Clustering Music Recordings Based on Genres*

WEI-HO TSAI AND DUO-FU BAO

Department of Electronic Engineering
Institute of Computer and Communication Engineering
National Taipei University of Technology
Taipei, 106 Taiwan

Existing systems for automatic genre classification follows a supervised framework that extracts genre-specific information from manually-labeled music data and then identifies unknown music data. However, such systems may not be suitable for personal music management, because manually labeling music based on individually-defined genres can be labor intensive and subject to inconsistency from time to time. In this paper, we study an unsupervised paradigm for music genre classification. It is aimed to partition a collection of unknown music recordings into several clusters such that each cluster contains recordings in only one genre, and different clusters represent different genres. This enables users to organize their personal music database without needing specific knowledge about genre. This study investigates how to measure the genre similarities between music recordings and estimate the number of genres in a music collection. Our experiment results show the feasibility of clustering music recordings by genre.

Keywords: music genre, purity, Rand index, supervised classification, unsupervised clustering

1. INTRODUCTION

Explosive growth in the Internet and digital media has motivated recent research into developing techniques of music information retrieval [1, 2] for helping users locate their desired music from numerous options. Among such techniques, automatic classification of music genres [3-8] attracts particular attention as it serves as an effective way to structure and organize the large numbers of music files available on the Web. A music genre represents a type of music that has the common characteristics shared by its members and can be distinguished from other types of music. These characteristics are usually related to the instrumentation, rhythm, harmony, and melody of the music. A good review of automatic genre classification can be found in [3]. In essence, the research of music genre classification can be divided into two sub-topics: feature extraction, and classifier design. Some prominent work includes: Tzanetakis and Cook [4] explore a comprehensive set of audio features for genre classification using $K$-Nearest Neighbors and Gaussian mixture models; Li et al. [5] propose a feature extraction method based on wavelet coefficients histogram and study the multi-class learning problem. Lidy and Rauber [6] investigate psycho-acoustic transformations for music genre classification; Pampalk et al. [7] tackle the genre classification problem by trying to improve the measurement of audio similarities.

Received February 3, 2009; revised June 1 & August 26, 2009; accepted September 24, 2009.
Communicated by Chih-Jen Lin.
* This paper was supported in part by the National Science Council, Taiwan, under Grant No. NSC 96-2221-E-027-097-MY3.
To date, most genre-classification systems [4-7] follow a supervised framework, which consists of two phases: training and testing. In the training phase, it is required that genre-specific information is extracted or modeled using music data labeled manually. Then, in the testing phase, an unknown music recording is identified based on the genre patterns obtained in the training phase. However, such systems may not be suitable for personal music management, because manually labeling music based on individually-defined genres can be labor intensive and subject to inconsistency from time to time. In contrast to the supervised genre classification, this study investigates an unsupervised paradigm for music genre classification. It is aimed to partition a collection of unknown music recordings into several clusters such that each cluster contains recordings in only one genre, and different clusters represent different genres. This enables users to organize their personal music database without needing specific knowledge about genre.

As a closest related work to this study, [8] proposed an unsupervised classification framework to identify the genres of music clips. They used hidden Markov modeling to characterize the genre information of each music clip, thereby measuring the similarities between clips. Although music clips can then be grouped into clusters after knowing which clips are similar or dissimilar to each other, [8] did not study the issue as to how many clusters should be generated. In general, the greater the number of clusters generated, the higher the chance that within-cluster music clips belong to the same genre. However, if too many clusters are generated, music clips in the same genre would be split across multiple clusters, and hence the clustering will not be completed. Clearly, the optimal number of clusters is equal to the number of genres involved, which is, however, unknown and needs to be estimated.

In this study, we develop methods to identify music data in the same genre and estimate the number of genres involved. The methods start by representing each music recording as a Gaussian mixture model (GMM), and computing the likelihood probability that a music recording belong to the genre characterized by a certain GMM. Then, three inter-recording similarity measurements based on likelihoods are derived, namely, cross likelihood ratio, inverse Euclidean norm, and cosine measure. By applying the hierarchical agglomerative clustering, music recordings are partitioned as a tree of clusters. Next, a metric, called Rand index [9], is estimated to examine if some branches of the cluster tree should be deleted. The Rand index indicates the number of music recording pairs in the same genre that are placed in different clusters, or music recording pairs in different genres that are placed in the same cluster. It is found that the minimal value of the Rand index only appears when the number of clusters equals the true number of genres. Motivated by this fact, we propose determining the optimal number of clusters by searching for the branch of the cluster tree that produces the minimal value of the Rand index.

