)}$ samples are drawn from the i^{th} component. We denote the set of N_i virtual audio samples for the i^{th} component as $Y_i = \{y_{1:\tau}^{(i,m)}\}_{m=1}^{N_i}$, where $y_{1:\tau}^{(i,m)} \sim \Theta_i^{(s)}$ is a single audio sample and τ is the length of the virtual audio (a parameter we can choose). The entire set of N samples is denoted as $Y = \{Y_i\}_{i=1}^{K^{(s)}}$. To obtain a consistent hierarchical clustering, we also assume that all the samples in a set Y_i are eventually assigned to the same tag-level component $\Theta_j^{(a)}$. The parameters of the tag-level model can then be estimated by maximizing the likelihood of the virtual audio samples,

$$\Theta^{(a)^*} = \arg\max_{\Theta^{(a)}} \log p(Y|\Theta^{(a)}), \qquad (7)$$

where

$$\log p(Y|\Theta^{(a)}) = \log \prod_{i=1}^{K^{(s)}} p(Y_i|\Theta^{(a)}) \qquad (8)$$

$$= \log \prod_{i=1}^{K^{(s)}} \sum_{j=1}^{K^{(a)}} \pi_j^{(a)} \int p(Y_i, X_i|\Theta_j^{(a)}) dX_i \qquad (9)$$

and $X_i = \{x_{1:\tau}^{(i,m)}\}$ are the hidden-state variables corresponding to Y_i. Computing the log-likelihood in (9) requires marginalizing over the hidden assignment variables $z_i^{(a)}$ and hidden state variables X_i. Hence, (7) can also be solved with recourse to the EM algorithm [5]. In particular, each iteration consists of

$$\text{E-Step: } \mathcal{Q}(\Theta^{(a)}, \hat{\Theta}^{(a)}) = \mathbb{E}_{X,Z|Y,\hat{\Theta}^{(a)}}[\log p(X, Y, Z|\Theta^{(a)})]$$

$$\text{M-Step: } \Theta^{(a)^*} = \arg\max_{\Theta^{(a)}} \mathcal{Q}(\Theta^{(a)}, \hat{\Theta}^{(a)})$$

where $\hat{\Theta}^{(a)}$ is the current estimate of the tag-level model, $p(X, Y, Z|\Theta^{(a)})$ is the "complete-data" likelihood, and $\mathbb{E}_{X,Z|Y,\hat{\Theta}^{(a)}}$ is the conditional expectation with respect to the current model parameters.

As is common with the EM formulation, we introduce a hidden assignment variable $\mathbf{z}_{i,j}$, which is an indicator variable for when the audio sample set Y_i is assigned to the j^{th} component of $\Theta^{(a)}$, e.g., when $z_i^{(a)} = j$. The complete-data log-likelihood is then

$$\log p(X, Y, Z|\Theta^{(a)}) \qquad (10)$$

$$= \sum_{i=1}^{K^{(s)}} \sum_{j=1}^{K^{(a)}} \mathbf{z}_{i,j} \log \pi_j^{(a)} + \mathbf{z}_{i,j} \log p(Y_i, X_i|\Theta_j^{(a)}).$$

The $\mathcal{Q}$ function is then obtained by taking the conditional expectation of (10), and using the law of large numbers to remove the dependency on the virtual samples. The result is a $\mathcal{Q}$ function that depends only on the parameters of the song-level DTs $\Theta_i^{(s)}$.

The HEM algorithm for DTM is summarized in Algorithm 1. In the E-step, the expectations in Eq. (11) are computed for each song-level DT $\Theta_i^{(s)}$ and current tag-level DT $\hat{\Theta}_j^{(a)}$. These expectations can be computed using "suboptimal filter analysis" or "sensitivity analysis" [8] on the Kalman smoothing filter (see [4]). Next, the probability of assigning the song-level DT $\Theta_i^{(s)}$ to the tag-level DT $\hat{\Theta}_j^{(a)}$ is computed according to (12), and the expectations are aggregated over all the song-level DTs in (14). In the M-step, the parameters for each tag-level component $\hat{\Theta}_j^{(a)}$ are recomputed according to the update equations in (15). More details are available in [4].

4. MUSIC DATA

In this section we describe the music collection and the audio features used in our experiments.

The CAL500 [14] dataset consists of 502 Western popular songs from the last 50 years from 502 different artists. Each song has been annotated by at least 3 humans, using a semantic vocabulary of 174 words that includes genres, instruments, vocal characteristics, emotions, acoustic characteristics, and song usages. CAL500 provides hard binary annotations, which are 1 when a tag applies to the song and 0 when the tag does not apply. We find empirically that accurately fitting the HEM-DTM model requires a significant number of training examples so we restrict our attention to the 78 tags with at least 50 examples.

