Modeling Concept Dynamics for Large Scale Music Search

Jialie Shen† HweeHwa Pang† Meng Wang‡ Shuicheng Yan⋆
† School of Information Systems, Singapore Management University, Singapore
{jlshen, hhpang}@smu.edu.sg
‡ Hefei University of Technology, Hefei, China
eric.mengwang@gmail.com
⋆ Department of ECE, National University of Singapore, Singapore
eleyans@nus.edu.sg

ABSTRACT

Continuing advances in data storage and communication technologies have led to an explosive growth in digital music collections. To cope with their increasing scale, we need effective Music Information Retrieval (MIR) capabilities like tagging, concept search and clustering. Integral to MIR is a framework for modelling music documents and generating discriminative signatures for them. In this paper, we introduce a multimodal, layered learning framework called $DMCM$. Distinguished from the existing approaches that encode music as an ensemble of order-less feature vectors, our framework extracts from each music document a variety of acoustic features, and translates them into low-level encodings over the temporal dimension. From them, $DMCM$ elucidates the concept dynamics in the music document, representing them with a novel music signature scheme called Stochastic Music Concept Histogram ($SMCH$) that captures the probability distribution over all the concepts. Experiment results with two large music collections confirm the advantages of the proposed framework over existing methods on various MIR tasks.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query formulation, Search process; H.5.5 [**Sound and Music Computing**]: Systems

General Terms

Algorithms, Design, Experimentation, Human Factors

Keywords

Music Information Retrieval, Similarity Measure, Music Concepts

1. INTRODUCTION

The past decade has witnessed a tremendous growth in the availability of digital music on various application platforms. At the same time, the pervasiveness of social media and affordability of home media servers are bringing about a fundamental change in the way people enjoy and share music. Indeed, online music distribution has overtaken physical media like compact discs [2, 6, 7, 13], becoming the dominant distribution channel with consumers. These trends call for Music Information Retrieval (MIR) capabilities for tagging (for example to enable browsing and faceted retrieval), searching by concepts (in a similar manner as text search), and clustering/classification (to automatically organize the music library). Underpinning these capabilities is an effective framework for modeling rich musical content, so as to generate signatures that capture the distinct characteristics of individual music documents.

While music modeling has been a long-standing research topic, substantial scope remains for further advances. To illustrate the challenges involved, consider the song "Bohemian Rhapsody" by Queen - a British rock band. There are six sections in the song: intro, balla, guitar solo, opera, hard rock and outro. At various points, the song is performed by different singers and features different musical instruments like piano and guitar. Moreover, its tempo speeds up mid-way through. In characterizing the song, at the accoustic level the importance of the timbral, spectral and rhythmic features would vary over time. On a semantic level, the concepts associated with segments of the song, like "tender", "comforting" and "romantic", also vary as shown in Figure 1 [1].

Due to those challenges, music modeling goes beyond extraction of acoustic features, to learning their inter-play and, from there, deducing semantic concepts. However, existing approaches in [28, 34, 35, 33, 17, 25, 23] simply represent a music document as a bag of audio features and directly apply machine learning techniques on the features. Surprisingly, less attention has been paid on modeling the interaction or association among musical features, subpieces and dependency between different levels of concept. Moreover, the free-patch representation ignores information about temporal dynamics and order, which has been proven to be very important for accurate music search and analysis.

In this paper, we postulate three principles for an effective music modeling framework. First, the framework should capture both low-level acoustic features and high-level concepts. Second, the acoustic features and concepts may exhibit non-linear dependencies. Effectively modeling the complex associations between different concepts can be very helpful in improving system performance. Third, the characteristics of a music document are likely to vary over the temporal dimension. Building on the principles, we introduce a new Dynamic Musical Concept Mixture ($DMCM$) framework to facilitate comprehensive music modelling. To account for temporal variations, the framework splits each music doc-

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

[1]The wave form is generated using Audacity.

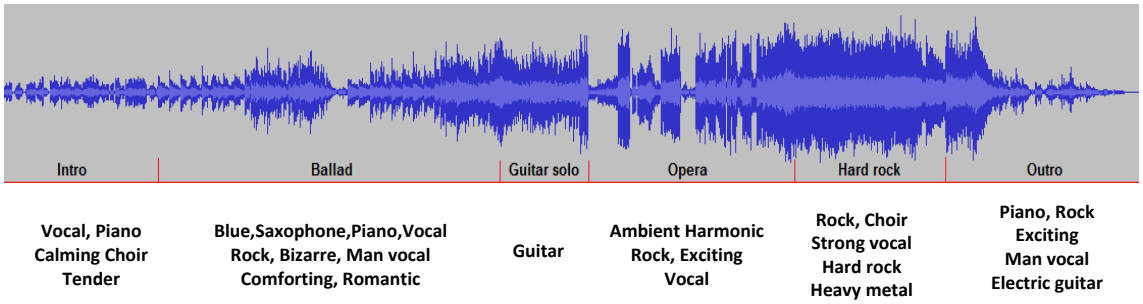

Figure 1: Various concepts related to song "Bohemian Rhapsody" by the British rock band Queen

ument by time into multiple segments, and derives a two-layer model for each segment:

- The *music preprocessing* layer extracts multiple acoustic features and maps them into an audio word from a pre-computed codebook.
- The *concept dynamics modeling* layer derives from the underlying audio words a Stochastic Music Concept Histogram ($SMCH$), essentially a probability distribution over the high-level concepts.

