Discovering Time-Constrained Sequential Patterns for Music Genre Classification

Jia-Min Ren, *Student Member, IEEE*, and Jyh-Shing Roger Jang, *Member, IEEE*

***Abstract*—A music piece can be considered as a sequence of sound events which represent both short-term and long-term temporal information. However, in the task of automatic music genre classification, most of text-categorization-based approaches could only capture temporal local dependencies (e.g., unigram and bigram-based occurrence statistics) to represent music contents. In this paper, we propose the use of time-constrained sequential patterns (TSPs) as effective features for music genre classification. First of all, an automatic language identification technique is performed to tokenize each music piece into a sequence of hidden Markov model indices. Then TSP mining is applied to discover genre-specific TSPs, followed by the computation of occurrence frequencies of TSPs in each music piece. Finally, support vector machine classifiers are employed based on these occurrence frequencies to perform the classification task. Experiments conducted on two widely used datasets for music genre classification, GTZAN and ISMIR2004Genre, show that the proposed method can discover more discriminative temporal structures and achieve a better recognition accuracy than the unigram and bigram-based statistical approach.**

***Index Terms*—Data mining, hidden Markov model (HMM), music genre classification, time-constrained sequential pattern (TSP).**

I. Introduction

How to organize large-scale music databases effectively is an important issue for digital music distribution. Music genre, an important search criterion for music information retrieval, probably provides the most popular description of music contents [1]. However, manually annotating music pieces with genre labels is a time-consuming and laborious task. Specifically, the inherent subjectivity of the notion of genres for different listeners leads to potential inconsistencies of genre labels [2], [3]. Therefore, a number of supervised classification techniques have been proposed in the last decade [4], [15].

Generally, the automatic music genre classification task includes two stages: feature extraction and classifier design. In the stage of feature extraction, short-term features, typically representing the spectral characteristic of music signals, are extracted from short-time window (or frame). Frequently used timbral features include spectral centroid, spectral bandwidth, spectral flux, spectral rolloff, zero-crossing rate, Mel-scale frequency cepstral coefficients (MFCCs) [4], octave-based spectral contrast (OSC) [7], etc. To represent more long-term or high-level characteristics of music signals, rhythmic content features (e.g., beat and tempo), pitch content features (e.g., a pitch histogram and pitch statistics), Daubechies wavelet coefficient histograms (DWCHs) [5], and octave-based modulation spectral contrast (OMSC) [13] are also considered for the classification task. After extracting features from each music piece, one can employ supervised learning approaches to determine the genre for a given music piece. Several popular classifiers include K-nearest neighbor (KNN) [4], [6], Gaussian mixture models (GMMs) [4], [6], [8], linear discriminant analysis (LDA) [6], Adaboost [11], and support vector machines (SVMs) [6], [12].

A. Related Work on Text-Categorization-Based Approaches

The temporal sequences of features provide important information for measuring music similarity [16]. Typically, approaches that consider the genre classification task as a text-categorization problem [17]–[20] use temporal sequences of features to represent temporal information for classification. Li and Sleep [17], [18] quantized MFCCs to create a codebook, and used the codebook assignments to represent the frames of a song as a sequence of codeword indices. To capture temporal information, they utilized a modified Lempel–Ziv coding algorithm to obtain n-gram co-occurrence statistics. Langlois and Marques [19] also applied a similar idea to obtain a textural representation of a song, and then simply used the bigram-based co-occurrence statistics to perform the classification. To capture longer temporal information, Chen *et al.* [20] trained a hidden Markov model (HMM) with 128 states, and then applied Viterbi decoding to represent a music piece as a sequence of HMM states. After counting the occurrences of unigrams and bigrams, they further performed latent semantic indexing (LSI) to obtain weighted features. Finally, multi-class maximal figure-of-merit (MC MFoM) was applied to train classifiers. Although the use of n-grams and HMM states is able to capture temporal information, these representations of short temporal sequences are difficult to model musically meaningful parts [21].

In contrast, it has been suggested that music genre classification can be considered as a spoken language identification problem [22]. Just as language imposes probabilistic constraints on phone and word transitions, syntactic constraints imposed by music genres also influence transition probabilities between fundamental acoustic units (e.g., note and chord) [23]. For example, the 12-bar-blues usually consists of three chords, namely I, IV, and V chords. Therefore, the inherent transition within

Manuscript received March 24, 2011; revised August 01, 2011 and October 07, 2011; accepted October 08, 2011. Date of publication October 17, 2011; date of current version February 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Patrick Naylor.

The authors are with the Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: jmren@mirlab.org; jang@mirlab.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2172426

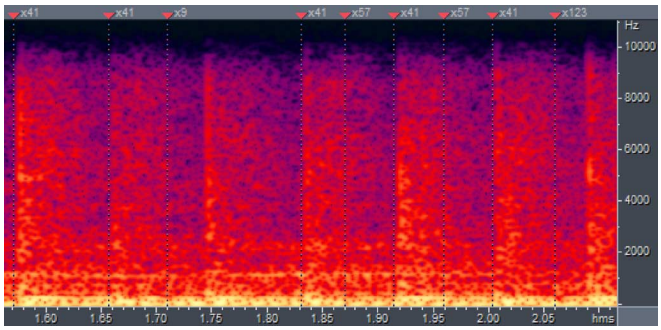


Fig. 1. Spectrogram of a typical clean sound of the Jazz drum in a Metal music clip, where each token (e.g., x41) denotes the start time of a sound event.

note structures could be further exploited once the music pieces can be transcribed with acoustic units. However, unlike automatic speech recognition (ASR) and chord recognition [24], no appropriate segmentations (e.g., phonemes or words) exist for polyphonic music. One possible solution is to build a dictionary of acoustic units in an unsupervised fashion, and then represent a polyphonic music piece with the most probable sequence of these acoustic units. Therefore, Reed and Lee [23] used an acoustic segment modeling framework to transcribe music pieces. To model the temporal nature of music, they used an HMM with three left-to-right states to represent each acoustic segment model (ASM), which is considered as a fundamental acoustic unit for music. After transcribing each music piece into a sequence of ASM indices, Reed and Lee further counted the occurrences of unigrams and bigrams to incorporate temporal information for classification. However, the use of unigram and bigram counts may not be able to capture some temporal structures. Fig. 1 shows the spectrogram of a clean sound of Jazz drum, where each token (e.g., ASM index x41) denotes the start time of a music event. From this figure, we can find that the event x41 nonconsecutively occurs at time 1.83 s, 1.92 s, and 2 s, respectively. This repeating but nonconsecutive music event, which indeed captures the characteristic of Metal music, cannot be represented by any unigram and bigram.

B. Proposed Time-Constrained Sequential Patterns (TSPs)-Based Music Genre Classification System

One possible way to find essential but nonconsecutive temporal structures in genre-specific songs is to apply sequential patterns (SPs) mining on genre-specific music transcripts [25]. Here an SP consists of a sequence of one or more tokens (where a token denotes an ASM index) that are likely to occur in music transcripts. In other words, we can use SPs to capture the common characteristics of the temporal structures within a specific genre. However, since there is no time constraint between any two consecutive tokens in SPs, some of the mined SPs may not correspond well to human aural perception. For example, the time gap between two consecutive tokens may be longer than 5 seconds. To tackle this problem, we proposed the use of time-constrained sequential patterns (TSPs) which limit the time gap between any two consecutive tokens [26]. Moreover, since music of different genres may have different tempos (and thus different distributions for token durations), this paper proposes a systematic way to identify a genre-specific time gap value such that a suitable set of TSPs can be mined from each genre. As shown in our experiments (Section IV-C), such genre-specific time gap values did improve the classification accuracy.

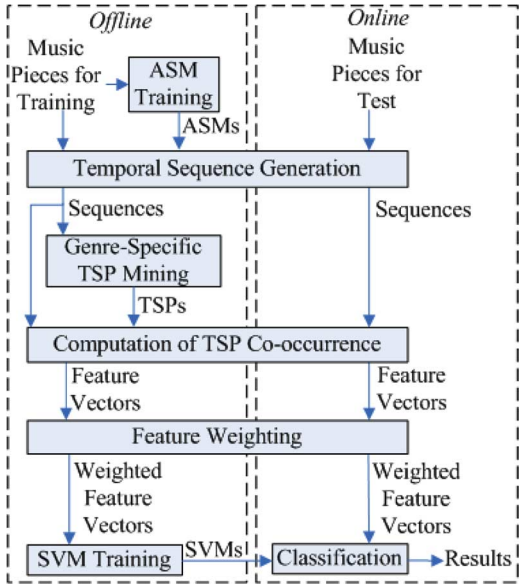


Fig. 2. Flowchart of the proposed music genre classification system.

