DYNAMIC TEXTURE MODELS OF MUSIC

Luke Barrington, Antoni B. Chan*, Gert Lanckriet*

Electrical and Computer Engineering Department
University of California, San Diego

*These authors contributed equally to this work

ABSTRACT

In this paper, we consider representing a musical signal as a *dynamic texture*, a model for both the timbral and rhythmical qualities of sound. We apply the new representation to the task of automatic song segmentation. In particular, we cluster *sequences* of audio feature-vectors, extracted from the song, using a dynamic texture mixture model (DTM). We show that the DTM model can both detect transition boundaries and accurately cluster coherent segments. The similarities between the dynamic textures which define these segments are based on both timbral and rhythmic qualities of the music, indicating that the DTM model simultaneously captures two of the important aspects required for automatic music analysis.

***Index Terms*—** Music modeling, dynamic texture model, automatic segmentation, music similarity

1. INTRODUCTION

It is common practice in music information retrieval to represent a song as a bag of audio feature-vectors (e.g., Mel-frequency cepstral coefficients). While this has shown promise in many applications, e.g. music annotation and retrieval [1], audio similarity [2] and song segmentation [3], the bag-of-features representation is fundamentally limited by its assumption that the feature-vectors are *independent* of each other, i.e., the representation ignores the *dependencies* between feature-vectors. As a result, the bag-of-features fails to represent the rhythmic qualities (e.g., tempo and beat patterns) and temporal structure (e.g. repeated riffs and arpeggios) of the audio signal. In this paper, we consider simultaneously modeling both the spectral and rhythmical qualities of a music clip as a *dynamic texture* [4], a generative probabilistic model that models both the timbre of the sound, and its evolution over time. We apply this new audio representation to the task of automatic song segmentation.

The goal of automatic song segmentation is to automatically divide a song into self-coherent units such as the chorus, verse, bridge, etc. Foote [5] segments music based on self-similarity between timbre features. Goto adds high-level assumptions about repeated sections to build a system for automatically detecting choruses [6]. Turnbull et al. [7] present both an unsupervised (picking peaks of difference features) and supervised (boosted decision stump) method for identifying musical segment boundaries (but not labeling the segments themselves). Other methods attempt to explicitly model music and then cast segmentation as a clustering problem. Gaussian mixture models (GMMs) ignore temporal relations between features but have worked well for segmentation and similarity [3] as well as classification of a variety of semantic musical attributes [1]. Hidden Markov models (HMMs) consider transitions between feature states and have offered improvements for segmentation [8] and genre classification [9]. Abdallah et al. [10] incorporate prior knowledge about segment duration into an HMM clustering model to address the problem of over-segmentation. Levy and Sandler [11] realize that feature-level HMMs do not encode sufficient temporal information and constrain their clustering based on the musical structure.

In contrast to these methods which do not explicitly model the temporal qualities of the signal, we introduce a new segmentation algorithm that accounts for both the rhythmic and timbral qualities of the signal. In particular, we cluster *sequences* of audio feature-vectors, extracted from the song, using a mixture of dynamic textures. The new algorithm explicitly models the temporal dynamics of the musical texture, capturing more of the information required to determine the structure of music.

2. DYNAMIC TEXTURE MODELS

Although the dynamic texture (DT) and dynamic texture mixture (DTM) models were originally proposed in the computer vision literature as generative models for video sequences, they are generic models that can be applied to any time-series data. In this paper, we will use the dynamic texture to model a *sequence of audio feature vectors* extracted from a song (e.g., a sequence of Mel-frequency cepstral coefficients). Because we are modeling *sequences*, we are able to capture both the instantaneous audio content (e.g., instrumentation and timbre), and the melodic and rhythmic content (e.g., guitar riffs, drum patterns, and tempo), with a single probabilistic model. Segmentation is performed by extracting a set of sequences from a song using a sliding window, and clustering them with a mixture of dynamic textures. This is analogous to clustering feature-vectors using a Gaussian mixture model (GMM), but the DTM clusters time-series (sequences of feature-vectors), whereas the GMM clusters only feature-vectors. We begin the section with a review of DT and DTM models, followed by a detailed description of the song segmentation algorithm.