The remainder of this paper is organized as follows. In section 2, we formulate the problem of genre-based clustering of music data and describe the performance assessment method used in this study. Section 3 introduces the system design approach. Sections 4, 5, and 6 detail the major components of the proposed system, namely, inter-recording similarity computation, cluster generation, and the estimation of the optimal number of clusters. Section 7 discusses our experiment results. Then, in section 8, we present our conclusions and indicate the direction of our future work.
2. PROBLEM FORMULATION

As illustrated in Fig. 1, given a set of \( N \) unknown music recordings, each in one of the \( P \) different genres, where \( N \geq P \), and \( P \) is also unknown, our aim is to produce a partitioning of the \( N \) music recordings into \( M \) clusters such that \( M = P \), and each cluster consists exclusively of recordings associated with only one genre.

Fig. 1. The concept of clustering unknown music recordings based on genres.

Depending on the application, there are a number of ways to evaluate the clustering performance. In this study, we use two metrics: cluster purity [11] and the Rand index [9-11]. Cluster purity indicates the degree of correct clustering. It is represented by the probability that if we pick any music recording from a cluster twice at random, with replacement, both of the selected recordings in the same genre. Specifically, the purity of cluster \( c_m \) is computed by

\[
\rho_m = \frac{\sum_{p=1}^{P} \left( \frac{n_{mp}}{n_{m*}} \right)^2}{n_{m*}},
\]

(1)

Where \( n_{m*} \) is the total number of recordings in cluster \( c_m \), and \( n_{mp} \) is the number of recordings in cluster \( c_m \) associated with the \( p \)th genre. From Eq. (1), it follows that \( n_{m*} \leq \rho_m \leq 1 \), in which the upper bound and lower bound reflect that all the within-cluster recordings are in the same genre and completely different genres, respectively. To evaluate the overall performance of \( M \)-clustering, we compute an average purity

\[
\overline{\rho} = \frac{1}{N} \sum_{m=1}^{M} n_{m*} \rho_m.
\]

(2)

In contrast, the Rand index indicates the degree of incorrect clustering. It is defined as the number of recording pairs in the same genre that are placed in different clusters, or recording pairs in different genres that are placed in the same cluster, \( i.e., \)

\[
R(M) = \frac{M}{M} \sum_{m=1}^{M} n_{m*}^2 + \sum_{p=1}^{P} n_{p}^2 - 2 \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp},
\]

(3)

where \( n_{p} \) is the number of recordings associated with the \( p \)th genre. Obviously, the
smaller the value of \( R(M) \), the better the clustering performance will be. The Rand index can be alternatively represented as a mis-clustering rate:

\[
R(M) = \frac{\sum_{m=1}^{M} n_{m}^2 + \sum_{p=1}^{P} n_{p}^2 - 2 \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2}{\sum_{m=1}^{M} n_{m}^2 + \sum_{p=1}^{P} n_{p}^2}.
\] (4)

3. SYSTEM OVERVIEW

As shown in Fig. 2, the proposed clustering system consists of three major components: computation of inter-recording similarities, generation of clusters, and estimation of the number of genres. The similarity computation is designed to produce larger values for similarities between recordings in the same genre and smaller values for similarities between recordings in different genres. Then, clusters are generated in a bottom-up agglomerative manner, which sequentially merges the recordings deemed similar to each other. The outcome of the agglomeration procedure is a cluster tree with the number of clusters ranging from 1 to \( N \). The tree is then cut by determining the optimal number of clusters, which corresponds to an estimation of the number of genres involved.

4. INTER-RECORDING SIMILARITY COMPUTATION

The inter-recording similarity computation begins by characterizing each music recording as a set of genre-related parameters. Motivated by [3], our strategy for generating genre-related parameters is to perform timbre-based feature extraction followed by Gaussian mixture modeling.

