Genre Classification for Musical Documents Based on Extracted Melodic Patterns and Clustering

Bor-Shen Lin
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
bslin@cs.ntust.edu.tw

Tai-Cheng Chen
Department of Information Management
National Taiwan University of Science and Technology
Taipei, Taiwan
m9909106@mail.ntust.edu.tw

***Abstract*—Genre classification for musical documents is conventionally based on keywords, statistical features or low-level acoustic features. Such features are either lack of in-depth information of music content or incomprehensible for music professionals. This paper proposed a classification scheme based on the correlation analysis of the melodic patterns extracted from music documents. The extracted patterns can be further clustered, and smoothing techniques for the statistics of the patterns can be utilized to improve the performance effectively. The accuracy of 70.67% for classifying five types of genre, including jazz, lyric, rock, classical and others, can be achieved, which outperforms an ANN-based classifier using statistical features significantly. The patterns can be converted into symbolic forms such that the classification results are meaningful and comprehensible for most music workers.**

Keywords- Genre Classification; Automatic Tagging; N-gram Melodies; Repeated Melodic Patterns; Music Information Retrieval

I. INTRODUCTION

Genre classification has been an important issue in automatic tagging for music information retrieval [1] because genre is a popular feature for categorizing music in almost any music playing software or web stores of music. If the genres of the music works can be automatically labeled, they can be used not only for music retrieval but for music analysis. In the past, the genres of music documents could be classified based keywords, statistical features or acoustic features [2-4]. Keywords are lack of the information of music content, and cannot achieve good discriminative capabilities among songs. Statistical features, such as the features about the number of instruments, the average tempo and pitch, and so on, contain the statistical information of music content, but lose in-depth messages in the melodies. Acoustic features, such as MFCC, can provide low level information of real performance, and genre classifiers based on such features can usually achieve good performance. However, the messages conveyed in acoustic features are not in symbolic form that can be easily interpreted by music professionals. As a consequence, though the classification accuracy is acceptable, it is difficult for music professionals to understand or analyze the results.

This paper proposed a classification scheme based on the correlation analysis of the melodic patterns extracted from music documents [5-8]. The extracted patterns can be further clustered, and the smoothing techniques for the statistics of the patterns can then be utilized to improve the performance effectively. The accuracy of 70.67% for classifying five types of genre, including jazz, lyric, rock, classical and others, can be achieved, which outperforms an ANN-based classifier using statistical features significantly. The patterns can be converted into symbolic forms such that the results are meaningful and comprehensible for most music workers.

II. EXTRACTION OF MELODY PATTERNS

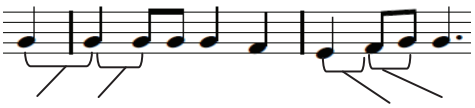
First, how the notes in musical documents are represented and encoded is introduced. The extraction and filtering of melodic patterns are then illustrated respectively, while distance measures are finally depicted.

A. Note Encoding

To extract the repeated melodic patterns, every musical document need first be transcribed into a sequence of notes consisting of pitch and duration. Assume the pitch and duration for a note in a sequence at position i are denoted as p_i and d_i, respectively. In order to make the representation of the melodies independent of pitch translation or relative speed, the pitch and duration are further encoded through a delta-logarithm operation as defined below.

$$p_i' = (\log_2(p_i) - \log_2(p_{i-1})) \cdot 12$$
$$d_i' = \log_2(d_i) - \log_2(d_{i-1}) \quad (1)$$

In this formula, the delta-logarithm of pitch is measured in the unit of semi-note, since every octave can be equally divided into 12 semi-notes. When $p_i = 2^{3/12} p_{i-1}$, for example, p_i' is 3, which signifies that the interval between p_{i-1} and p_i is minor three. When $p_i = p_{i-1}$, p_i' becomes 0, which signifies that p_{i-1} and p_i are of the same pitch. In this way, a sequence of encoded notes, $\{\boldsymbol{n_i}\}$, where $\boldsymbol{n_i}$ equals the pair (p_i', d_i'), can be used to represent a melody with relative pitch and speed. Fig. 1 shows an example of encoded output for a melody using this encoding scheme.