Fig. 2 shows a flowchart of the proposed system. We first train a set of acoustic units (e.g., ASMs) in an unsupervised fashion, and then each piece of music is transcribed into a temporal sequence of tokens. In order to discover temporal structures in a specific genre, we mine TSPs from genre-specific sequences. All the mined TSPs are then used to compute their occurrence frequencies in a piece of music so that each piece of music can be represented as a feature vector. By weighting each feature vector with respect to TSP length and TSP's probability within each genre, we finally use these weighted feature vectors to train SVMs for classification.

To the best of our knowledge, we are the first to propose the framework of SP and TSP mining in representing essential but nonconsecutive temporal structures for audio music genre classification. Previous studies mostly focused on discovering repeating parts between music pieces in order to compute music similarity [21], [27] or mining repeatedly consecutive parts on symbolic music data [3], [28]. However, few studies discussed how to discover similar patterns among different genre of audio music. In addition, SP mining has been successfully applied to the classification of images [29], trajectories on road networks [30], and bio-sequence (e.g., DNA or proteins) [31] in recent years. Similar to these tasks, music sequences also contain temporal structures, and the successful application of SPs to genre classification of audio music can be expected.

The remainder of this paper is organized as follows. Section II describes how to generate music transcripts in an unsupervised fashion. Section III introduces the proposed TSP-based classification method. Section IV presents our experiments and compares the performance of the proposed method with other existing approaches. Finally, Section V provides conclusions and possible future directions of this work.

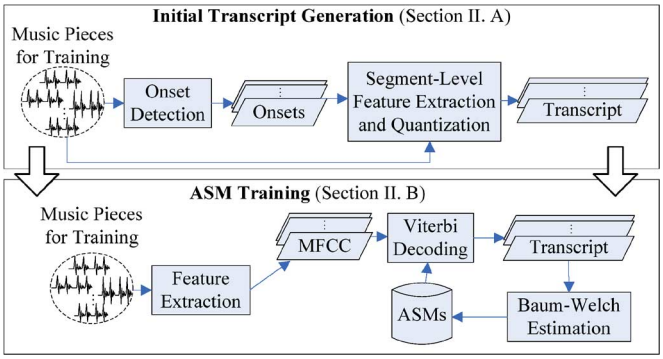


Fig. 3. Flowchart of training acoustic segment models (ASMs) in an unsupervised fashion.

II. Acoustic Units Based Music Transcripts

In conventional ASR, an utterance is usually modeled as a temporal concatenation of acoustic units (e.g., phonemes or words). Due to acoustic variability, these acoustic units are probabilistically estimated from manually transcribed utterances via a supervised process. Although there are many transcribed speech corpora, few transcribed databases exist for polyphonic music. In addition, it is unclear how to efficiently segment polyphonic music into sound events in a supervised fashion. Therefore, based on an unsupervised language identification approach [32], we train a vocabulary of acoustic units to model universal sound events, such that a given music piece can be tokenized into a sequence of acoustic units. As shown in Fig. 3, the tokenization procedure consists of two stages. The first stage is to quantize onset-derived segments into an initial transcript for each music piece in the training corpus (Section II-A). The second stage is to model a universal set of acoustic units, namely acoustic segment models (ASMs), in an iterative training procedure (Section II-B). These two stages are described as follows.

A. Initial Transcript Generation

The first step to creating music transcripts is to find an appropriate segmentation for each music piece. In [23], Reed and Lee divided each music piece into a fixed number of segments, and then refined the segment boundaries in a maximum-likelihood manner. However, transcribing music pieces of different genres into the same number of segments is not reasonable since the tempos of different genres vary. For example, the tempo of Jazz is usually slower than that of Hiphop. Therefore, it is more reasonable to divide music of different genres into different number of segments. To tackle this problem, music onsets (which describe the start time of a sudden change in the short-time spectrum) are used to determine the initial boundaries of segments. More specifically, we use the spectral flux to determine an onset function [33]

$$SF(n) = \sum_{k=1}^{N} H\left(|X(n,k)| - |X(n-1,k)|\right) \qquad (1)$$

where $|X(n,k)|$ is the magnitude spectrum of the kth frequency bin of the nth frame, N is the window size, and $H(x) = (x + |x|)/2$ is the half-wave rectifier. After normalizing the onset curve to zero mean and unit variance, we selected peaks which satisfy three local maximum conditions as onsets [33].

After the onsets are obtained, music signals between two adjacent onsets are considered as a segment. In order to construct the vocabulary (a set of labels) for these onset-derived segments, we use vector quantization to find the global codebook in the following steps.

1) The first eight MFCCs, which describe the slowly changing spectral shape [23], are extracted from the frames within each segment (zero-mean, unit-variance normalization is also performed on each MFCC dimension within a music piece).
2) Each segment is represented as the concatenation of the mean and standard deviation along each MFCC dimension to form a segment-level feature vector of 16 dimensions.
3) Vector quantization is performed on the segment-level features of all training music pieces to obtain a global codebook, and then each segment is assigned with a label which is the index of the nearest codeword. Hence, each music piece is converted into a sequence of tokens.

These sequences are considered as initial transcripts of the music pieces. It should be noted that the number of ASMs (or equivalently, the codebook size, or the number of labels in the vocabulary) should be given in advance before vector quantization is performed. Note also that the reason to use only the first eight MFCCs to construct initial transcripts is that since the later ASM training will refine better boundaries of segments, at this stage, sacrificing a finer segment for speed is a fair tradeoff [23].

B. ASM Training

The labels and segments obtained in the previous stage can be used to train a set of acoustic units. In order to capture the temporal sequence of music, we use an HMM with three left-to-right states to model each ASM. This setup is the same as most of the state-of-the-art ASR systems which use an HMM with three left-to-right states to model a tri-phone in order to capture the temporal structures of speech. Following the work of Reed and Lee [23], here we use 39-dimensional MFCCs (including the first 12 MFCCs, energy, and their derivatives and accelerations) for ASM training. Zero-mean and unit-variance normalization is also performed on each feature dimension within a music piece.

More specifically, the process of ASM training can be summarized as follows. First, the transcripts as well as all MFCC features in the training corpus are used to train a set of ASMs via Baum–Welch estimation. Next, the trained ASMs are used to decode each music piece into a sequence of ASMs indices via Viterbi decoding. Note that these new transcripts may be different from the original ones. Then, in order to expect these ASMs to be able to better represent music events, these new transcripts are used to further refine the ASMs in an iterative process between Baum–Welch estimation and Viterbi decoding. This iterative process is generally repeated three to five times until convergence is reached. The whole off-line training process is usually time-consuming, but it will not affect the online response time of the classification system.

III. TSP-Based Music Genre Classification

After using ASMs to transcribe each music piece in the training corpus into a sequence of tokens, we can obtain

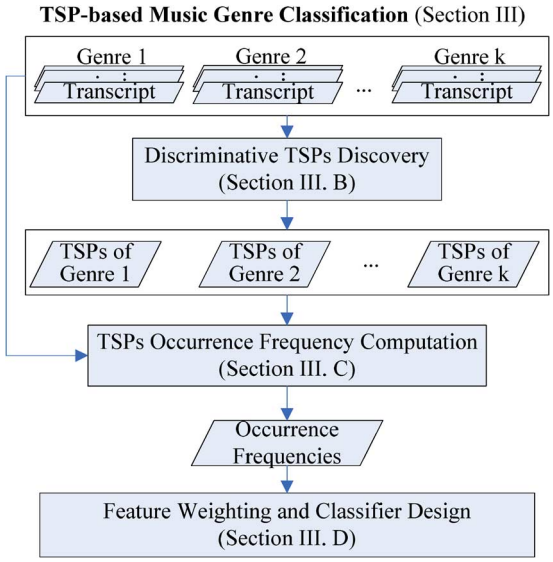


Fig. 4. Flowchart of the TSP-based music genre classification system.

genre-specific music transcripts. Fig. 4 shows a flowchart of constructing a music genre classification system based on these music transcripts. First of all, in order to discover essential but potentially nonconsecutive temporal structures in a specific genre, we can mine TSPs from genre-specific music transcripts (Section III-B). In particular, the tempos of different genres usually vary such that the distribution of token durations in each genre is diverse. Therefore, we further derive a genre-specific time gap based on the distribution of token durations of a specific genre for mining genre-specific TSPs. Next, we compute the feature vector of a music piece as the occurrence frequency of each mined TSP since these mined TSPs can be used to represent genre-specific characteristics (Section III-C). Finally, each feature vector is weighted by its TSP length and the TSP's probability within each genre, and then all the weighted vectors are fed to linear SVMs for training classifiers (Section III-D).