A popular feature for content-based music analysis, Mel-frequency cepstral coefficients (MFCCs) concisely summarize the short-time content of an acoustic waveform by using the discrete cosine transform (DCT) to decorrelate the bins of a Mel-frequency spectral histogram [2]. In Section 3.1 we noted how the DT model can be viewed as a time varying PCA representation of the audio features. This idea suggests that we can represent the spectrum over time as the output of the DT model y_t. In this case, the columns of the observation matrix C (PCA matrix) are analogous to the DCT basis functions, and the hidden states x_t are the coefficients (analogous to the MFCCs). The advantage with this formulation is that a different C matrix, i.e., basis functions, can be learned to best represent the particular song or semantic concept of interest.

[2] This decorrelation is usually convenient in that it reduces the number of parameters to be estimated.

Algorithm 1 HEM algorithm for DTM

Input: combined song-level DTM $\{\Theta_i^{(s)}, \pi_i^{(s)}\}_{i=1}^{K^{(s)}}$, number of virtual samples N.

Initialize tag-level DTM, $\{\hat{\Theta}_j^{(a)}, \pi_j^{(a)}\}_{j=1}^{K^{(a)}}$.

repeat

{E-step}

Compute expectations using sensitivity analysis for each $\Theta_i^{(s)}$ and $\hat{\Theta}_j^{(a)}$ (see [4]):

$$\begin{aligned}
\hat{x}_{t|j}^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[\mathbb{E}_{x|y,\hat{\Theta}_j^{(a)}}[x_t]\right], \\
\hat{P}_{t,t|j}^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[\mathbb{E}_{x|y,\hat{\Theta}_j^{(a)}}[x_t x_t^T]\right], \\
\hat{P}_{t,t-1|j}^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[\mathbb{E}_{x|y,\hat{\Theta}_j^{(a)}}[x_t x_{t-1}^T]\right], \\
\hat{W}_{t|j}^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[(y_t-\bar{y}_j)\mathbb{E}_{x|y,\hat{\Theta}_j^{(a)}}[x_t]^T\right], \\
\hat{U}_{t|j}^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[(y_t-\bar{y}_j)(y_t-\bar{y}_j)^T\right], \\
\hat{u}_t^{(i)} &= \mathbb{E}_{y|\Theta_i^{(s)}}\left[y_t\right], \\
\ell_{i|j} &= \mathbb{E}_{\Theta_i^{(s)}}[\log p(y_{1:\tau}|\hat{\Theta}_j^{(a)})].
\end{aligned} \tag{11}$$

Compute assignment probability and weighting:

$$\hat{\mathbf{z}}_{i,j} = \frac{\pi_j^{(a)} \exp\left(N_i \ell_{i|j}\right)}{\sum_{j'=1}^{K^{(a)}} \pi_{j'}^{(a)} \exp\left(N_i \ell_{i|j'}\right)} \tag{12}$$

$$\hat{\mathbf{w}}_{i,j} = \hat{\mathbf{z}}_{i,j} N_i = \hat{\mathbf{z}}_{i,j} \pi_i^{(s)} N \tag{13}$$

Computed aggregate expectations for each $\hat{\Theta}_j^{(a)}$:

$$\begin{aligned}
\hat{N}_j &= \textstyle\sum_i \hat{\mathbf{z}}_{i,j}, & \eta_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \hat{P}_{1,1|j}^{(i)}, \\
\hat{M}_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j}, & \gamma_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=1}^{\tau} \hat{u}_t^{(i)}, \\
\xi_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \hat{x}_{1|j}^{(i)}, & \beta_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=1}^{\tau} \hat{x}_{t|j}^{(i)}, \\
\Phi_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=1}^{\tau} \hat{P}_{t,t|j}^{(i)}, & & \\
\Psi_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t,t-1|j}^{(i)}, & & \\
\varphi_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t,t|j}^{(i)}, & & \\
\phi_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=2}^{\tau} \hat{P}_{t-1,t-1|j}^{(i)}, & & \\
\Lambda_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=1}^{\tau} \hat{U}_{t|j}^{(i)}, & & \\
\Gamma_j &= \textstyle\sum_i \hat{\mathbf{w}}_{i,j} \sum_{t=1}^{\tau} \hat{W}_{t|j}^{(i)}. & &
\end{aligned} \tag{14}$$