(0 0) (0 -1) (0 0) (0 1) (-2 0) (-1 0) (1 -1) (2 0) (0 2)

Figure 1. Encoding a melody as a sequence of notes.

978-0-7695-4919-4/12 $26.00 © 2012 IEEE
DOI 10.1109/TAAI.2012.23

B. Prefix Tree

For every musical document represented as a sequence of notes, the subsequences with length less than a threshold are extracted and inserted to a prefix tree of patterns such that the occurrence counts of the n-grams patterns could be accumulated, as illustrated in the simplified example of Fig. 2. The threshold is set as 11 here in this paper because the patterns with length larger or equal to 10 are rare. Fig. 2(a) shows a sequence "ABBAC" and the corresponding subsequences, while Fig. 2(b) displays the prefix tree built from them. Notice that the symbols A, B and C signify the encoded notes ($\boldsymbol{n_i}$'s) consisting of pitch and duration which need to be matched while building the tree. The comparison function could be defined flexibly, such as matching the pitch only, matching the duration only, or matching both, according to which features are of interest.

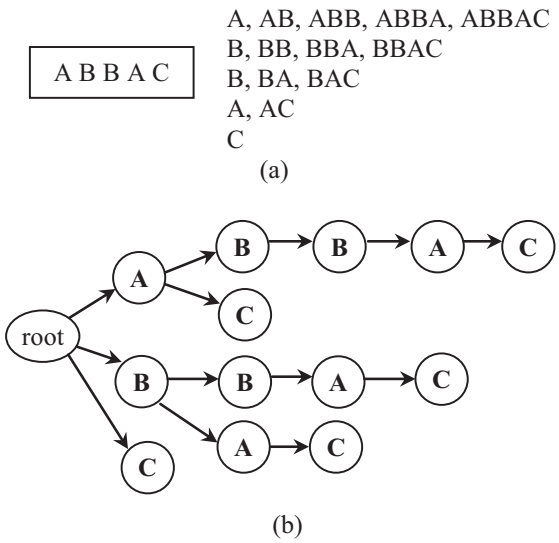


Figure 2. Procedures for building the prefix tree of n-gram patterns (a) producing the subsequences for a note sequence "ABBAC" (b) inserting the subsequences into the tree.

The prefix tree for the melodic patterns is similar to conventional lexical tree for storing a dictionary, but here each tree node stores a note object instead of a character. At the terminal node of a pattern, such as the ending node A of the pattern A-B-B-A, the occurrence count of that pattern is accumulated, and used for pattern filtering later on.

C. Pattern Filtering

In a musical document, the note sequence is in general long, so the combinations of subsequences might be very huge, while only a few of them are redundant. If the occurrence count of a pattern A-B-C-B-C is high, for example, the counts of the subsequences A-B-C-B and A-B-C will also be high, but the latters might not be discriminative patterns. Hence, in case the occurrence count of a pattern is equal to that of its path extension on the tree as described above, that pattern will be removed during filtering. Besides, some filtering criteria are further used to reject less desired patterns, as described below.

- The occurrence count is larger than 10.
- The length of the pattern is higher than 3.
- The number of sources is at least 2.

The first criterion in the list is used because those patterns occurring frequently tend to be important patterns, while the second because the shorter patterns often occur but have poor discriminative capabilities. The third criterion is because, the task in this paper is to perform genre classification for music documents by making use of the melodic similarities across documents, so the patterns occurring only in single document are ignored. Furthermore, the accepted patterns are sorted based on the inverse document frequency (IDF) which could indicate the relevance of the pattern. N-best patterns with the highest values of IDFs are finally extracted and used for further analysis. The set of melodic patterns is denoted as $\{\boldsymbol{m_l}\}$ where $\boldsymbol{m_l}$ is the l-th melodic pattern.