Before describing the proposed method in detail, we have to first offer a formal definition of TSP. TSP can be defined as follows.

A. Definition of Time-Constrained Sequential Pattern (TSP)

After transcribing each music piece as a temporal sequence of ASMs, we can represent a music transcript as n ASM indices $\langle (e_1)_{t_1}(e_2)_{t_2}\cdots(e_n)_{t_n}\rangle$, where e_i represents a token (e.g., an ASM index), t_i denotes the start time of e_i, and $t_1 < t_2 < \cdots < t_n$. For clarity, we also define the following notations.

1) A music transcript $\alpha = \langle (a_1)_{t_1}(a_2)_{t_2}\cdots(a_m)_{t_m}\rangle$ is said to contain a sequence (without time label) $\beta = \langle (b_1)(b_2)\ldots(b_n)\rangle$, denoted as $\beta \sqsubseteq \alpha$, iff $\exists i_1, i_2, \ldots, i_n$ such that a) $1 \le i_1 < i_2 < \ldots < i_n \le m$, b) $b_j = a_{i_j}$, $1 \le j \le n$, and c) $t_{i_j} - t_{i_{j-1}} \le maxgap$, $2 \le j \le n$, where $maxgap$ (maximum gap) is a pre-defined threshold. In other words, β is a subsequence of α, and the time gap between the start time of a_{i_j} (which is equal to b_j in α) and the start time of $a_{i_{j-1}}$ (which is equal to b_{j-1} in α) must be equal to or smaller than $maxgap$.
2) $|D|$ indicates the number of transcripts in a set of music transcripts D.
3) $D_\beta = \{s | s \in D \text{ and } \beta \sqsubseteq s\}$ denotes the set of transcripts in D which contains β.
4) The support of a sequence β is $|D_\beta|/|D|$.

Here these discovered sequences are named TSPs rather than SPs since they are mined under a specific time constraint. Moreover, the supports of the mined TSPs exceed another predefined threshold, $minsup$ (minimum support). Note that the value of $minsup$ is between 0 and 1, and the value of $maxgap$ must be positive. In other words, we consider a sequence as a TSP only if there are at least $\lceil minsup \times |D| \rceil$ music transcripts which contain this sequence. In addition, while there is no time label in TSPs, $maxgap$ indirectly constrains the time gap between any two consecutive tokens in TSPs.

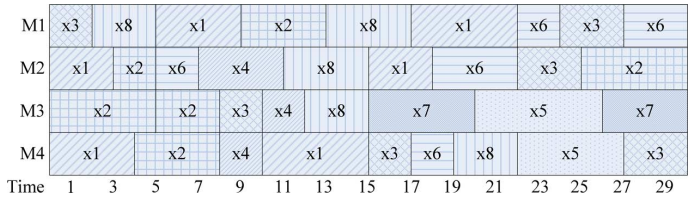


Fig. 5. Four music transcripts.

TABLE I
TSPs MINED FROM MUSIC TRANSCRIPTS IN FIG. 5. ($minsup = 0.5$ AND $maxgap = 4$)

(x1):3, (x1, x2):2, (x1, x2, x4):2, (x2):4, (x2, x4):2, (x3):4, (x3, x6):2
(x3, x8):3, (x4):3, (x4, x8):2, (x5):2, (x6):3, (x6, x3):2, (x8):4, (x8, x1):2

Here, four music transcripts of a specific genre, as depicted in Fig. 5, are used to explain the kinds of TSP that can be mined from this collection. In this figure, each transcript is represented by a vector of tokens and their corresponding start time. For example, the first two tokens in M2 are $(\text{x1})_0$ and $(\text{x2})_3$, indicating that these two tokens start at time 0 and 3, respectively. Given this collection, we can apply any TSP mining algorithm to discover TSPs. Table I shows the results of the mined TSPs under the constraints of $minsup = 0.5$ and $maxgap = 4$. Here (x3, x8):3 denotes that the TSP (x3, x8) is contained in three transcripts (M1, M3, and M4 in this case). It should be noted that the tokens of this TSP appear in both M3 and M4 but not consecutively. In other words, both the subsequence $\langle (\text{x3})_8(\text{x4})_{10}(\text{x8})_{12}\rangle$ in M3 and the subsequence $\langle (\text{x3})_{15}(\text{x6})_{17}(\text{x8})_{19}\rangle$ in M4 contain this TSP. Moreover, the TSP (x2, x4) appears in M2 and M4; however, M3 does not contain this TSP, since the time gap between the tokens $\langle (\text{x2})_5(\text{x4})_{10}\rangle$ exceeds 4. Interestingly, the TSP (x8, x1) can be considered as a bigram (i.e., an ordered ASM pair) since its tokens appear consecutively in M1 and M2. In general, n-grams are special cases of TSPs.

B. Discriminative TSPs Discovery

In order to discover TSPs which can represent genre-specific characteristics, we apply TSP mining on a collection of genre-specific music transcripts. For efficiency, a state-of-the-art algorithm [34] is chosen for the mining of TSPs.

Before TSP mining, we need to set the values of $minsup$ and $maxgap$ such that discriminative TSPs can be identified. To setup the value of $minsup$, we used the information gain [35]

(which is more intuitive and can be empirically set) to guide the search. To set the value of $maxgap$, we used the mean and standard deviation of token durations (which measure the distribution of token durations in a specific genre) to derive an appropriate value for each genre. These two values are described in the two subsections below.

1) Minimum Support: By representing a pattern as a random variable X and a genre as a random variable G, we can define the information gain as (2), where $E(G)$ and $E(G|X)$ denote the *entropy* and the *conditional entropy*, respectively,

$$IG(G|X) = E(G) - E(G|X). \tag{2}$$

Cheng *et al.* [36] demonstrated that the discriminative power of lower support patterns is limited. That is, the infrequent patterns have a very low information gain upper bound. More specifically, $E(G)$ is a constant for a given dataset with a fixed genre distribution. Thus, the upper bound of the information gain (IG_{ub}) is obtained when $E(G|X)$ reaches its lower bound ($E_{lb}(G|X)$). $E(G|X)$ is defined as

$$\begin{aligned} E(G|X) = &-P(x) \sum_{i=1}^{numgenre} P(g_i|x) \log P(g_i|x) \\ &-P(\overline{x}) \sum_{i=1}^{numgenre} P(g_i|\overline{x}) \log P(g_i|\overline{x}) \end{aligned} \tag{3}$$

where $P(x)$ is the probability of a pattern x (which is also equal to the relative support θ of x), $P(\overline{x})$ is the probability of the absence of x, $P(g_i|x)$ is the proportion of music transcripts that contains a pattern x belonging to the genre g_i, and $P(g_i|\overline{x})$ is the ratio of music transcripts that does not contain a pattern x belonging to the genre g_i [30].