{M-step}

Recompute parameters for each component $\hat{\Theta}_j^{(a)}$:

$$\begin{aligned}
C_j^* &= \Gamma_j \Phi_j^{-1}, & R_j^* &= \tfrac{1}{\tau \hat{M}_j}(\Lambda_j - C_j^* \Gamma_j), \\
A_j^* &= \Psi_j \phi_j^{-1}, & Q_j^* &= \tfrac{1}{(\tau-1)\hat{M}_j}(\varphi_j - A_j^* \Psi_j^T), \\
\mu_j^* &= \tfrac{1}{\hat{M}_j}\xi_j, & S_j^* &= \tfrac{1}{\hat{M}_j}\eta_j - \mu_j^*(\mu_j^*)^T, \\
\pi_j^* &= \tfrac{\sum_{i=1}^{K^{(s)}} \hat{\mathbf{z}}_{i,j}}{K^{(s)}}, & \bar{y}_j^* &= \tfrac{1}{\tau \hat{M}_j}(\gamma_j - C_j^* \beta_j).
\end{aligned} \tag{15}$$

until convergence

Output: tag-level DTM $\{\Theta_j^{(a)}, \pi_j^{(a)}\}_{j=1}^{K^{(a)}}$.

Furthermore, since we explicitly model the temporal evolution of the spectrum, we do not need to include the instantaneous deltas of the MFCCs.

Our experiments use 34 Mel-frequency bins, computed from half-overlapping, 46ms audio segments. Each audio fragment is described by a time series $y_{1:\tau}^t$ of $\tau = 450$ sequential audio feature vectors, which corresponds to 10 seconds. Song-level DTM models are learned from a dense sampling of audio fragments of 10 seconds, extracted every 1 second.

Model	P	R	F-score	AROC	MAP	P10
HEM-GMM	0.49	0.23	0.26	0.66	0.45	0.47
CBA	0.41	0.24	0.29	0.69	0.47	0.49
HEM-DTM	0.47	0.25	0.30	0.69	0.48	0.53

Table 1. Annotation and retrieval results for HEM-DTM and HEM-GMM.

5. EVALUATION

Song-level DTMs were learned with $K = 16$ components and state-space dimension $n = 7$, using EM-DTM. Tag-level DTMs were learned by pooling together all song models associated with a given tag and reducing the result to a DTM with $K^{(r)} = 2$ components with HEM-DTM. To reduce the effect of low likelihoods in high dimensions, we normalize the single-segment likelihood terms, e.g., $p(y_{1:\tau}^t|w_k)$, by the length of the sequence τ.

To investigate the advantage of the DTM's temporal representation, we compare the auto-tagging performance of our model (HEM-DTM) to the hierarchically trained Gaussian mixture models (HEM-GMM) from [14], a generative model that ignores temporal dynamics. A comparison to the CBA model of [9] is provided as well. We follow the procedure of [14] for training HEM-GMMs, and our CBA implementation follows [9], with the modification that the codebook is constructed using only songs from the training set. All reported metrics are the results of 5-fold cross validation where each song appeared in the test set exactly once.

5.1 Annotation and Retrieval

Annotation performance is measured following the procedure in [14]. Test set songs are annotated with the 10 most likely tags in their semantic multinomial (Eq. 2). Annotation accuracy is reported by computing precision, recall and F-score for each tag, and then averaging over all tags. For a detailed definition of the metrics, see [14].

To evaluate retrieval performance, we rank-order test songs for each single-tag query in our vocabulary, as described in Section 2. We report mean average precision (MAP), area under the receiver operating characteristic curve (AROC) and top-10 precision (P10), averaged over all the query tags. The ROC curve is a plot of true positive rate versus false positive rate as we move down the ranked list. Random guessing would result in an AROC of 0.5. The top-10 precision is the fraction true positives in the top-10 of the ranking. MAP averages the precision at each point in the ranking where a song is correctly retrieved.

5.2 Results

Annotation and retrieval results are presented in Table 1, demonstrating superior performance for HEM-DTM, compared to HEM-GMM,for all metrics except for precision. This indicates that HEM-DTM is slightly more aggressive when annotating songs, but still its annotations are more accurate, as evidenced by the higher F-score. HEM-DTM performs better than CBA in all metrics. For retrieval, although AROC scores are comparable for CBA and HEM-DTM, HEM-DTM clearly improves the top of the ranked list more, as evidenced by the higher precision-at-10 score.

Tag	HEM-DTM		HEM-GMM	
	F-score	MAP	F-score	MAP
HEM-DTM better than HEM-GMM				
male lead vocals	0.44	0.87	0.08	0.81
female lead vocals	0.58	0.69	0.42	0.44
fast rhythm	0.40	0.48	0.20	0.42
classic rock	0.41	0.37	0.18	0.36
acoustic guitar	0.44	0.43	0.31	0.44
electric guitar	0.32	0.35	0.14	0.34
HEM-GMM better than HEM-DTM				
mellow	0.34	0.41	0.37	0.49
slow rhythm	0.45	0.60	0.44	0.62
weak	0.22	0.26	0.26	0.25
light beat	0.36	0.58	0.53	0.61
sad	0.13	0.23	0.28	0.30
negative feelings	0.27	0.33	0.35	0.36

Table 2. Annotation and retrieval results for some tags with HEM-DTM and HEM-GMM.