Thus, the music document is translated into a set of low-level audio words, and a corresponding set of high-level $SMCH$s. To the best of our knowledge, no prior work has adopted a similar approach as ours. To validate the effectiveness of the proposed framework, we have carried out a comprehensive set of experiment studies with two large music collections. A comparative analysis involving existing state-of-the-art methods confirms that our framework achieves substantial improvement in accuracy and robustness for various MIR tasks.

The rest of the paper is structured as follows: In Section 2, we review and analyze related work in the area of music signature generation. Their assumptions, limitations and application domains are discussed in detail. In Section 3, we introduce our feature extraction scheme and audio word generation process. Following that, Section 4 presents the proposed $DCMC$ framework. Section 5 reports our experiment configuration, including test collection, evaluation metrics and evaluation methodology. The experiment results are reported in Section 6. Finally, the article is concluded in Section 7.

2. LITERATURE REVIEW

Computing effective music signatures is an important but challenging problem. Various research communities have proposed approaches that built upon different musical features. They include textual labels for the title, performers, composers, style and symbolic representations of melody (e.g., MIDI and digital music scores). Due to space limitation, we focus on existing studies based on acoustic (content-based) features, which are the most relevant to our paper.

Music descriptor generation aims to derive effective content representation for information management or analysis applications (e.g., search, classification or tagging). While many systems exist for content-based speech recognition, there is much less effort on descriptive music feature extraction or modeling. Most of them directly treat low level spectral features as music signatures. Typical examples include the system developed by Nam and Berger [24], which applies three low-level acoustic features (spectral centroid, short time energy, and zero crossing rate) for automatic music genre classification. Another example is the work by Daudet [9] which applies pruned wavelet trees to model transients inside music signals. In [19], Lu et al. apply nine different audio features for audio classification. The feature set used in this study consists of MFCCs, zero crossing rates (ZCR), short time energy(STE), sub-band power distribution, brightness, bandwidth, spectrum flux (SF), band periodicity (BP) and noise frame ratio (NFR). Moreover, a support vector machine (SVM) is used for statistical classification. Tzanetakis et al. develop MARSYAS system - an advanced infrastructure for characterizing different acoustic properties of music signals [37]. The musical features extracted by MARSYAS include timbral texture, pitch content and rhythm. The features are combined linearly, and input to a SVM classifier. The study achieved a 61% classification accuracy on a small test collection.

More recently, Li et al. propose a DWCHs scheme to calculate Daubechies wavelet coefficients of music signal [16]. The key conjecture underlying the approach is that compared to conventional spectral analysis, a wavelet histogram technique is better able to capture both local and global temporal information inside the music signal. Their empirical results obtained on large scale test collections demonstrate that due to the wavelets' accurate summarization of the probability distribution over time and frequency domains, DWCHs outperforms MARSYAS with different machine learning classifiers for music genre classification.

The techniques described above rely on either one type of low level feature extracted using spectral analysis, or a linear concatenation of multiple acoustic features. It has been proven that such approaches generally do not produce a comprehensive and discriminative representation of music sequences. This is because the human auditory system senses and analyzes music sequences by integrating multiple characteristics in a nonlinear fashion. Consequently, any single type of acoustic feature is unlikely to contain sufficient information to represent the music effectively. Further, the acoustic features extracted from a music sequence are not weighted equally in human music perception and comprehension; in other words, the human hearing system gives different responses/weights to timber, rhythm and pitch. Thus, systems that assume a linear combination of acoustic features are inherently handicapped.

There is also an emerging stream of literature exemplified by Sheh et.al [27], Turnbull et.al [32] and Zhang et.al [39] that adopts statistical modeling to generate discriminative music descriptors from raw acoustic features. Shen et.al [30] develop the InMAF scheme, with a hybrid architecture that incorporates principal component analysis (PCA) and a multilayer perceptron neural network, to combine multiple musical properties in a nonlinear fashion. Experiment results obtained with two small data sets show its effectiveness and robustness against various kinds of audio alteration. Song et.al [31] propose a semi-supervised distance-based learning framework to integrate music features for genre classification.