$E_{lb}(G|X)$ is reached when x appears in as fewer genres as possible. To simplify the analysis, suppose that x exists only in a genre g_i. Then in the case of $\theta \leq \max(P(g_i))$, $E_{lb}(G|X)$ is formulated by (4). Similar result can also be derived for $\theta > \max(P(g_i))$:

$$\begin{aligned} &E_{lb}(G|X)_{|P(g_i|x)=1 \wedge P(g_j|x)=0, \forall j \neq i} \\ &= -P(\overline{x}) \left(\frac{P(g_i) - P(x)}{P(\overline{x})} \log \frac{P(g_i) - P(x)}{P(\overline{x})} \right. \\ &\qquad \left. + \sum_{j \neq i} \frac{P(g_j)}{P(\overline{x})} \log \frac{P(g_j)}{P(\overline{x})} \right) \\ &= -(P(g_i) - \theta) \log \frac{P(g_i) - \theta}{1 - \theta} \\ &\quad - \sum_{j \neq i} P(g_j) \log \frac{P(g_j)}{1 - \theta}. \end{aligned} \tag{4}$$

Given an information gain threshold IG_0, a strategy for setting the value of $minsup$ is described as follows [36]. First, the theoretical information gain upper bound is computed as a function $IG_{ub}(\theta)$ of the support θ. Here, the function $IG_{ub}(\theta)$ is the information gain upper bound at the support θ, and is computed directly from the genre label distribution. Then, we decide IG_0 and find $\theta^* = \arg\max_\theta(IG_{ub}(\theta) \leq IG_0)$, which is the value of $minsup$. TSPs with support $\theta < \theta^*$ are discarded because $IG(\theta) \leq IG_{ub}(\theta) < IG_{ub}(\theta^*) \leq IG_0$. In this way, TSPs with better discriminative power are identified efficiently.

2) Maximum Gap: After representing a music piece as a temporal sequence of ASM tokens, we observed that more acoustic units are usually needed to represent a fast-tempo song due to large variations between music notes. In other words, the average duration of tokens in a fast-tempo music piece (e.g., a Hiphop song) is usually shorter than that of a slow-tempo music piece (e.g., a Jazz song). Therefore, we can define the value of $maxgap$ for genre g_i as

$$maxgap_{g_i} = mean_{g_i} + \alpha \times std_{g_i} \tag{5}$$

where $mean_{g_i}$ and std_{g_i} are the mean and standard deviation of token durations of genre g_i, and α is an integer constant. Note that the values of $mean_{g_i}$ and std_{g_i} are automatically determined according to the distribution of token durations of genre g_i. In addition, an experiment on different values of α will be conducted and discussed in Section IV.

C. TSPs Occurrence Frequency Computation

Since TSPs can be used to capture genre-specific temporal structures, it is intuitive to represent each music transcript as a feature vector composed of the occurrence frequency of TSPs mined from each genre. Note that each TSP is identified with a genre-specific $maxgap$. Hence, when we compute the occurrence frequency of a TSP within a music transcript, the time constraint imposed by $maxgap$ must be satisfied. Specifically, the procedure for computing the occurrence frequency of a TSP within a music transcript is described as follows.

Goal: compute the occurrence frequency of a TSP in a music transcript

Input: a music transcript: $\gamma = \langle (r_1)_{t_1} (r_2)_{t_2} \cdots (r_m)_{t_m} \rangle$

a TSP:$\omega = \langle (w_1)(w_2) \ldots (w_n) \rangle$,

time constraint: $maxgap$

Output: the occurrence frequency of ω in γ, $freq$

---

1. set $freq = 0$;
2. **while** true
3. set n =the number of tokens in γ;
4. find the minimum integers $i_1, i_2, \ldots, i_n$ such that the following constraints hold:
   1) $1 \leq i_1 < i_2 < \ldots < i_n \leq m$;
   2) $w_j = r_{i_j}$, $1 \leq j \leq n$;
   3) $t_{i_j} - t_{i_{j-1}} \leq maxgap$, $2 \leq j \leq n$;
5. **if** no such integers exist, then **break**; **endif**
6. $freq = freq + 1$;
7. remove the first i_n tokens from γ;
8. **end while**

Take the TSP (x3, x8) in Table I for instance. Given the time constraint, $maxgap = 4$, and a music transcript, $\langle (x2)_0 (x4)_4 (x3)_9 (x6)_{11} (x8)_{13} (x3)_{19} (x8)_{23} (x4)_{26} \rangle$, we can find that this TSP appears at time 9 and 19, respectively. Therefore, the occurrence frequency of this TSP in this music

transcript is two. After performing the procedure of computing the occurrence frequency of TSPs mined from all genres within a music transcript, we can represent each music transcript in the training corpus as a feature vector, where the ith entry is equal to the occurrence frequency of the ith TSP in this music transcript. The same operation is also performed to convert a music transcript of the test set into a feature vector.

D. Feature Weighting and Classifier Design

In text retrieval, the term frequency and inverse document frequency are often used to adjust the weights of different terms [37]. By the same token, we use the genre probability of a TSP to enhance its discriminative power. The genre probability of a TSP is defined as

$$P(TSP, g_i) = \frac{\sum_{s \in g_i} freq(TSP, s)}{\sum_r freq(TSP, r)} \tag{6}$$

where s is a music sequence in genre g_i, r is one of the training sequences, and $freq(TSP, s)$ is the occurrence frequency of this TSP in s. A genre probability of a TSP close to one indicates that this TSP mostly appears in a specific genre, whereas a value close to zero means this TSP rarely occurs in the genre. In addition, the length of a TSP (i.e., the number of tokens in a TSP) is also considered as a significant factor since the longer a TSP is, the more important it should be. Therefore, a weighted feature vector is obtained by multiplying each entry in the original feature vector with the corresponding TSP's genre probability and length.

Finally, linear SVMs [38] are used as classifiers due to the following two considerations [30]. First, the feature vector is high-dimensional since quite a number of TSPs may be discovered from each genre's music transcripts. Second, the feature vector is sparse since each music transcript contains most of the TSPs that are mined from the same genre as the music transcript. On the other hand, they usually contain only a few TSPs that are mined from the other genre's music transcripts. SVM is eminently suitable for classifying such high-dimensional and sparse data [39].

IV. Experimental Results

A. Datasets

Two popular datasets were used to evaluate the performance of the proposed method. The first dataset (GTZAN), collected by Tzanetakis [4], consists of ten genre classes: Blues (Bl), Classical (Cl), Country (Co), Disco (Di), Hiphop (Hi), Jazz (Ja), Metal (Me), Pop (Po), Reggae (Re), and Rock (Ro). Each class contains 100 30-s music pieces, and each piece is stored at a 22 050 Hz, 16-bit, and mono audio file. The second dataset (ISMIR2004Genre), used in the *ISMIR2004 Genre Classification Contest* [40], has six genre classes and 1458 full music tracks in total. The number of tracks in each class is described as follows: Classical (640), Electronic (229), Jazz/Blues (52), Metal/Punk (90), Rock/Pop (203), and World (244). The format of these tracks is 44 100 Hz, 128-kbps, 16-bit, and stereo MP3 file. In this study, we converted these full tracks into 22050 Hz, 16-bit, and mono audio files. In addition, we also extracted a segment of 30 seconds from the middle of each full track since the introductory part may not be directly related to music genre. In other words, all of the music pieces used in the following experiments is 30 seconds in duration.

B. Evaluation Method and Experimental Setups

For the comparison with other existing approaches on the GTZAN dataset (see Section IV-F for more details), a stratified 10-fold cross-validation is considered in the following experiments. In other words, this dataset is equally divided into 10 folds, where nine folds are used for ASM training and the remaining fold is used to evaluate the performance. The overall accuracy is computed by averaging the accuracy of each fold. For the ISMIR2004Genre dataset, we used the same training and test set partitions as that used in the *ISMIR2004 Genre Classification Contest*, where 729 music tracks are used for training and the rest for test. The classification accuracy is calculated as the number of correctly classified music tracks divided by the number of all test music tracks. Note that these two datasets are treated independently. In other words, the ASMs trained from the GTZAN dataset were not involved in the tokenization process for the music pieces in the other dataset and vise versa.

Several important parameters involved in different stages of the proposed method are discussed next. In the stage of ASM training, the number of ASMs affects whether the trained ASMs are able to capture the common and salient characteristics of the music corpus under consideration. Specifically, a small number of ASMs are not able to model acoustic features of all music adequately, whereas a large number of ASMs are likely to have poor generalization unless we have a large quantity of training data. We set the number of ASMs to be 64, 128, and 256 respectively in our experiments, with their generalization capability discussed in Section IV-C.