2.1. Dynamic textures

In computer vision, a dynamic texture (DT) [4] is a generative model that treats a video sequence as a sample from a linear dynamical system (LDS). Similarly for audio, we can model a sequence of audio feature-vectors as a sample from an LDS. The model captures both the sound and the dynamics of the sequence with two random variables: an *observed variable* $y_t \in \mathbb{R}^m$, which encodes the sound component (audio feature at time t), and a *hidden state variable* $x_t \in \mathbb{R}^n$, which encodes the dynamics (evolution of the sound over time), where $n < m$. The state and observed variables are related through the *linear dynamical system* (LDS) defined by

$$\begin{cases} x_{t+1} = Ax_t + v_t \\ y_t = Cx_t + w_t \end{cases} \quad (1)$$

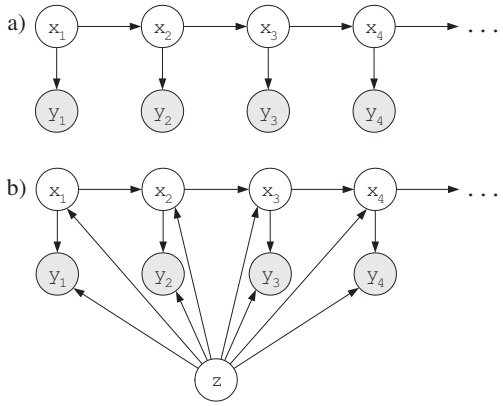

Fig. 1. a) Dynamic texture; b) Dynamic texture mixture. The hidden variable z selects the parameters of the remaining nodes.

The parameter $A \in \mathbb{R}^{n \times n}$ is a *state transition matrix* and $C \in \mathbb{R}^{m \times n}$ is an *observation matrix* (e.g., containing the *principal components* of the audio sequence when learned with [4]). The *driving noise process* v_t is normally distributed with zero mean and covariance Q, i.e., $v_t \sim \mathcal{N}(0, Q)$ where $Q \in \mathbb{S}_+^n$ is a positive-definite $n \times n$ matrix. The *observation noise* w_t is also zero mean and Gaussian, with covariance R, i.e., $w_t \sim \mathcal{N}(0, R)$ where $R \in \mathbb{S}_+^m$. Finally, the initial state is also normally distributed with mean μ and covariance S, i.e., $x_1 \sim \mathcal{N}(\mu, S)$. The dynamic texture is completely specified by the parameters $\Theta = \{A, Q, C, R, \mu, S\}$.

The graphical model of the dynamic texture is shown in Figure 1a. A number of methods are available to learn the parameters of the dynamic texture from a training sequence, including maximum-likelihood methods (e.g., expectation-maximization [12]), non-iterative subspace methods (e.g., N4SID [13]) or a suboptimal, but computationally efficient, procedure [4]. The dynamic texture model has been successfully applied to various computer vision problems, including video texture synthesis [4], video recognition [14, 15], and motion segmentation [16, 17].

2.2. Mixture of Dynamic Textures

While the DT models a single observed sequence, the *mixture of dynamic textures* (DTM) [17] models a collection of sequences as samples from a set of K dynamic textures. This is a useful extension of the DT for clustering time-series. In computer vision, the model has been used to cluster video sequences, and to segment motion in video by clustering patches of video. In this paper, we will use the DTM to segment a song into sections (e.g., verse, chorus, and bridge) in a similar way by clustering sequences of audio feature vectors extracted from the song.

Formally, the DTM is a mixture model where each mixture component is a dynamic texture, and is defined by the system of equations

$$\begin{cases} x_{t+1} = A_z x_t + v_t \\ y_t = C_z x_t + w_t \end{cases} \tag{2}$$

where the random variable $z \sim \text{multinomial}(\alpha_1, \cdots, \alpha_K)$, with $\sum_{j=1}^K \alpha_j = 1$, signals the mixture component from which each sequence is drawn. The remaining variables y_t and x_t form a standard dynamic texture, but with parameters $\Theta_z = \{A_z, Q_z, C_z, R_z, \mu_z, S_z\}$ that are dependent on the active mixture component. The graphical model for the dynamic texture mixture is presented in Figure 1b. Given a set of observed sequences $\{y^{(i)}\}_{i=1}^N$, the maximum-likelihood parameters of the DTM can be learned using the EM algorithm [17].