4.1 Timbre-Based Feature Extraction

Although music genre information resides in timbre, rhythm, harmony, melody, etc., it is found [3] that timbre-based features are more useful than the others in distinguishing between genres. Among the timbre-based features investigated in [3], Mel-scale Frequency Cepstral Coefficients (MFCCs) feature [12] is observed superior to the others. However, this study found that another feature not explored in [3], called Renyi Entropy (RE), outperforms MFCCs in music genre classification. Based on this finding, we further propose a new feature, called Renyi Entropy Cepstral Coefficients (RECCs), by combining the MFCCs with RE.
In computing these features, a waveform signal is first divided into frames by using a fixed-length sliding window. Every frame then undergoes Hamming windowing and fast Fourier transform (FFT) with size $J$. Next, each frame is passed through a set of triangular filter banks, equally spaced on a Mel scale. Let $|X_{t,j}|$ denote the signal’s magnitude with respect to FFT index $j$ in frame $t$, where $1 \leq j \leq J$. Then,

(1) MFCCs are derived as follows:

$$
\text{MFCC}_{t,i} = \frac{1}{B} \sum_{b=1}^{B} \left\{ \log \left( \sum_{j=l_b}^{u_b} |X_{t,j}|^2 T_b(j) \right) \cdot \cos \left( \frac{\pi i}{B} (b - 0.5) \right) \right\}, \ 1 \leq i \leq B,
$$

where $B$ is the total number of the filter banks, $l_b$ is the lowest frequency index in the $b$th bank, $u_b$ is the highest frequency index in the $b$th bank, and $T_b(j)$ is the response of the $b$th bank. Briefly, MFCCs represent the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. It is found that the nonlinear mel scale of frequency approximates the human auditory system’s response more closely than the linearly-spaced frequency bands used in the regular cepstrum.

(2) RE, which is adapted from [13], can be computed as follows:

$$
\text{RE}_{t,b} = \frac{1}{1-\gamma} \log \left( \sum_{j=l_b}^{u_b} |X_{t,j}| T_b(j) \right), \ 1 \leq b \leq B,
$$

where $\gamma$ is a non-negative integer. We set $\gamma = 3$ empirically in the study. The Renyi Entropy, which is derived from information theory, is used to quantify the diversity, uncertainty or randomness of a system. When applying Renyi Entropy to characterize spectrum, it is assumed that a spectrum associated with higher level of diversity in energy distribution carries more information than a flat spectrum with equal energy in all frequencies. Compared to Eq. (5), we can see that Renyi Entropy uses the power of $\gamma$ of signal magnitudes, while MFCCs use the square of signal magnitudes. When $\gamma$ is set larger than two, it is expected that Renyi Entropy characterizes spectrum better than MFCCs in terms of information richness.

(3) RECCs, which can be viewed as a cepstral version of RE, are derived by

$$
\text{RECC}_{t,i} = \frac{1}{B} \sum_{b=1}^{B} \left\{ \text{RE}_{t,b} \cdot \cos \left( \frac{\pi i}{B} (b - 0.5) \right) \right\}, \ 1 \leq i \leq B.
$$

The RE in Eq. (6) uses a constant $\gamma = 3$ to intensify the normal power spectrum. However, it is generally recognized that audio features based on spectrum are sensitive to noise disturbance, and this is why cepstrum-based features are commonly-used instead. Similar to MFCCs which applies DCT to convert mel-spectrum to mel-cepstrum, RECCs are the results of the discrete cosine transform of RE spectrum. In our experiment, we examined the three features separately.
4.2 Gaussian Mixture Modeling

Each stream of feature vectors extracted from a music recording is then represented by a Gaussian Mixture Model (GMM). The purpose of using GMM is to characterize the occurrence of feature vectors for a music recording. It is assumed that each piece of music has its own genre pattern that reflects in the distribution of timbre-based features over a span of time. A GMM approximates any genre patterns by a mixture of Gaussian densities. The parameters of a GMM consist of means, covariances, and mixture weights, which are commonly estimated using Expectation-Maximization (EM) algorithm [14].

If there are \( N \) music recordings, \( X_1, X_2, \ldots, X_N \), to be clustered, where \( X_i = \{ x_{i,1}, x_{i,2}, \ldots, x_{i,T} \} \), 1 \( \leq i \leq N \), is a stream of MFCCs, RE, or RECCs with length \( T \), we generate \( N \) recording individual GMMs \( \lambda_1, \lambda_2, \ldots, \lambda_N \). It is hoped that, if two music recordings, say \( X_i \) and \( X_j \), belong to the same genre, the GMMs \( \lambda_i \) and \( \lambda_j \) would be more similar in some sense, compared to any two GMMs associated with different genres. To measure the similarity between GMMs, we consider the following metrics.