In the stage of TSP mining, the value of $minsup$ is derived from the information gain threshold IG_0. As suggested by Lee *et al.* [30], a desirable value of IG0 is around 0.2. Therefore, we empirically set this value to 0.25 and 0.15 for the GTZAN and the ISMIR2004Genre datasets, respectively. For the GTZAN dataset, the information gain function can be formulated by (7), and this function is illustrated in Fig. 6. Therefore, we decided $minsup$ to be 0.06. Similarly, $minsup$ for the ISMIR2004Genre dataset was set to 0.11. In addition, the value of $maxgap$ was determined according to the distribution of token durations in a specific genre. Fig. 7 shows a box-and-whisker plot for the average token durations of music pieces of the GTZAN dataset. From this figure, it is obvious that the token durations of music pieces in different genre are very diverse. In addition, Fig. 8 also shows an example of the histogram of token durations of Hiphop and Jazz (GTZAN). As we can see from this figure, Hiphop has more shorter-duration tokens while Jazz has more longer-duration tokens. Therefore, we used (5) to measure the distribution of token durations. Specifically, we empirically set α to 0, 1, 2, 3, and 4, and then discussed the classification accuracies and the number of mined TSPs under these five different setups in the experiments:

$$IG_{ub}(\theta) = -10 \times 0.1 \log 0.1 - \left\{ -(0.1-\theta) \log \frac{0.1-\theta}{1-\theta} - 9 \times 0.1 \log \frac{0.1-\theta}{1-\theta} \right\}. \tag{7}$$

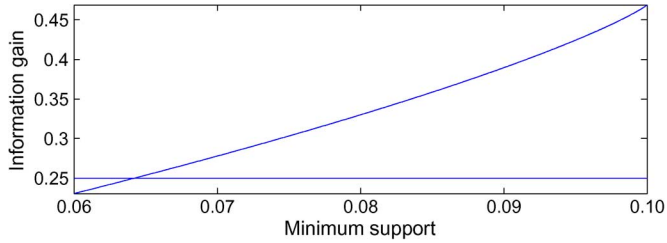


Fig. 6. Plot of (7).

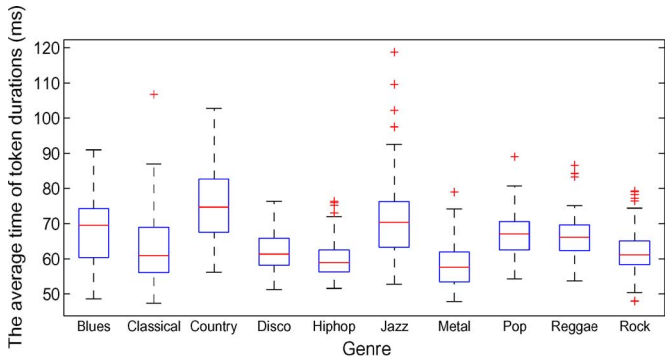


Fig. 7. Average token durations of music pieces of different genres on the GTZAN dataset.

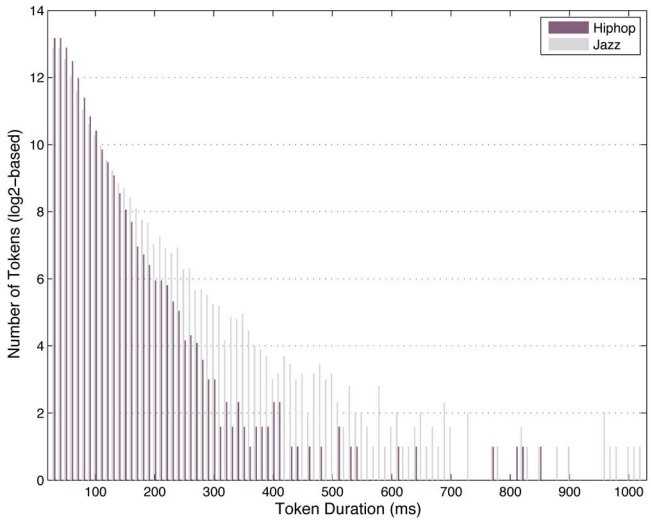


Fig. 8. Histogram of token durations of Hiphop versus Jazz for the GTZAN dataset (mean value of token durations: 59.47 versus 69.44; standard deviation of token durations: 34.23 versus 57.26; note that we have added 1 to the original numbers to avoid minus infinity after taking logarithm.)

In the stage of classifier design, linear SVMs are used as classifiers. Here, we applied a grid search algorithm to tune the penalty parameter [41]. It should be noted that since the number of TSPs (usually more than 2000 for each genre) is much larger than the number of training instances, it is unlikely that the projection to a higher dimensional space by kernel function will improve the performance. [41]. Therefore, we do not need to train radial basis function (RBF) SVMs as classifiers in this case.

C. Classification Results

The approach of unigrams and bigrams adopted by Reed and Lee [23] is considered as the baseline in our experiments. The

TABLE II
CLASSIFICATION ACCURACY OF UNIGRAM AND BIGRAM METHOD (BASELINE)

# of ASM	Dataset	Accuracy (Std. Dev.)
64	GTZAN	72.6% (3.86)
128	GTZAN	75.4% (3.72)
256	GTZAN	75.2% (4.13)
64	ISMIR2004Genre	72.57%
128	ISMIR2004Genre	74.35%
256	ISMIR2004Genre	74.07%

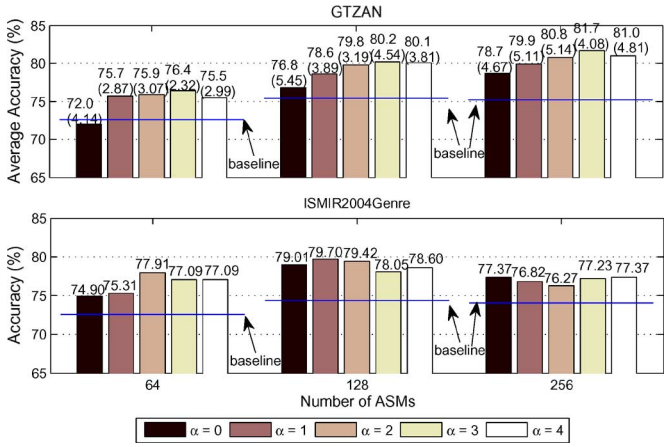


Fig. 9. Classification accuracy of the proposed method with different numbers of ASMs and different values of α. The values in the parentheses of the top panel (GTZAN dataset) indicate the standard deviation of accuracy of 10-fold cross-validation.

baseline is briefly described next. We first count the co-occurrences of unigrams and bigrams for each music transcript, and then transform the original feature vectors into weighted vectors. Finally, these weighted vectors are used to train linear SVM classifiers. Table II shows the accuracy of the baseline. It is obvious that the use of 128 ASMs can achieve a better result than the use of only 64 ASMs in both datasets. An explanation to this observation is that the use of 64 ASMs is not enough to model different genre's characteristics. Besides, a better accuracy cannot be achieved when we use 256 ASMs to model music characteristics. This means that the use of 128 ASMs is enough to cover music variations without over generalization. Therefore, we set the largest number of ASMs to 256 in all experiments.

Fig. 9 shows the accuracy of the proposed method with different numbers of ASMs and different values of α. It is obvious that in both datasets, the proposed method outperforms the baseline, except for the case of the GTZAN dataset when $\alpha = 0$ and the number of ASMs is 64 (the leftmost bar in the top panel). This demonstrates that TSPs can be used to discover more discriminative features than that of unigrams and bigrams in most cases. For the GTZAN dataset, we also found that under the same value of α, a higher accuracy can be achieved when the number of ASMs is larger. Specifically, the use of 256 ASMs achieves a better result than the use of only 128 ASMs. However, the similar phenomenon cannot be observed in the baseline. This observation indicates that the use of TSPs can indeed discover some discriminative temporal structures that cannot be simply represented by unigrams and bigrams in this dataset. On

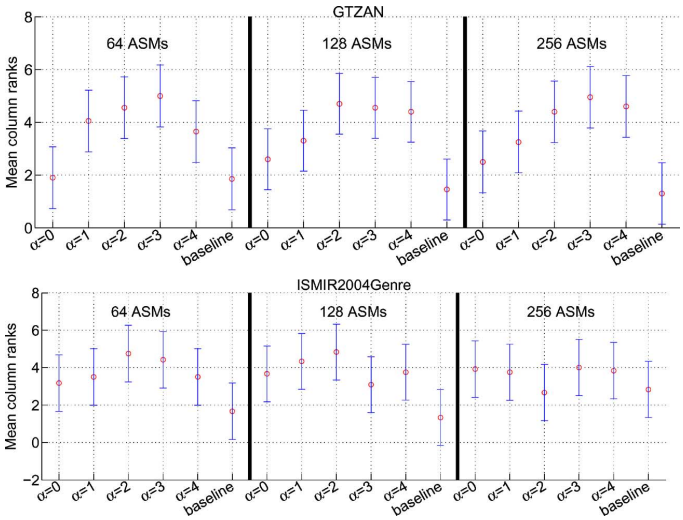


Fig. 10. Results of Friedman's test (at the 5% level) of the proposed method and the baseline under different numbers of ASMs and different values of α (the top panel: GTZAN; the bottom panel: ISMIR2004Genre).