2.3. Song Segmentation

Song segmentation is performed with the DTM using a coarse-to-fine approach. First, audio features-vectors are extracted from the audio signal (e.g., Mel-frequency cepstral coefficients). To produce a coarse segmentation, short sequences are extracted from the full sequence of audio feature-vectors using a sliding window ($\sim$5 sec) with a large step-size ($\sim$0.5 sec). A DTM is learned from the collection of windowed sequences using EM, and the coarse song segmentation is formed by assigning each windowed sequence to the component with largest posterior probability, i.e.,

$$j^* = \underset{j}{\text{argmax}} \frac{\alpha_j p(y^{(i)}; \Theta_j)}{\sum_{j=1}^K \alpha_j p(y^{(i)}; \Theta_j)} \tag{3}$$

where $p(y^{(i)}; \Theta_j)$ is the likelihood of sequence $y^{(i)}$ under the j-th mixture component Θ_j. This first segmentation is relatively coarse (at best within 0.25 sec), due to the large step-size and the poor-localization properties of using a large window. Next, we refine the boundaries of the coarse segmentation. Sequences are extracted from the song using a smaller sliding window ($\sim$1.75 sec) and a finer step-size ($\sim$0.05 sec). A fine-grain segmentation is formed by assigning these sequences to the most-likely components of the DTM learned in the coarse-segmentation. Finally, the boundaries of the coarse segmentation are refined by searching for the closest boundaries in the fine-grain segmentation, producing the final segmentation. Note that using a large 5 second window for the coarse segmentation allows the DTM to model musical characteristics with long temporal durations (e.g., beat patterns, riffs, sustained notes, etc.). This is not possible when using a shorter window.

3. EXPERIMENTS

3.1. Data

We experiment on 100 pop songs from the RWC music database (RWCMDB- P-2001) [18] where each song has been segmented into coherent parts by a human listener [19]. The segments are accurate to 10ms and are labeled with great detail. For this work we group the labeled segments into 4 possible classes: "verse" (i.e., including verse A, verse B, etc.), "chorus", "bridge" and "other" ("other" includes labels such as "intro", "ending", "pre-chorus", etc. and is also used to model any silent parts of the song). This results in a "ground truth" segmentation of each song with 4 possible segments classes. On average, each song contains 11.1 segments.

3.2. Features

The content of each 22050Hz-sampled, monaural waveform is represented using two types of music information features:

3.2.1. Mel-Frequency Cepstral Coefficients

Mel-frequency cepstral coefficients (MFCCs), developed for speech analysis [20], describe the timbre or spectral shape of a short time piece of audio and are a popular feature for a number of MIR tasks, including segmentation [5, 3, 7]. We compute the first 13 MFCCs for half-overlapping frames of 256 samples (one feature vector every $\sim$ 6 msec). In music information retrieval, it is common to augment the MFCC feature vector with its instantaneous first and second

Model	Error Rate	Rand	# Segments
DTM-MFCC	**0.20**	**0.79**	16.9
GMM-MFCC	0.42	0.66	58.7
Constant	0.59	0.35	1
Random	0.64	0.54	279.0
Truth	0.00	1.00	11.1

Table 1. Song segmentation using MFCC features.

Model	Error Rate	Rand	# Segments
DTM-Chroma	0.26	0.76	13.5
GMM-Chroma	0.46	0.60	24.1
Constant	0.58	0.32	1
Random	0.67	0.56	329.3
Truth	0.00	1.00	11.1

Table 2. Song segmentation using Chroma features.

derivatives, in order to capture some information about the temporal evolution of the feature. When using the DT, this is unnecessary since the temporal evolution is modeled explicitly by the DT.

3.2.2. Chroma

Chroma features have also been successfully applied for song segmentation [6]. They represent the harmonic content of a short-time window of audio by computing the spectral energy present at frequencies that correspond to each of the 12 notes in a standard chromatic scale. We compute a 12-dimensional chroma feature vector from three-quarter overlapping frames of 2048 samples (one feature vector every $\sim$ 23 msec).

3.3. Segmentation

The songs in the RWC database were segmented into $K = 4$ segments using the DTM method described in Section 2.3 on the MFCC or Chroma features, which we denote DTM-MFCC and DTM-Chroma, respectively. For DTM-MFCC, we use a window size of 900 MFCC frames and a step-size of 100 frames, while for DTM-Chroma, we use a window size of 600 Chroma frames and a step-size of 20 frames. The dimension of the hidden state-space of the DTM was $n = 7$ for MFCC, and $n = 6$ for Chroma.

For comparison, we also segment the songs using a Gaussian mixture model (GMM) trained on the same feature data [3]. We learn a $K = 4$ component GMM for each song, and segment by assigning features to the most likely Gaussian component. Since segmentation decisions are now made at the short time-scale of individual features, we smooth the GMM segmentation with a length-1000 maximum-vote filter. We compare the models against two baselines: "constant" assigns all windows to a single segment, "random" selects segment labels for each window at random.