1. Cross likelihood ratio [15]:

\[
S(X_i, X_j) = \log \frac{\Pr(X_i | \lambda_j)}{\Pr(X_j | \lambda_j)} + \log \frac{\Pr(X_j | \lambda_j)}{\Pr(X_i | \lambda_i)},
\]

where \( \Pr(X_i | \lambda_j) \) is the likelihood probability that music recording \( X_i \) belong to the genre characterized by GMM \( \lambda_j \). Specifically, the likelihood probability is computed using

\[
\Pr(X_i | \lambda_j) = \prod_{t=1}^{T} \sum_{k=1}^{K} w_{j,k} \cdot \frac{1}{\pi^K |C_{j,k}|} \exp\left\{ -\frac{1}{2} (x_{i,t} - \mu_{j,k})' C_{j,k}^{-1} (x_{i,t} - \mu_{j,k}) \right\},
\]

where \( K \) is the number of mixture Gaussian components; \( w_{j,k}, \mu_{j,k}, \text{ and } C_{j,k} \) are the \( k \)th mixture weight, mean, and covariance of model \( \lambda_j \), respectively; and prime (’') denotes the vector transpose.

2. Inverse Euclidean norm:

\[
S(X_i, X_j) = 1 / \left\| V_i - V_j \right\| = 1 / \left[ \sum_{k=1}^{N} \left( \log \Pr(X_i | \lambda_k) - \log \Pr(X_j | \lambda_k) \right)^2 \right]^{1/2},
\]

where \( V_j = \begin{bmatrix} \log \Pr(X_j | \lambda_1) \\ \log \Pr(X_j | \lambda_2) \\ \vdots \\ \log \Pr(X_j | \lambda_N) \end{bmatrix} \)

and \( \left\| \cdot \right\| \) represents the Euclidean norm (or 2-norm) of a vector.

3. Cosine measure
\[ S(X_i, X_j) = \frac{\sum_{k=1}^{N} \left[ \log \Pr(X_j | \hat{\lambda}_k) \cdot \log \Pr(X_j | \hat{\lambda}_k) \right]}{\sqrt{\sum_{k=1}^{N} \left[ \log \Pr(X_i | \hat{\lambda}_k) \right]^2} \sqrt{\sum_{k=1}^{N} \left[ \log \Pr(X_j | \hat{\lambda}_k) \right]^2}}, \]

where \(<,>\) represents the dot product.

5. CLUSTER GENERATION

After computing the inter-recording similarities, the next step is to assign the recordings deemed similar to each other to the same cluster. This is done by an agglomerative hierarchical clustering method [16], which consists of the following procedure:

1. begin initialize \( M \leftarrow N \), and form clusters \( c_i \leftarrow \{X_i\}, i = 1, 2, \ldots, N \)
2. do
3. find the most similar pair of clusters, say \( c_i \) and \( c_j \)
4. merge \( c_i \) and \( c_j \)
5. \( M \leftarrow M - 1 \)
6. until \( M = 1 \)
7. end

The similarities between a pair of clusters, say \( c_i \) and \( c_j \), can be derived from the inter-recording similarities, according to one of the following heuristic measures:

(1) complete linkage \( S_c(c_i, c_j) = \min_{X_n \in c_i, X_k \in c_j} S(X_n, X_k) \),
(2) single linkage \( S_c(c_i, c_j) = \max_{X_n \in c_i, X_k \in c_j} S(X_n, X_k) \),
(3) average linkage \( S_c(c_i, c_j) = \frac{1}{\#(X_n \in c_i, X_k \in c_j)} \sum_{X_n \in c_i, X_k \in c_j} S(X_n, X_k) \),

where \( \#(X_n \in c_i, X_k \in c_j) \) denotes the number of recording pairs involved in the summation.

6. ESTIMATION OF THE NUMBER OF GENRES

In general, the greater the number of clusters specified, the higher the level of homogeneity within a cluster. However, if too many clusters are generated, music recordings associated with a single genre would spread over multiple clusters; hence, the clustering would not be complete. Clearly, the optimal number of clusters is equal to the number of genres, which is unknown and must be estimated.

Consider a collection of \( N \) music recordings to be partitioned into \( M \) clusters. The optimal value of \( M \) must be an integer between 1 and \( N \). Thus, if we produce a set of possible partitionings, in which the number of clusters ranges from 1 to \( N \), the task of determining the optimal value of \( M \) would amount to selecting one of the \( N \) partitionings that achieves the level of within-cluster homogeneity as high as possible with the number of
clusters as small as possible. Such a task is often referred to as cluster validity [17]. In general, the algorithms to cluster validity can be divided into two categories. One is to define a validity function and then minimize or maximize the validity function to find the optimal number of clusters. For example, [18] propose a fuzzy hyper-volume measure to validate clusters based on a fuzzy maximum likelihood estimation. In using hierarchical clustering, the most popular approach to cluster validity is based on maximum lifetime criterion [16], where the lifetime is defined as the difference between the distances at two successive nodes. It is aimed to track the lifetime of all clusters and search for clusters that have a large lifetime. However, one major drawback of maximum lifetime criterion is that a single large distance can dominate the decision when the inter-class or intra-class variation is large. On the other hand, another category of cluster validity does not use any validity functions. It is performed by examining and comparing the duration or survival period of clusters when some clustering parameters are varied. For example, an optimal number of clusters can be obtained by finding the longest lasting cluster set in a hierarchy [19]. However, our experiments find that such an approach tend to overestimate the number of genres seriously.