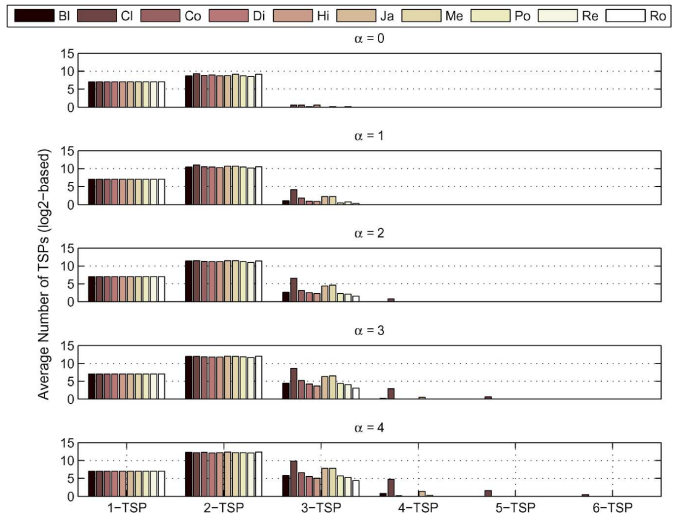


Fig. 11. Average numbers of TSPs mined from ten genres under 128 ASMs and different values of α on the GTZAN dataset. (Note that we have added 1 to the original average numbers to avoid minus infinity after taking logarithm.)

the other hand, under the same number of ASMs, the accuracy increases with α until $\alpha = 4$. This reflects that the TSPs mined under the parameter setting $\alpha = 3$ are enough to represent the characteristics in different genres. A large value of α may lead to more non-discriminative TSPs which reduce the performance. However, these two observations are not applicable to the ISMIR2004Genre dataset due to its unbalanced distribution of music pieces in each genre. As mentioned earlier, the distribution of ISMIR2004Genre is: Classical (640), Electronic (229), Jazz/Blues (52), Metal/Punk (90), Rock/Pop (203), and World (244). In particular, we have only 52 tracks for Jazz/Blues. Such unbalanced distribution makes the performance highly sensitive to the partition of the dataset. This issue is further complicated because the evaluation of the dataset is based on the hold-out test (training and test sets specified by MIREX [40]), which is less reliable than, say, 10-fold cross-validation. As a result, the observations for GTZAN do not hold true for ISMIR2004Genre.

In comparison to our previous work [26], in which we used a global $maxgap$ to mine TSPs of different genres and obtained an accuracy of 74.5% for the GTZAN dataset, the proposed method in this study achieves an accuracy of 80.2% (under 128 ASMs). This indicates that genre-specific $maxgap$ can be used to discover genre-specific temporal structures with better performance. To consolidate our findings, we used Friedman's test [42] to evaluate the statistical significance of the classification results at the 5% level. (This test is also used to evaluate the performance of submissions to MIREX tasks, such as mixed popular genre classification, music mood classification, audio tag classification, etc.). Fig. 10 shows the results of this test. Under the same number of ASMs, if there is any overlap between two confidence intervals, then the performances of the corresponding algorithms are not significantly different. More specifically, here we used the test to compare the accuracies of different algorithms on each fold under the same number of ASMs for the GTZAN dataset (the top panel). And for the ISMIR2004Genre dataset (the bottom panel), we used the test to compare genre accuracies of different algorithms under the same number of ASMs. The results show that the proposed method *significantly* outperforms the baseline in most cases on the GTZAN dataset. For the ISMIR2004Genre dataset, the proposed method *significantly* outperforms the baseline when $\alpha = 2$ and the number of ASMs is 64 and 128, respectively.

D. TSPs Analysis—Lengths and Counts

In TSP analysis, we compared the lengths and counts of TSPs in different genres to show that the use of TSPs can indeed discover long-term temporal information, which is critical for genre classification. For simplicity, a TSP with l tokens is denoted as l-TSP.

Fig. 11 shows the average numbers of TSPs (log2-based) mined from different genres under 128 ASMs and different values of α on the GTZAN dataset. From this figure, the average number of 1-TSP does not vary much in each genre, since 1-TSP appears almost in each genre. For 2-TSP, we can see that the average number of 2-TSP increases when α increases. The phenomenon is due to the fact that a larger α leads to a larger $maxgap$ and this allows us to discover more 2-TSPs. For the TSPs with long-term temporal information (e.g., 3-TSPs, 4-TSPs, etc.), we also observed a similar phenomenon for the same reason. Interestingly, the average numbers of TSPs in Classical, Metal, and Jazz rank top three out of the ten genres studied. This is reasonable since most of the music pieces in these three genres have more repeating patterns than the others. For instance, according to our observation, there are dense sounds of Jazz drum in most of Metal music pieces.

Fig. 12 demonstrates the result of the same experiment on the ISMIR2004Genre dataset with 64 ASMs. For TSPs with short-term temporal information (e.g., 1-TSP and 2-TSP), we also observed a similar phenomenon where a larger α leads to an increasing number of TSPs. On the other hand, for the TSPs with long-term temporal information (e.g., 3-TSPs, 4-TSPs, etc), the numbers of TSPs in Jazz/Blues, Metal/Punk, and Classical also rank top three out of the six genres. This indicates that TSPs with different lengths can indeed capture the characteristics of each genre. More specifically, the more repeating pattern in a

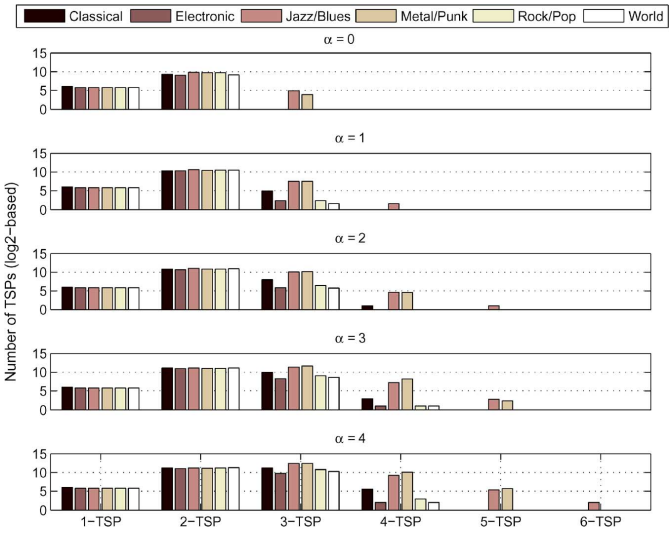


Fig. 12. Numbers of TSPs mined from six genres under 64 ASMs and different values of α on the ISMIR2004Genre dataset. (Note that we have added 1 to the original average numbers to avoid minus infinity after taking logarithm.)