D. Distance Measures

Before applying the clustering algorithm, the metric of distance or similarity between two patterns need first be defined. Since the patterns signify the note sequences whose lengths might change, it is hence required for the distance metric to align two sequences of different lengths. Dynamic time warping (DTW) is such an algorithm, and is therefore utilized to measure the distance of two melodic patterns [9,10]. Fig. 3 shows an example of aligning two patterns $\mathbf{m}_i$ and $\mathbf{m}_j$ in DTW, where w stands for the maximum number of notes in one pattern that can be matched to a note in the other pattern. The distance between the two patterns is the minimum path distance achievable among all the paths that satisfy the search constraints. The distance of the optimal aligned path need further be normalized by the length of the path so as to fall onto the range between 0 and 1 for measuring the ratio of different notes in the two melodic patterns. With the distance measure defined for two patterns, it is then feasible to measure the distance between two music documents (songs), say S_m and S_n, as below since every music document consists of a set of patterns.

$$d(S_m, S_n) = \frac{1}{|S_m||S_n|}\sum_{\mathbf{m}_i \in S_m, \mathbf{m}_j \in S_n} d(\mathbf{m}_i, \mathbf{m}_j) \qquad (2)$$

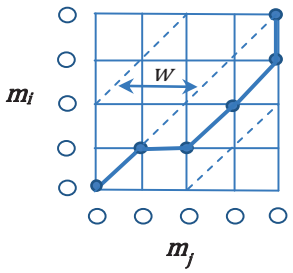


Figure 3. Dynamic time warping for two patterns.

III. CALSSIFICATION SCHEMES

Based on the extracted melodic patterns, the statistical correlation between the tag and the pattern can be further analyzed, as given below.

A. Correlation Analysis

For two binary discrete random variables, A and B, with joint probability mass $P(A,B) = r$ and marginal functions $P(A) = p$ and $P(B) = q$, it can be derived that $P(A,\bar{B}) = p - r$, $P(\bar{A},B) = q - r$, and $P(\bar{A},\bar{B}) = 1 - p - q + r$. The correlation coefficient between A and B is then

$$\rho_{AB} = \frac{E(AB)-E(A)E(B)}{\sqrt{var(A)var(B)}} = \frac{r-pq}{\sqrt{p(1-p)q(1-q)}}, \quad (3)$$

where $E(\cdot)$ and $var(\cdot)$ denote respectively the expectation and the variance for the random variables. Let random variable A stand for the tag and random variable B stand for the melodic pattern. The correlation coefficient between the tag and the pattern can then be computed according to their statistics in the musical documents. Assume t_k be the k-th tag and $\mathbf{m}_l$ be the l-th pattern, respectively. The probability $P(t_k)$ can then be estimated with the number of documents with tag t_k divided by the total number of documents. The probability $P(\boldsymbol{m}_l)$ can be estimated with the number of documents containing the pattern m_l divided by the total number of documents. The probability $P(t_k, \boldsymbol{m}_l)$, on the other hand, is estimated with the ratio of documents that have the tag t_k and contain the pattern $\boldsymbol{m}_l$. With $P(t_k)$, $P(\boldsymbol{m}_l)$ and $P(t_k, \boldsymbol{m}_l)$, the correlation coefficient $\rho(t_k, \boldsymbol{m}_l)$ can be computed according to Eq. (3).

B. Pattern Clustering

Though the number of patterns can be reduced through pattern filtering, for statistical analysis the problem of data sparseness might arise when the amount of training samples is insufficient relatively. Hence, here in this paper the agglomerative clustering [11] is proposed to group the patterns into clusters such that smoothing techniques based on the clusters could be applied.

During the bottom-up process of agglomerative clustering, the distances between cluster pairs need be computed in order to decide two closest clusters to be merged among all cluster pairs. Here in this paper that distance between two clusters is defined as the maximum distance among all distances of pair-wise patterns respectively in the two clusters, as shown in Fig. 4(a). With the cluster distance defined as such, agglomerative clustering can then be performed for the patterns to obtain a *dendrogram* that signifies the hierarchy of clusters based on the distances, as shown in Fig. 4(b). Different number of clusters can be obtained flexibly by simply shifting the threshold of distance. When the threshold of distance is lowered, the number of clusters becomes large and the patterns in every cluster tend to be close, and vice versa. The number of clusters is an adjustable parameter for optimizing the ultimate system performance, which refers to the classification accuracy in this paper. Given the clusters, every pattern, say $\boldsymbol{m}_l$, is uniquely assigned to a cluster, say c_j. Such statistics as $P(c_j)$, and $P(t_k, c_j)$ can all be computed, based on which the correlation coefficient $\rho(t_k, c_j)$ between the tag t_k and the cluster c_j can also be obtained easily.