We quantitatively measure the correctness of a segmentations by comparing with the ground-truth using two metrics: 1) the error rate, which is the proportion of the entire song that is assigned to an incorrect segment; 2), the Rand index [21], a clustering metric that intuitively corresponds to the probability that any pair of items will be clustered correctly, with respect to each other (i.e, in the same cluster, or in different clusters). We also report the average number of segments per song. The results are averaged over 100 songs.

Tables 1 and 2 report the segmentation results for the MFCC and Chroma features, respectively. DTM-MFCC outperforms all other models, with an error rate of 0.20 and Rand index of 0.79. GMM performs significantly worse than DTM, e.g., the error rate increases to 0.42 on the MFCC features. Both models tend to over-segment songs although this problem is less severe for DTM. This suggests that there is indeed a benefit in modeling the temporal dynamics with the DTM.

An example of the DTM segmentation of one song is compared to the ground truth in Figure 2 where we see that, while most of the DTM segments are accurate, there are some errors due to imprecise borders, and some cases where the model over-segments.

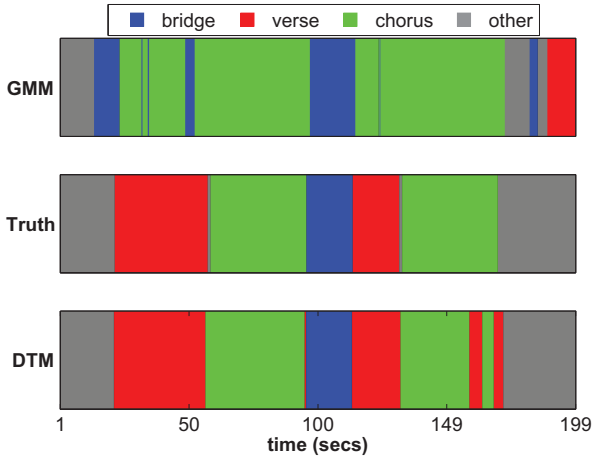

Fig. 2. Example of the true segmentation of a song compared to the automatic GMM (top) and DTM (bottom) segmentations.

3.4. Boundary Detection

In addition to evaluating the segmentation performance of the DTM model, we can consider its accuracy in simply detecting the boundaries between segments (without trying to label the segment classes). The song boundaries are computed by segmenting the song using DTM-MFCC with $K = 5$, and then finding the time instances where the segmentation changes. We compare results with Turnbull et. al [7], which tackles the boundary detection problem, using the same RWC data set, by learning a supervised classifier that is optimized for boundary detection. We also compare with the music analysis company EchoNest [22], which offers an online service for automatically detecting music boundaries.

The evaluation criteria are two median time metrics: true-to-guess and guess-to-true, respectively measure the median time from each true boundary to the closest estimate, and the median time from each estimate to the closest true boundary. The results are averaged over 100 songs and are presented in Table 3. DTM-MFCC achieves both lower guess-to-true and true-to-guess times, indicating that DTM-MFCC is more accurate at finding the song boundaries. Note that DTM-MFCC is an unsupervised method, whereas the next best performer [7] is a supervised algorithm.

Model	Guess-to-True (sec)	True-to-Guess (sec)
DTM-MFCC	**4.06**	**1.76**
Turnbull et. al [7]	4.29	1.82
EchoNest [22]	5.09	1.84

Table 3. Boundary detection using MFCC features.

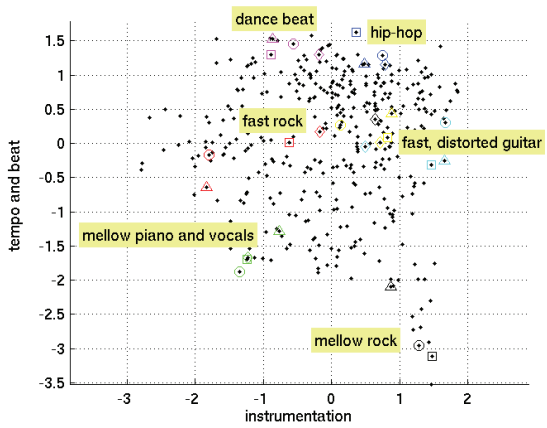

Fig. 3. 2-D visualization of the distribution of song segments. Each black dot is a song segment. Seven songs are highlighted in different colors, with segments marked as ○ (verse), □ (chorus), ♢ (bridge), and △ ("other").