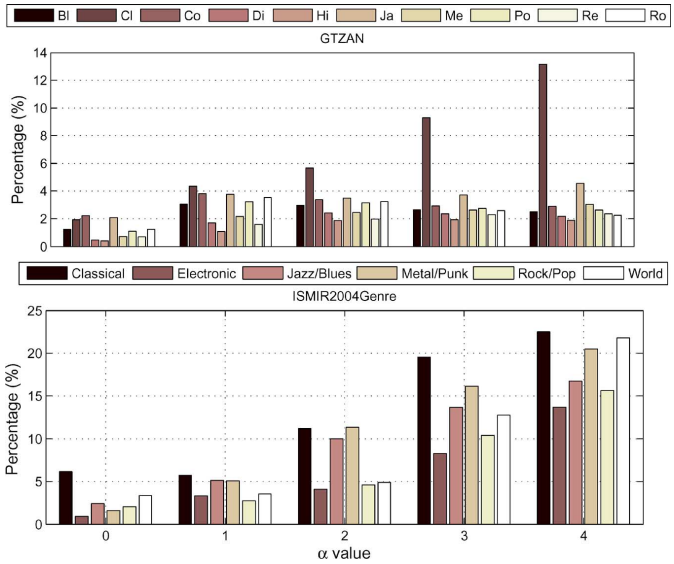


Fig. 13. Percentages of TSPRT with respect to TSP within different genres under different values of α on the GTZAN (with 128 ASMs) and ISMIR2004Genre (with 64 ASMs) datasets.

music piece of a genre leads to the discovery of more TSPs with longer lengths.

In summary, the use of TSPs can discover more long-term temporal information than that of unigram and bigram, which can effectively reflect the characteristics of each genre. Moreover, it should be noted that when α is 0, the average number of 2-TSP in each genre is less than 2^{14} (see Fig. 11), which is less than the number of bigrams $(128 \times 128 = 2^{14})$ in the baseline. However, the proposed method achieves a better performance, indicating that the selected 2-TSPs, though less in quantity, still have better discriminative power than the baseline.

E. TSPs Analysis—Repeating Tokens

In addition to long-term temporal information, repeating tokens within TSPs of same-genre music are also important cues for classification. A TSP which contains repeating tokens is here denoted as TSPRT (TSP with Repeating Tokens). For instance, a 3-TSP (x41, x20, x41) is a TSPRT since the first token and the last one are the same. For simplicity, we use p_{genre} to represent the percentage of TSPRT with respect to TSP within a given genre.

Fig. 13 shows the values of p_{genre} for 10 genres under different value of α on the GTZAN dataset (top panel, with 128 ASMs) and the ISMIR2004Genre dataset (bottom panel, with 64 ASMs). For both datasets, it is obvious that p_{Classic} is higher than the percentages of the other genres. This reflects that there is a high percentage of TSPRTs within the Classical music pieces, which is intuitive since Classical music is likely to have more repeating phrases than others. Similar phenomenon occurs to Jazz/Blues.

In summary, the use of TSPs can capture the music characteristics of a specific genre (in particular, the repeating nature of Classical music), leading to better performance than the baseline.

F. Comparisons to Other Existing Approaches

In addition to comparing the proposed method with the baseline, Table III also lists other approaches evaluated on both datasets (for the GTZAN dataset, all the accuracies were evaluated via 10-fold cross-validation). Before comparing our approach with these approaches in terms of accuracy, we first give a brief description of these methods. Tzanetakis and Cook [4] used short-time features (e.g., timbre, rhythm, and pitch) and trained GMM classifiers. Lidy and Rauber [10] extracted features (including rhythm patterns, a statistical spectrum descriptor, and rhythm histograms), and then employed pairwise SVMs for classification. Li and Ogihara [6] used DWCHs as features and SVMs as a set of classifiers. Panagakis *et al.* [12] first extracted spectro-temporal modulation features; and then applied multilinear subspace techniques, namely non-negative tensor factorization (NTF), high-order singular value decomposition (HOSVD), and multilinear principle component analysis (MPCA) to derive compact feature vectors; and finally trained SVMs for classification. This work was further improved by Panagakis *et al.* [14], which derived non-negative multilinear principle component analysis (NMPCA) to extract more discriminating features for SVMs or nearest neighbor classifiers. Benetos and Kotropoulos [43] used a variety of short-time and long-term features (e.g., spectral, temporal, perceptual, energy, and pitch descriptors), and then trained NTF bases for classification. Panagakis and Kotropoulos [15] employed topological preserving non-negative tensor factorization (TPNTF) on spectro-temporal modulation features, and then classified these reduced dimensionality features via sparse representation based classification (SRC) [44].

Here classification accuracy is used to compare the performance among different approaches. For the GTZAN dataset, our performance is better than most of these approaches. For the ISMIR2004Genre dataset, our performance is comparable to Panagakis *et al.*'s approach [12] which ranked No. 3. Note that here we did not evaluate the statistical significance of these accuracies since we do not have the original test results of these approaches. Instead, we evaluated the significant differences of the proposed method and the top-3 methods submitted to

TABLE III
COMPARISONS WITH OTHER APPROACHES ON BOTH DATASETS

Reference	Dataset	Accuracy
Panagakis and Kotropoulos [15]	GTZAN	93.7%
Panagakis *et al.* [14]	GTZAN	84.3%
Our approach	GTZAN	81.7%
Benetos and Kotropoulos [43]	GTZAN	78.9%
Li and Ogihara [6]	GTZAN	78.5%
Panagakis *et al.* [12]	GTZAN	78.2%
Lidy and Rauber [10]	GTZAN	74.9%
Tzanetakis and Cook [4]	GTZAN	61.0%
Panagakis and Kotropoulos [15]	ISMIR2004Genre	94.9%
Pampalk [40] (winner)	ISMIR2004Genre	84.0%
Panagakis *et al.* [12]	ISMIR2004Genre	80.9%
Our approach	ISMIR2004Genre	79.7%
Lidy and Rauber [10]	ISMIR2004Genre	79.7%
West [40] (2nd)	ISMIR2004Genre	78.3%
Tzanetakis [40] (3rd)	ISMIR2004Genre	71.3%

TABLE IV
AVERAGE COMPUTATION TIME OF ONLINE TASKS FOR GTZAN VERSUS ISMIR2004GENRE UNDER 128 ASMS AND $\alpha = 4$ (IN SECOND)

Online task	Computation time
Feature extraction	0.15 vs. 0.15
Viterbi decoding	1.42 vs. 1.40
Computation of TSP co-occurrence	1.10 vs. 0.70
Feature weighting	0.03 vs. 0.02
SVM classification	0.45 vs. 0.32
Total time for online processing	3.15 vs. 2.59

the *ISMIR2004 Genre Classification Contest* (of which the results are available at the MIREX official page). The result of Friedman's test shows that Pampalk [40] (the winner of this contest) does *not significantly* outperform the proposed method at the 5% level.

G. Computation Complexity (Computation Time)

In addition to comparing the classification accuracies of the proposed system with other existing approaches, here we also show that the proposed system is efficient in terms of computation time. As shown in Fig. 2, most lengthy processes of the system construction, e.g., ASM training, temporal sequence generation (including feature extraction and Viterbi decoding), genre-specific TSP mining, TSP co-occurrence statistics, feature weighting, and SVM training, can be performed offline. The two main tasks involved in online operations are the Viterbi decoding for generating a test music transcript and the computation of the co-occurrence statistics of TSPs for the generated transcript. More specifically, under 128 ASMs and $\alpha = 4$, the average computation time of online tasks for classifying music pieces of these two datasets (GTZAN versus ISMIR2004Genre) is shown in Table IV.[1]

V. CONCLUSION AND FUTURE WORK

In this paper, we propose the use of time-constrained sequential patterns (TSPs) for music genre classification. First of all, we perform spoken language identification technique to tokenize each music piece into a sequence of ASM indices. Next, we apply TSP mining to discover genre-specific TSPs. Finally, by representing each music piece as the weighted occurrence frequencies of all the mined TSPs, linear SVM classifiers are employed to facilitate the classification task. From the analysis of TSPs' lengths and counts, we found that TSPs with different lengths/counts can be used to capture the characteristics of specific genres. For instance, there tend to be more and longer TSPs in music pieces of Metal than music pieces of other genre. This is intuitive since Metal is traditionally characterized by dense Jazz and drum sound. Moreover, the accuracy of the proposed method is approximately 2% to 6% higher than that of the text-categorization-based approach, which can be considered as a basic scheme based on music tokenization.