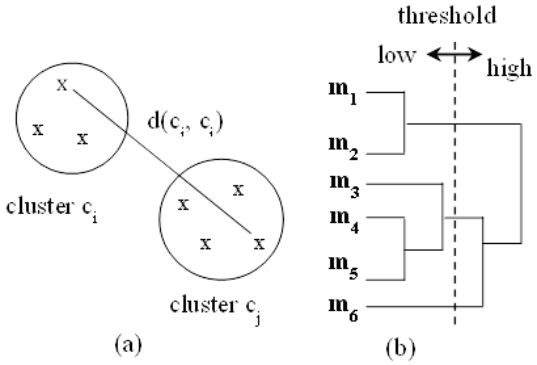


Figure 4. Illustrative example of agglomerative clustering for extracted patterns. (a) the distance between two clusters (b) an example *dendrogram* and adjustment of the threshold.

C. Correlation-based Classifier and Smoothing Methods

Since every musical document contains a set of melodic patterns, it is feasible to compute the correlation between the document, say S, and the tag, say t_k, by summing the correlations for the patterns in the document,

$$\rho(S, t_k) = \sum_{\boldsymbol{m}_l \, in \, S} \rho(\boldsymbol{m}_l, t_k). \quad (4)$$

Assume there are K possible tags, t_k, $k = 1,2, \ldots, K$, and the document S needs to be assigned to exactly one of the tags. Then, the optimal genre can be determined by

$$k^* = argmax_k \, \rho(S, t_k). \quad (5)$$

When the amount of training data is insufficient, the problem of data sparseness might happen. In such case it may be helpful to use the correlation of the cluster instead of that of the pattern, since any pattern within a cluster may share the statistics from other patterns. The correlation between the document S and the tag t_k can then be obtained by summing the correlations for the clusters of the patterns within the document.

$$\rho(S, t_k) = \sum_{\boldsymbol{m}_l \, in \, S, \boldsymbol{m}_l \in c_j} \rho(c_j, t_k). \quad (6)$$

where c_j is the cluster that the pattern $\boldsymbol{m}_l$ is assigned to. The genre type of the document S can then be decided according to Eq. (5) by applying the smoothing approach in Eq. (6).

However, if the problem of data sparseness is not serious, using Eq. (6) unconditionally might degrade the classification performance contrarily because the statistics for the pattern $\boldsymbol{m}_l$ may be well trained and the correlation obtained from Eq. (4) is more precise. To compromise between Eq. (4) and Eq. (6), a scheme of selective smoothing is proposed. The idea is based on the concept that, it is better to use $\rho(\boldsymbol{m}_l, t_k)$ when the training is sufficient relatively,

and to use $\rho(c_j, t_k)$ otherwise since $\boldsymbol{m_l} \in c_j$. It might be possible to achieve better performance provided the smoothing strategy can be adjusted dynamically according to the sufficiency of training data for the pattern. In order to measure the sufficiency, the occurrence counts for all patterns are sorted and the ranking ratio for each pattern, defined as the ratio for the order of the occurrence count of a pattern among all patterns, can be obtained. For example, if the occurrence count of a pattern is at top 10 among 1000 patterns after sorting, its ranking ratio is 0.01. Lower ranking ratio for a pattern implies the pattern has more sufficient training data relatively. Therefore, the smoothing strategy becomes not to perform smoothing when the ranking ratio of pattern $\boldsymbol{m_l}$ (i.e. $\mathrm{r}(\boldsymbol{m_l})$) is lower than a threshold r, which is between 0 and 1.

$$\rho(S, t_k) = \sum_{m_l \, in \, S} \hat{\rho}(\mathbf{m}_l, t_k)$$

$$\hat{\rho}(\mathbf{m}_l, t_k) = \begin{cases} \rho(\mathbf{m}_l, t_k) & if \ \mathrm{r}(\mathbf{m}_l) < \mathrm{r} \\ \rho(c_j, t_k) & otherwise \end{cases} \quad (7)$$

This strategy is named as selective smoothing in later analysis, and can be regarded as interpolation between Eq. (4) and Eq. (6), which correspond to r=1.0 and r=0.0, respectively.