HEM-DTM	**classic rock**, **driving**, **energy**, **fast**, **male lead vocals**, **electric guitar**, **electric**, indifferent, powerful, rough
HEM-GMM	boring, major, acoustic, **driving**, not likeable, female lead vocals, **recording quality**, cold, **synthesized**, pop , guitar

Table 3. Automatic 10-word annotations for 'Every little thing she does is magic' by Police.

HEM-DTM performs better on average by capturing temporal dynamics (e.g., tempo, rhythm, etc.) over seconds of audio content. Modeling temporal dynamics can be expected to prove beneficial for some tags, while adding no benefit for others. Indeed, some tags might either be modeled adequately by instantaneous characteristics alone (e.g., timbre), or require a global song model. Table 2 lists annotation (F-score) and retrieval (MAP) results for a subset of our vocabulary. As expected, HEM-DTM shows strong improvements for tags associated with a clear temporal structure. For the genre "classic rock", which has characteristic tempo and rhythm, HEM-DTM achieves an F-score of approximately 0.4, doubling the performance of HEM-GMM. Similarly, HEM-DTM proves particularly suitable for tags with significant temporal structure, e.g., "male lead vocals" and "fast rhythm", or the instruments such as electric or acoustic guitar. Conversely, our HEM-DTM shows no improvement over HEM-GMM when predicting tags for which temporal structure is less significant, such as "mellow" and "negative feelings".

Finally, Tables 3 and 4 show example annotations and retrieval rankings for both HEM-DTM and HEM-GMM. Ground truth results are marked in bold.

6. CONCLUSIONS

We have presented the dynamic texture mixture model; a principled approach for capturing the temporal, as well as timbral qualities of music. We derived a hierarchical algorithm for efficiently learning DTM models from large training sets, enabling its usage as a tag model for semantic annotation and retrieval. Experimental results demonstrate that the new model improves accuracy over current bag-of-feature approaches.

Rank	HEM-DTM	
1	**James Taylor**	**'Fire and rain'**
2	**Arlo Guthrie**	**'Alices restaurant massacree'**
3	Zombies	'Beechwood park'
4	**Crosby, Stills, Nash and Young**	**'Teach your children'**
5	**Donovan**	**'Catch the wind'**
6	**American music club**	**'Jesus hands'**
7	Aaron Neville	'Tell it like it is'
8	10cc	'For you and i'
9	Byrds	'Wasn't born to follow'
10	Beautiful south	'One last love song'
Rank	HEM-GMM	
1	Stranglers	'Golden brown'
2	**Crosby, Stills, Nash and Young**	**'Teach your children'**
3	Pet shop boys	'Being boring'
4	Counting crows	'Speedway'
5	Beth quist	'Survival'
6	Beautiful south	'One last love song'
7	Neutral milk hotel	'Where you'll find me now'
8	Police	'Every little thing she does is magic'
9	Eric clapton	'Wonderful tonight'
10	Belle and Sebastian	'Like Dylan in the movies'

Table 4. Top retrieved songs for 'acoustic guitar'.

Acknowledgments E.C., L.B. and G.R.G.L. wish to acknowledge support from NSF grants DMS-MSPA 0625409 and CCF-0830535.

7. REFERENCES

[1] L. Barrington, A.B. Chan, and G. Lanckriet. Modeling music as a dynamic texture. *IEEE TASLP*, 18(3):602–612, 2010.

[2] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE TPAMI*, 29(3):394–410, March 2007.

[3] A. B. Chan and N. Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *IEEE TPAMI*, 30(5):909–926, May 2008.

[4] A.B. Chan, E. Coviello, and G. Lanckriet. Clustering dynamic textures with the hierarchical em algorithm. In *CVPR*, 2010.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *JRSS B*.

[6] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *Intl. J. Computer Vision*, 51(2):91–109, 2003.

[7] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Advances in Neural Information Processing Systems*, 2007.

[8] A. Gelb. *Applied Optimal Estimation*. MIT Press, 1974.

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

[10] M.I. Mandel and D.P.W. Ellis. Multiple-instance learning for music information retrieval. In *International Conference on MIR*, 2008.

[11] S.R. Ness, A. Theocharis, G. Tzanetakis, and L.G. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proceedings of ACM MULT.*, 2009.

[12] J. Reed and C.H. Lee. A study on music genre classification based on universal acoustic models. *ISMIR*, 2006.

[13] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. *SIGIR*, page 439446, 2007.

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

[15] N. Vasconcelos and A. Lippman. Learning mixture hierarchies. In *NIPS*, 1998.