To better handle this problem, we propose a new cluster validity approach based on Rand index. It is aimed to estimate the Rand index for each of the possible partitionings, and then choosing the one achieving the smallest value of Rand index. The idea is motivated by the fact that the Rand index usually decreases with an increase in the value of \( M \) initially, and reaches the minimum at \( M = P \). When \( M > P \), the Rand index starts to increase as the value of \( M \) increases. To illustrate why the minimal value of \( R(M) \) only occurs when \( M = P \), let us consider the following cases.

(1) The clustering is perfect, which satisfies

\[
\begin{pmatrix}
  n_1 & n_{21} & \ldots & n_{21} \\
  n_2 & n_{22} & \ldots & n_{22} \\
  \vdots & \vdots & \ddots & \vdots \\
  n_P & n_{P2} & \ldots & n_{P2}
\end{pmatrix}
= \begin{pmatrix}
  n_1 & 0 & \ldots & 0 \\
  0 & n_2 & \ldots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \ldots & n_P
\end{pmatrix}_{P \times P},
\]

where \( n_i = n_{ij} = n_{ji}, 1 \leq i \leq P \). Then, the resulting Rand index is

\[
R^2(P) = \sum_{m=1}^{P} n_m^2 + \sum_{p=1}^{P} n_p^2 - 2 \sum_{m=1}^{P} \sum_{p=1}^{P} n_{mp}^2 = \sum_{m=1}^{P} n_m^2 + \sum_{p=1}^{P} n_p^2 - 2 \sum_{k=1}^{P} n_k^2 = 0.
\]

(2) Let \( M = P + 1 \), and modify Eq. (13) by splitting cluster \( c_k \) into two clusters, \( c_k \) and \( c_{P+1} \), i.e.,

\[
\begin{pmatrix}
  n_1 & n_{21} & \ldots & n_{21} \\
  n_2 & n_{22} & \ldots & n_{22} \\
  \vdots & \vdots & \ddots & \vdots \\
  n_P & n_{P2} & \ldots & n_{P2}
\end{pmatrix}
= \begin{pmatrix}
  n_1 & 0 & \ldots & 0 & 0 & 0 \\
  0 & n_2 & \ldots & 0 & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
  0 & 0 & \ldots & n_k & 0 & n_{(P+1)k} \\
  \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
  0 & 0 & \ldots & 0 & n_P & 0
\end{pmatrix}_{(P+1) \times P}.
\]
where \( n_{1k} + n_{(p+1)k} = n_k \). Then, the resulting Rand index is

\[
R(P + 1) = \sum_{m=1}^{P+1} n_{m}^2 + \sum_{p=1}^{P} n_{p}^2 - 2 \sum_{m=1}^{P+1} \sum_{p=1}^{P} n_{mp}^2
\]

\[
= \left( \sum_{m=1}^{P} n_{m}^2 - n_k^2 + n_{k}^2 + (n_{p+1})k \right) + \sum_{p=1}^{P} n_{p}^2 - 2 \left( \sum_{m=1}^{P} n_{m}^2 - n_k^2 + n_{k}^2 + (n_{p+1})k \right)
\]

\[
= n_k^2 - n_{p+1}^2 = n_k^2 - n_{p+1}^2 - (n_k - n_{p+1})^2 = 2n_k(n_k - n_{p+1}) > 0.
\]

(3) Let \( M = P - 1 \), and modify Eq. (13) by merging cluster \( c_P \) into cluster \( c_k \), i.e.,

\[
\begin{pmatrix}
  n_1 & n_{21} & \cdots & n_{M1} \\
  n_2 & n_{22} & \cdots & n_{M2} \\
  \vdots & \vdots & \ddots & \vdots \\
  n_P & n_{2P} & \cdots & n_{MP}
\end{pmatrix} =
\begin{pmatrix}
  n_1 & 0 & \cdots & 0 & 0 \\
  n_2 & 0 & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \vdots & \vdots \\
  0 & 0 & \cdots & n_k & 0 \\
  0 & 0 & \cdots & n_P & 0
\end{pmatrix}_{P+M}.
\]