D. ANN/KNN Classifiers

The genre classification with extracted melodic patterns can also be realized using such classification schemes as artificial neural network (ANN) or k-nearest neighbor (KNN). It is similar to document classification with keywords, but here the patterns play the roles of the keywords. For ANN-based classification scheme in form of multiple-layer perceptrons, the number of neurons in the input layer is equal to the number of patterns, while that in the output layer is equal to the number of genre types. The index of the output neuron with maximum value is the output genre. For KNN-based classification scheme, k-nearest documents with the label of genre in the training set are picked for the query document, and a majority vote among the k documents is conducted to decide the genre of the query document. The distance between two documents for picking the k-nearest documents is computed using Eq. (2).

IV. Experiments And Analysis

2187 MIDI files containing songs in Taiwan area in recent years were collected, and the tracks containing the main melodies were extracted semi-automatically and converted into digital scores in Lilypond format. The genres for all documents were manually labeled by two music professionals with five types of genre: jazz, lyric, rock, classical and others. The distribution of songs with respect to the genre types is shown in Table 1. From the digital scores, the prefix tree for melodic n-gram patterns was built and totally 10,000 patterns were finally extracted for experiments.

First, the experiments of genre classification for the classification schemes described in Section III were conducted, and the results are shown in Table 2. The correlation-based classifier, here denoted as COR, as illustrated in Section III.A through III.C can achieve the average accuracy of 66.20%, which is better than that of the KNN-based classifier but a little worse than that of the ANN-based classifier. Notice that the correlation-based classifier here is based on Eq. (4) with no smoothing technique applied. It can be observed in Table 2, the classification performances for the genre types of *jazz* and *classical* are relatively worse because of fewer training data, while the genre type of *lyric* with the most training data achieves the highest accuracy.

Table 1. Distribution of songs for various genre types

Genre	Sum	Training	Testing	Ratio
Jazz	100	60	40	4.6%
Lyric	1,050	630	420	48%
Rock	450	270	180	20.6%
Classical	35	21	14	1.6%
Others	552	331	221	25.2%
Total	2,187	1,312	875	100%

Table 2. Comparison of classification performance for various classification schemes

Genre	COR	ANN	5-KNN
Jazz	26%	27%	23%
Lyric	78%	80%	76%
Rock	45%	43%	40%
Classical	38%	40%	35%
Others	55%	55%	52%
Average	66.20%	67.10%	64.98%

When the clustering algorithm was applied and Eq. (6) was used for smoothing, the performance of the correlation-based classifier can be improved significantly from 66.20% to 70.00%, as shown in Table 3. This implies that the smoothing approach by sharing the statistics of the patterns through clusters can alleviate the problem of data sparseness effectively. It can also be seen in Table 3, the improved performance is obtained when the number of clusters is 7,000. Some auxiliary experiments show that, when the number of cluster is further reduced to lower than 7,000 through the aggregations of clusters, over-smoothing starts to occur and the performance decreases persistently, since the patterns within a cluster might not be close enough to one another.

Table 3. Comparison of classification performance with or without clustering/smoothing.

Genre	Without clustering	With clustering (7,000 clusters)
Jazz	26%	32%
Lyric	78%	80%
Rock	45%	49%
Classical	38%	40%
Others	55%	57%
Average	66.2%	70.00%

As the selective smoothing depicted in Eq. (7) was applied, the accuracy can be further increased by adjusting the threshold of ranking ratio r, as depicted in Fig. 5. The optimal performance of 70.67% can be obtained at r = 0.2. This result shows that to smooth the statistics selectively with Eq. (7) (corresponding to r=0.2) is better than to smooth them unconditionally with Eq. (6) (corresponding to r=0.0). Both are better than the case without smoothing (corresponding to r = 1.0).