3.5. Song Segment Similarity

Given the automatic segmentation of a song, we can retrieve other similar song clips in the database, answering questions like "what song sounds similar to the verse of this song?" We represent each song segment by its corresponding dynamic texture component in the DTM-MFCC, and measure similarities between dynamic textures with the Kullback-Leibler (KL) divergence [15]. The five closest segments were retrieved for each song segment, and the results are presented online[1]. Qualitatively, the system finds segments that are similar in both audio texture and temporal characteristics. For example, a segment with slow piano will retrieve other slow piano songs, whereas a rock song with piano will retrieve more upbeat segments. This indicates the dynamic texture model is capturing both the "texture" of the audio content (e.g., timbre and instrumentation), along with temporal characteristics (e.g. tempo, beat structures, style).

In order to visualize the distribution of songs in the database, the song segments were embedded into a 3-d manifold using local-linear embedding (LLE) [23] and the KL similarity matrix computed above. Two dimensions of the embedding are shown in Figure 3. We observed that these two axes of the embedding correspond to the tempo and beat of the segment (e.g., dance beat, hip-hop, rock, or mellow), and the instrumentation of the segment (e.g., piano, synthesizers, or distorted guitar). Again, this demonstrates that the dynamic texture model is successfully modeling both the audio texture and the temporal characteristics of the songs. Finally, we selected seven songs that are stylistically different, and highlight them in Figure 3. While most songs are concentrated in specific regions of the manifold (e.g. the sections of the hip-hop song are similar sounding), some songs span multiple regions (e.g., the song highlighted in red contains piano in the verse, and fast upbeat rock in the chorus and bridge).

4. CONCLUSIONS

We have presented the Dynamic Texture Mixture (DTM) model and applied it to analysis of music time series. By describing music as a mixture of coherent textures, we demonstrate that the DTM model can accurately segment music and detects boundaries between segments as accurately as leading research and commercial systems. Examining a low-dimensional representation of the DTM-derived similarity between musical segments illustrates that the model is capturing both timbral and dynamical elements of the music and it shows promise as a new tool for automatic music analysis.

[1] `http://cosmal.ucsd.edu/cal/projects/segment/`

5. REFERENCES

[1] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE TASLP*, vol. 16, no. 2, pp. 467–476, February 2008.

[2] L. Barrington, A.B. Chan, D. Turnbull, and G. Lanckriet, "Audio information retrieval using semantic similarity," in *ICASSP*, 2007.

[3] J.-J. Aucouturier, F. Pachet, and Mark Sandler, "'The Way It Sounds': Timbre models for analysis and retrieval of music signals," *IEEE Transactions on Multimedia*, vol. 7, no. 6, pp. 1028–1035, 2005.

[4] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Intl. J. Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[5] J. Foote, "Visualizing music and audio using self-similarity," in *International Multimedia Conference*, 1999, pp. 77 – 80.

[6] M. Goto, "A chorus-section detection method for musical audio singals and its application to a music listening station," *IEEE TASLP*, vol. 14, no. 1, pp. 1783–1794, 2006.

[7] D. Turnbull, G Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting," in *ISMIR*, 2007.

[8] M. Levy, M. Sandler, and M. Casey, "Extraction of high-level musical structure from audio data and its application to thumbnail generation," in *IEEE ICASSP*, 2006.

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

[10] S. Abdallah, M. Sandler, C. Rhodes, and M. Casey, "Using duration models to reduce fragmentation in audio segmentation," *Machine Learning: Special Issue on Machine Learning in and for Music*, vol. 65, no. 2-3, pp. 485 515, December 2006.

[11] M. Levy and M. Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE TASLP*, vol. 16, no. 2, pp. 318–326, February 2008.

[12] R. H. Shumway and D. S. Stoffer, "An approach to time series smoothing and forecasting using the EM algorithm," *Journal of Time Series Analysis*, vol. 3, no. 4, pp. 253–264, 1982.

[13] P. Van Overschee and B. De Moor, "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, vol. 30, pp. 75–93, 1994.

[14] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *IEEE CVPR*, 2001, vol. 2, pp. 58–63.

[15] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *IEEE CVPR*, 2005, vol. 1, pp. 846–851.

[16] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *IEEE ICCV*, 2003, vol. 2, pp. 1236–42.

[17] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 5, pp. 909–926, May 2008.

[18] M. Goto, "Development of the RWC music database," in *International Congress on Acoustics*, 2004, pp. 553–556.

[19] M. Goto, "AIST annotation for RWC music database," in *ISMIR*, 2004, pp. 553–556.

[20] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[21] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, pp. 193–218, 1985.

[22] "EchoNest," `http://the.echonest.com`.

[23] S.T. Roweis and L.K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.