One possible future direction of this study is to use a metric (e.g., information gain) to remove non-discriminative TSPs first, and then apply the same method for feature extraction, classifier design, and evaluation (Sections III-C and III-D). In addition, to tackle the problem of ASM training on an unbalanced dataset, a possible solution is to obtain a set of genre-specific ASMs first, and then use all these trained ASMs to tokenize all music pieces for classification.

[1]The simulation platform is AMD 2.30 GHz CPU, 4G RAM, Windows 7 64-bit OS, MATLAB 2008b, and HTK toolkit v3.3.

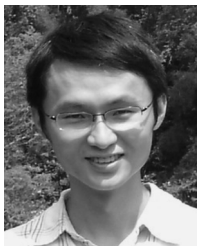
ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments so as to improve the presentation and quality of this paper. The authors would also like to thank Dr. Jeremy Reed for the discussion of the implementation of ASMs.

REFERENCES

[1] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *J. New Music Res.*, pp. 83–93, 2003.
[2] S. Lippens, J. P. Martens, M. Leman, B. Baets, H. Meyer, and G. Tzanetakis, "A comparison of human and automatic musical genre classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, pp. 233–236.
[3] L. Karydis, A. Nanopoulos, and Y. Manolopoulos, "Symbolic music genre classification based on repeating patterns," in *Proc. ACM Multimedia*, 2006, pp. 53–57.
[4] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
[5] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proc. ACM SIGIR*, 2003, pp. 282–289.
[6] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 564–573, Jun. 2006.
[7] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proc IEEE Int. Conf. Multimedia and Expo.*, 2002, vol. 1, pp. 113–116.
[8] K. West and S. Cox, "Features and classifiers for the automatic classification of musical audio signals," in *Proc. Int. Symp. Music Inf. Retrieval*, 2004, pp. 531–536.
[9] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. Int. Symp. Music Inf. Retrieval*, 2003, pp. 151–158.
[10] T. Lidy and A. Rauber, "Evaluation of feature extractors and psychoacoustic transformations for music genre classification," in *Proc. Int. Symp. Music Inf. Retrieval*, 2005, pp. 34–41.
[11] J. Bergatra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl, "Aggregate features and Adaboost for music classification," *Mach. Learn.*, vol. 65, no. 2–3, pp. 473–484, Jun. 2006.
[12] I. Panagakis, E. Benetos, and C. Kotropoulos, "Music genre classification: A multilinear approach," in *Proc. Int. Symp. Music Inf. Retrieval*, 2008, pp. 583–588.
[13] C.-H. Lee, J.-L. Shih, K.-M. Yu, and H.-S. Lin, "Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features," *IEEE Trans. Multimedia*, vol. 11, no. 4, pp. 670–682, Jun. 2009.

[14] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 576–588, Mar. 2010.
[15] Y. Panagakis and C. Kotropoulos, "Music genre classification via topology preserving non-negative tensor factorization and sparse representations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 249–252.
[16] M. Casey and M. Slaney, "The importance of sequences in musical similarity," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2006, pp. V-5–V-8.
[17] M. Li and R. Sleep, "Genre classification via an LZ78-based string kernel," in *Proc. Int. Symp. Music Inf. Retrieval*, 2005, pp. 252–259.
[18] M. Li and R. Sleep, "A robust approach to sequence classification," in *Proc. IEEE Int. Conf. Tools with Artif. Intell.*, 2005, pp. 197–201.
[19] T. Langlois and G. Marques, "A music classification method based on timbral features," in *Proc. Int. Symp. Music Inf. Retrieval*, 2009, pp. 81–85.
[20] K. Chen, S. Gao, Y. Zhu, and Q. Sun, "Music genre classification using text categorization method," in *Proc. MMSP*, 2006, pp. 221–224.
[21] J. Paulus and A. Klapuric, "Music structure analysis by finding repeated parts," in *Proc. ACM Multimedia*, 2006, pp. 59–67.
[22] A. G. Krishna and T. V. Sreenivas, "Music instrument recognition: From isolated notes to solo phrases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2004, pp. 265–268.
[23] J. Reed and C.-H. Lee, "A study of music genre classification based on universal acoustic models," in *Proc. Int. Symp. Music Inf. Retrieval*, 2006, pp. 89–94.
[24] A. Sheh and D. P. W. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proc. Int. Symp. Music Info. Retrieval*, 2003, pp. 183–189.
[25] J.-M. Ren, Z.-S. Chen, and J.-S. R. Jang, "On the use of sequential patterns mining as temporal features for music genre classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 2294–2297.
[26] J.-M. Ren and J.-S. R. Jang, "Time-constrained sequential pattern discovery for music genre classification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 173–176.
[27] L. Lu, M. Wang, and H.-J. Zhang, "Repeating pattern discovery and structure analysis from acoustic music data," in *Proc. ACM MIR*, 2004, pp. 275–282.
[28] S.-C. Chiu, M.-K. Shan, J.-L. Huang, and H.-F. Li, "Mining polyphonic repeating patterns from music data using bit-string based approaches," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, 2009, pp. 1170–1173.
[29] M.-Y. Lin, S.-C. Hsueh, M.-H. Chen, and H.-Y. Hsu, "Mining sequential patterns for image classification in ubiquitous multimedia systems," in *Proc. IEEE Int. Conf. Intell. Info. Hiding Multimedia Signal Process.*, 2009, pp. 303–306.
[30] J.-G. Lee, J. Han, X. Li, and H. Cheng, "Mining discriminative patterns for classifying trajectories on road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 5, pp. 713–726, Jun. 2011.
[31] T. P. Exarchos, M. G. Tsipouras, C. Papaloukas, and D. I. Fotiadis, "A two-stage methodology for sequence classification based on sequential pattern mining and optimization," *Data Knowl. Eng.*, vol. 66, pp. 467–487, 2008.
[32] B. Ma, H. Li, and C.-H. Lee, "An acoustic segment modeling approach to automatic language identification," in *Proc. Interspeech*, 2005, pp. 2829–2831.
[33] S. Dixon, "Onset detection revised," in *Proc. Int. Conf. Digital Audio Effects*, 2006, pp. DAFX-1–DAFX-6.
[34] M.-Y. Lin, S.-C. Hsueh, and C.-W. Chang, "Fast discovery of sequential patterns in large databases using effective time indexing," *Inf. Sci.*, vol. 178, no. 22, pp. 4228–4245, 2008.
[35] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proc. Int. Conf. Mach. Learn.*, 1997, pp. 412–420.
[36] H. Cheng, X. Yan, J. Han, and C.-W. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Proc. IEEE Int. Conf. Data Eng.*, 2007, pp. 716–725.
[37] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Inf. Process. Manage.*, vol. 24, no. 5, pp. 513–523, 1988.
[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machine," 2010 [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm
[39] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed. Waltham, MA: Morgan Kaufmann, 2006.
[40] [Online]. Available: http://ismir2004.ismir.net/genre_contest/index.htm
[41] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Dept. of Comput. Sci., National Taiwan Univ., 2003, Tech. Rep.
[42] M. L. Berenson, D. M. Levine, and M. Goldstein, *Intermediate Statistical Methods and Applications: A Computer Package Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[43] E. Benetos and C. Kotropoulos, "Non-negative tensor factorization applied to music genre classification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 1955–1967, Nov. 2010.
[44] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 210–227, Feb. 2009.



Jia-Min Ren (S'08) was born in Taiwan, in 1982. He received the B.S. and M.S. degrees in information management from National Formosa University, Yunlin County, Taiwan, in 2005 and 2007, respectively. He is currently pursing the Ph.D. degree in computer science at National Tsing Hua University, Hsinchu, Taiwan.

His research interests include data mining, digital watermarking, semantic analysis of musical signals, and music information retrieval.

Jyh-Shing Roger Jang (M'93) was born in Taiwan in 1962. He received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 1984, and the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1992.

He worked for MathWorks, Inc., from 1993 to 1995, and coauthored the *Fuzzy Logic Toolbox*. Since 1995, he has been with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. He has published two books: *Neuro-Fuzzy and Soft Computing* (Prentice-Hall, 1997) and *MATLAB Programming* (CWeb, 2004, in Chinese). His research interests include speech recognition/synthesis, melody recognition, singing voice synthesis, face detection/recognition, pattern recognition, neural networks, and fuzzy logic.