Then, the resulting Rand index is

\[
R(P - 1) = \sum_{m=1}^{P} n_{m}^2 + \sum_{p=1}^{P} n_{p}^2 - 2 \sum_{m=1}^{P} \sum_{p=1}^{P} n_{mp}^2
\]

\[
= \left( \sum_{m=1}^{P} n_{m}^2 - n_k^2 + (n_k + n_P)^2 \right) + \sum_{p=1}^{P} n_{p}^2 - 2 \sum_{m=1}^{P} n_{m}^2
\]

\[
= 2n_k n_P > 0.
\]

From these cases, we observe that, in general, \( R(M) > R(P) \) if \( M \neq P \). This property motivates us to determine the number of genres involved by finding the value of \( M \) that achieves the minimal Rand index.

Recalling the Rand index in Eq. (3), the first term in the right side of the equation, \( \sum_{m=1}^{M} n_{m}^2 \), can be computed based on the clustering result. The second term in the right side of the Eq. (3), \( \sum_{p=1}^{P} n_{p}^2 \), is a constant irrelevant to clustering. However, the third term of Eq. (3), \( \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2 \), requires that the true genre attribute of each recording be known in advance, which cannot be computed directly. To solve this problem, we represent \( \sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2 \) by

\[
\sum_{m=1}^{M} \sum_{p=1}^{P} n_{mp}^2 = \sum_{m=1}^{P} \sum_{p=1}^{P} \left[ \sum_{i=1}^{N} \delta(h_i^{(M)}, m) \delta(\alpha_i, p) \right] - 1
\]
\[
\begin{align*}
&= \sum_{m=1}^{M} \sum_{p=1}^{P} \left[ \sum_{i=1}^{N} \delta(h_i^{(M)}, m) \delta(o_i, p) \right] \left[ \sum_{j=1}^{N} \delta(h_j^{(M)}, m) \delta(o_j, p) \right] \\
&= \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i^{(M)}, m) \delta(o_i, p) \delta(h_j^{(M)}, m) \delta(o_j, p) \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i^{(M)}, h_j^{(M)}) \delta(o_i, o_j),
\end{align*}
\]

where \( \delta(\cdot) \) is a Kronecker Delta function, \( h_i^{(M)} \) is the index of cluster where the \( i \)th recording is located, and \( o_i \) is the true or manually-specified genre class for the \( i \)th recording. Note that \( h_i^{(M)} \), \( 1 \leq i \leq N \), is an integer between 1 and \( M \), if \( M \) clusters are generated. The term \( \delta(o_i, o_j) \) in Eq. (19) is then approximated by the similarity between \( X_i \) and \( X_j \). Specifically,

\[
\delta(o_i, o_j) \leftarrow \delta(X_i, X_j) = \begin{cases} 
1, & \text{if } i = j \\
\frac{S(X_i, X_j)}{S(X_i, X_{\hat{z}_i})}, & \text{if } i \neq j
\end{cases}
\]

where \( S(X_i, X_j) \) is a certain similarity measure between \( X_i \) and \( X_j \), and \( X_{\hat{z}_i} \) is the recordings most similar to \( X_i \). Note that \( 0 \leq \hat{\delta}(X_i, X_j) \leq 1 \). Hence, the optimal set of cluster indices can be determined by

\[
M^* = \arg \min_{1 \leq M \leq N} \hat{R}(M),
\]

where \( \hat{R}(M) \) is an approximated Rand index:

\[
\hat{R}(M) = \sum_{m=1}^{M} n_m^2 - 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i^{(M)}, h_j^{(M)}) \hat{\delta}(X_i, X_j).
\]

7. EXPERIMENTS

7.1 Music Data

The music data used in this study was extracted from the Training Set and Development Set of Magnatune [20] used in MIREX 2005. It comprised six music genres: classical, electronic, jazz, rock & pop, punk & metal, and world. We randomly selected 40 songs per genre to conduct the clustering experiments. Thus, there were a total of 240 songs. For convenience of experiment, we converted each song from its original MP3 (MPEG-1 Audio Layer 3) representation with 44.1KHz sampling rate to waveform representation with 22.05KHz sampling rate.
Table 1. Clustering results obtained with various audio features and inter-recording similarity measures, in which the inter-cluster similarity measure was based on complete linkage.