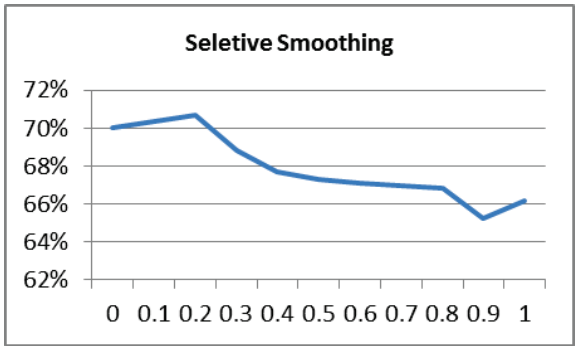


Figure 5. Selective smoothing with respect to the threshold of ranking ratio r for the patterns.

Finally, the experiment for genre classification was conducted for another ANN-based classifier using statistical features obtained from the MIDI files to replace the extracted patterns $\{\boldsymbol{m}_i\}$. The statistical features used here include the number of instruments, the tempo, the dynamic range of pitches, and the ratios of quarter notes, eighth notes and sixteenth notes. The experimental results were compared with those of the correlation-based classifier and shown in Table 4. It can be seen in this table that, the correlation-based classifier with extracted patterns can outperform the ANN-based classifier with statistical features. This is probably because the melodic patterns contain in-depth information of music content that is more informative and discriminative than typical statistical features that are simply the macro views of the music contents.

Table 4. Comparison between the proposed approach and the ANN classifier with statistical features.

	COR with selective smoothing	ANN classifier
Features	melodic patterns	statistical features
Jazz	31%	20%
Lyric	81%	70%
Rock	53%	32%
Classical	38%	5%
Others	59%	34%
Average	70.67%	57.63%

V. CONCLUSIONS

This paper proposed a classification scheme based on the correlation analysis of the melodic patterns extracted from music documents. The extracted patterns can be further clustered, and the smoothing techniques for the statistics of the patterns can then be utilized to improve the performance effectively. The accuracy of 70.67% for classifying five types of genre, including jazz, lyric, rock, classical and others, can be achieved, which outperforms an ANN-based classifier using statistical features significantly. The patterns can be converted into symbolic forms easily, so the results are meaningful and comprehensible for most music workers.

ACKNOWLEDGMENT

This research is under the support of NSC project No. NSC 99-2221-E-011 -129 from Taiwan government.

REFERENCES

[1] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-Based Music Information Retrieval: Current Directions and Future Challenges," Proceedings of the IEEE, Vol. 96, No. 4, pp. 668–696, 2008.

[2] M. Grimalidi, A. Kokaram, and P. Cunningham, "Classifying Music by Genre Using a Discrete Wavelet Transform And a Round-robin Ensemble," Technical report TCD-CS-2002-64, Trinity College, University of Dublin, Ireland, 2002.

[3] K. Koshina, "Music genre recognition," Diploma Thesis, Technical College of Hagenberg, Austria. 2002.

[4] G. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," IEEE Transactions on Speech and Audio Processing 10 (5).293-302. 2002.

[5] J. L. Hsu, C. C. Liu, and A.L.P. Chen: "Discovering Non-trivial Repeating Patterns in Music Data," IEEE Transactions on Multimedia, 2001.

[6] J.L. Hsu, A.L.P. Chen, and H.C. Chen: "Finding Approximate Repeating Patterns from Sequence Data," Proceedings of International Symposium on Music Information Retrieval, 2004.

[7] D. Meredith, K. Lemstrom and G.A. Wiggins, "Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music," Journal of New Music Research, vol. 31, no. 4, pp. 321-345, 2002.

[8] A. Uitdenbogerd and J. Zobel. "Melodic Matching Techniques for Large Music Databases," Proceedings of the ACM Multimedia Conference, 1999.

[9] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-26,no. 1, pp.43-49, 1978.

[10] W. Chai: "Melody as a Significant Musical Feature in Repertory Classification," Music Query: Methods, Models, and User Studies, Computing in Musicology, Vol. 13, the MIT Press, 2004.

[11] Mehmed Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms, pp. 117-135, Wiley Inter-Science, 2003.