<table>
<thead>
<tr>
<th></th>
<th>MFCCs</th>
<th>RE</th>
<th>RECCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Likelihood Ratio</td>
<td>0.58</td>
<td>0.61</td>
<td>0.62</td>
</tr>
<tr>
<td>Reverse Euclidean Norm</td>
<td>0.61</td>
<td>0.63</td>
<td>0.69</td>
</tr>
<tr>
<td>Cosine Measure</td>
<td>0.61</td>
<td>0.66</td>
<td>0.71</td>
</tr>
</tbody>
</table>

(b) Rand index.

<table>
<thead>
<tr>
<th></th>
<th>MFCCs</th>
<th>RE</th>
<th>RECCs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross Likelihood Ratio</td>
<td>0.43</td>
<td>0.37</td>
<td>0.35</td>
</tr>
<tr>
<td>Reverse Euclidean Norm</td>
<td>0.37</td>
<td>0.34</td>
<td>0.30</td>
</tr>
<tr>
<td>Cosine Measure</td>
<td>0.37</td>
<td>0.32</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 2. Performance comparison of different inter-cluster similarity measures, in which the system used RECC feature and cosine measure.

<table>
<thead>
<tr>
<th></th>
<th>Complete linkage</th>
<th>Single linkage</th>
<th>Average linkage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purity</td>
<td>0.71</td>
<td>0.51</td>
<td>0.63</td>
</tr>
<tr>
<td>Rand index</td>
<td>0.27</td>
<td>0.50</td>
<td>0.39</td>
</tr>
</tbody>
</table>

7.2 Experimental Results

The first experiment was conducted under the condition that the number of manually-specified genres is known, which means the number of clusters to be generated is given a priori. This allows us to test the validity of the first two components of our music clustering system, namely, the computation of inter-recording similarity and generation of clusters. In this case, our aim is to partition the music recordings into 6 clusters, such that each cluster contains music data associated with only one genre.

Table 1 shows the results obtained with various audio features and inter-recording similarity measures. Here, the inter-cluster similarity measure was based on complete linkage. In addition, the number of Gaussian components used in each GMM was empirically determined to be 32. Note that the system using MFCC feature and cross likelihood ratio could represent a baseline system proposed by [8]. The resulting purity and Rand index were computed using Eqs. (2) and (4), along with the ground truth. We can see that RE feature performs better than MFCC feature, which shows the higher value of purity and lower value of Rand index. It is also clear that RECC feature further outperforms both MFCC feature and RE feature. Among the three inter-recording similarity measures, we can observe that cosine measure performs best, cross likelihood ratio performs worst, and reverse Euclidean norm are between them. On the other hand, Table 2 shows a performance comparison of different inter-cluster similarity measures, in which the system used RECC feature and cosine measure. We can see that complete linkage is superior to single linkage and average linkage in this task. Thus, in the subsequent experiments, we used the system with RECC feature, cosine measure, and complete linkage.

Fig. 3 shows the resulting purity and Rand index as a function of the varied number of clusters. Naturally, the value of purity increases as the number of clusters increases. When 56 clusters were generated, we obtained a purity of 1.0. However, it is obvious that
Fig. 3. The resulting purity and Rand index as a function of varied number of clusters.

(a) For the first subset consisted of 40 classical songs, 40 electronic songs, and 40 jazz songs.
(b) For the second subset consisted of 40 rock & pop songs, 40 punk & metal songs, and 40 world songs.
(c) For the entire 240-song database consisted of six genres.

Fig. 4. Approximated Rand index as a function of varied number of clusters.
such a clustering result is not perfect, since multiple music recordings in the same genre are placed in different clusters. On the other hand, we can see from Fig. 3 that the value of Rand index increases with the increase of the number of clusters in the beginning, but declines gradually after an excess of clusters is created. The minimum of Rand index appears near the number of genres.

We then examined the system under the condition that the number of manually-specified genres is unknown and must be estimated. The database was divided into three subsets involving different numbers of genres. The first subset consisted of the 40 classical songs, 40 electronic songs, and 40 jazz songs. The second subset consisted of the 40 rock & pop songs, 40 punk & metal songs, and 40 world songs. The third subset was the whole database, i.e., 240 songs in the six genres. We conducted clustering experiment for each subset separately, in order to examine if the optimal numbers of clusters determined by using Eq. (21) could be close to 3 or 6. Fig. 4 shows the clustering results. We can observe that the minimum value of the approximated Rand indices were located close to the true numbers of genres, although the resulting estimated numbers of genres tend to be overestimated slightly. Such an overestimate is due to the fact that the term $\hat{\delta}(X_i, X_j)$ in Eq. (22) is almost always smaller than $\delta(o_i, o_j)$ when $o_i = o_j$, resulting that the approximated Rand indices does not reach its minimal value until the number of clusters exceeds the true number of genres. To elaborate further, Fig. 5 shows some quantities in Eqs. (19) and (22) computed for the third subset. Let $A(M) = \sum_{m=1}^{M} \sum_{nn'} \delta(h_i^{(M)}, h_j^{(M)}) \delta(o_i, o_j)$, and $B(M) = 2 \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(h_i^{(M)}, h_j^{(M)}) \hat{\delta}(X_i, X_j)$. We can see from Fig. 5 that the values of $B(M)$ are almost smaller than $B(M)$. As a result, the valley of $\hat{R}(M) = A(M) - B(M)$ curve appears in a larger value of $M$, compared to the counterpart of $A(M) - B(M)$ curve yielded with true genre classes. Despite this, the overestimate based on $\hat{R}(M)$ is under a controllable level.

Table 3 summarizes the results obtained with known numbers of genres and estimated numbers of genres. For the performance comparisons, we also implemented two cluster validity approaches, one is based on the maximum lifetime criterion [16], and the other is
Table 3. A summarization of the Rand indices obtained with known numbers of genres and estimated numbers of genres, based on the proposed method, maximum lifetime criterion [16], and longest lasting cluster set [19], respectively. The number inside the parenthesis indicates the estimated number of genres.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Known No. of genres</th>
<th>Estimated No. of genres</th>
<th>The proposed method</th>
<th>Maximum lifetime criterion</th>
<th>Longest lasting cluster set</th>
</tr>
</thead>
<tbody>
<tr>
<td>First subset</td>
<td>0.19</td>
<td>0.21 (5)</td>
<td>0.20 (4)</td>
<td>0.24 (9)</td>
<td></td>
</tr>
<tr>
<td>Second subset</td>
<td>0.23</td>
<td>0.22 (6)</td>
<td>0.39 (41)</td>
<td>0.33 (12)</td>
<td></td>
</tr>
<tr>
<td>Third subset</td>
<td>0.27</td>
<td>0.23 (10)</td>
<td>0.37 (36)</td>
<td>0.25 (16)</td>
<td></td>
</tr>
</tbody>
</table>

based on the longest lasting cluster set [19]. We can see from Table 3 that the estimated numbers of clusters based on the maximum lifetime criterion are not steadily near the true numbers of clusters. In particular, the approach fails to deal with the second and third subsets. On the other hand, we can also see from Table 3 that the approach based on the longest lasting cluster set largely over-estimates the true numbers of genres. It is clear from Table 3 that the proposed approach outperforms the other two cluster validity approaches. The Rand indices achieved with the estimated numbers of genres using the proposed approach were comparable to those achieved with known numbers of genres. This result confirms the feasibility of our clustering system.

The above experiments show that roughly 70% music recordings in the same genre were correctly clustered, given the result of Rand index = 0.28. Although the results are far from perfect, they show the viability of using unsupervised clustering to index music data by genre.

8. CONCLUSIONS

We have studied an unsupervised clustering paradigm for identifying music recordings in the same genre. This is done by measuring the similarities between music recordings and using the hierarchical agglomerative clustering to group together the recordings deemed similar to one another. We have also proposed a method based on the Rand index to determine the optimal number of clusters automatically, corresponding to the number of genres involved. Experiments show the feasibility of using unsupervised clustering to index music data.

Despite the viability, the methods proposed in this study can only be regarded as a preliminary investigation in realistic music data indexing applications. To be of more practical use, more work is needed to study the effectiveness and efficiency of clustering a large scale of music collection. The fundamental problem would be how to characterize music data as audio features more closely related to music genre.

REFERENCES

Wei-Ho Tsai (蔡偉和) received his B.S. degree in Electrical Engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1995. He received his M.S. and Ph.D. degrees in Communication Engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1997 and 2001, respectively. From 2001 to 2003, he was with Philips Research East Asia, Taipei, Taiwan, where he worked on speech processing problems in embedded systems. From 2003 to 2005, he served as a Postdoctoral Fellow at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently an Associate Professor in the Department of Electronic Engineering and Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan. His research interests include spoken language processing and music information retrieval.

Duo-Fu Bao (鮑多福) received his M.S. degrees in Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan, in 2008. He is currently an R&D engineer with Fotek Controls Co., Ltd. His research interests include audio engineering and electronic controllers.