Symbols	Definitions
$SMCH$	Stochastic music concept histogram
C	Total number of basic music concepts
S	Total number of music segments
R	Size of codebook
p_{cf}	Probability of music object belonging to concept c
$\Re^C$	Metric space generated by $SMCH$ with C dimensions
$\mathcal{M}$	A set of mappings modeled by $DMCM$
$\mathcal{CB}$	Audio codebook - a set of audio words
$\mathcal{T}$	A set of textual words to represent music concepts
$\mathcal{Z}$	A set of latent concepts
aw	Notation of audio word
$msim$	Notation of music similarity based on $SMCH$
A	Notation of feature for audio words
T	Notation of feature for textual words
P	A set of matching probabilities modelled by graph G
G	Notation of weighted bipartite graph for generative model
M	Notation of mapping between audio words and textual words
ϕ_A	Notation of kernel function for audio words
ϕ_T	Notation of kernel function for textual words
θ	Notation of system parameter for generative model

Table 1: Summary of symbols and definitions

Turnbull et.al [34, 35] propose a supervised multi-class labeling (SML) probabilistic scheme to generate semantic descriptors for large scale music retrieval; the framework builds upon GMMs. One of the common disadvantages suffered by the aforementioned methods is that feature vectors extracted from a music sequence are assumed to be independent and identically distributed (i.i.d.), and the music sequence is encoded as an ensemble of orderless feature vectors. Consequently, temporal dynamics in the music sequence are not accounted for properly. Motivated by the concerns, Coviello et.al [8] develop a dynamic texture model [10] to facilitate automatic music annotation and retrieval. It effectively accounts for not only temporal dynamics but also the timbral content in music. Experiment results obtained with the CAL500 dataset shows that the model improves the performance of music search and annotation significantly.

3. AUDIO WORD GENERATION

In this study, we represent audio features as a set of "audio words". To compute audio words to describe music sequences, raw input signal is firstly partitioned into several segments of equal length. From each segment, we extract five different kinds of features to form the low level music representation. They include timber, spectrum, rhythm, pitch and time as explained below.

- **Timbral features**: They characterize the timbral property of acoustic objects. In our system, short time Fourier transform is applied in the calculation. The timbral features computed include *Mel-Frequency Cepstral Coefficients* (MFCCs) [18], *Spectral Centroid*, *Rolloff*, *Flux*, *Low-Energy feature* [36], and *Spectral Contrast* [20]. The total dimensionality of the timbral features is 20.

- **Spectral features**: They characterize the spectral composition of music signal. In our implementation, each spectral feature vector contains *Auto-regressive (AR) features*; *Spectral Asymmetry, Kurtosis, Flatness, Crest Factors, Slope, Decrease, Variation*; *Frequency Derivative of Constant-Q Coefficients*; and *Octave Band Signal Intensities* [20]. The total dimensionality of these feature vectors is 20.

- **Rhythmic features**: They summarize the patterns of an acoustic object over a certain duration. The rhythmic features considered in this study include: *Beat Histogram* [36]; *Rhythm Strength*, *Regularity* and *Average Tempo* [20]. The total dimensionality is 12.

- **Wavelet features**: The wavelet transform provides a good scheme to divide raw signal into different frequency components over the temporal dimension. In our framework, the Daubechies wavelet filter with seven levels of decomposition is used for extracting a histogram of the wavelet coefficients at each subband [21]. For each subband, we compute the first three moments and the energy. Thus the dimensionality of the wavelet feature vectors is 28.

- **Time features**: Unlike traditional approaches, our system considers temporal coordinate information, which provides a more informative representation of local acoustic structure. Our time feature contains starting time, end time and the length of a sequence.

The k-mean algorithm is applied to cluster each of the feature spaces to form audio words [11]. The values of k are preconfigured to be 40 for timbral feature, 30 for spectral features, 35 for rhythmic features, 35 for wavelet features and 10 for time features. The whole process can be treated as a special transformation Φ:

$$\Phi_f : AF_f \rightarrow \boldsymbol{c}_f = \{c_{1f}, c_{2f}, ..., c_{kf}\}, \quad (1)$$

where c_{kf} denotes the centroid of cluster kf after k-means clustering on acoustic feature f. The audio word aw is defined as a sequence of audio feature cluster centroids and its representation is given as,

$$aw = [c_{ti}, c_r, c_{sp}, c_w, c_t] \quad (2)$$

where c_{ti}, c_r, c_{sp}, c_w, and c_t denote the cluster generated using timber feature, rhythm feature, spectral feature, wavelet feature and time feature respectively. Similar to keywords in text documents, we can construct an audio codebook $\mathcal{CB}$ containing R audio words,

$$\mathcal{CB} = \{aw_1, aw_2, ..., aw_R\} \quad (3)$$

where aw_r denotes audio word r in codebook $\mathcal{CB}$. In general, the music segments represented by the same set of audio words share certain levels of closeness and contain similar semantic concepts. Because huge variations are commonly encountered in raw music signal in practice, no existing scheme is able to group all the segments into one cluster based only on the temporal or (and) acoustic appearance. Thus, we employ the k-mean algorithm due to its simplicity and efficiency.

4. OUR METHOD

This section introduces a novel scheme to generate effective music descriptor for searching large scale music data collections. As depicted in Figure 2, the proposed system ($DMCM$) consists of two major components: 1) music preprocessing layer and 2) music concept modeling layer. In following sections, details of the system architecture and related algorithms are presented.

4.1 Music Preprocessing Layer

The key functionality of the music preprocessing layer is to calculate multiple acoustic features and project them onto an audio

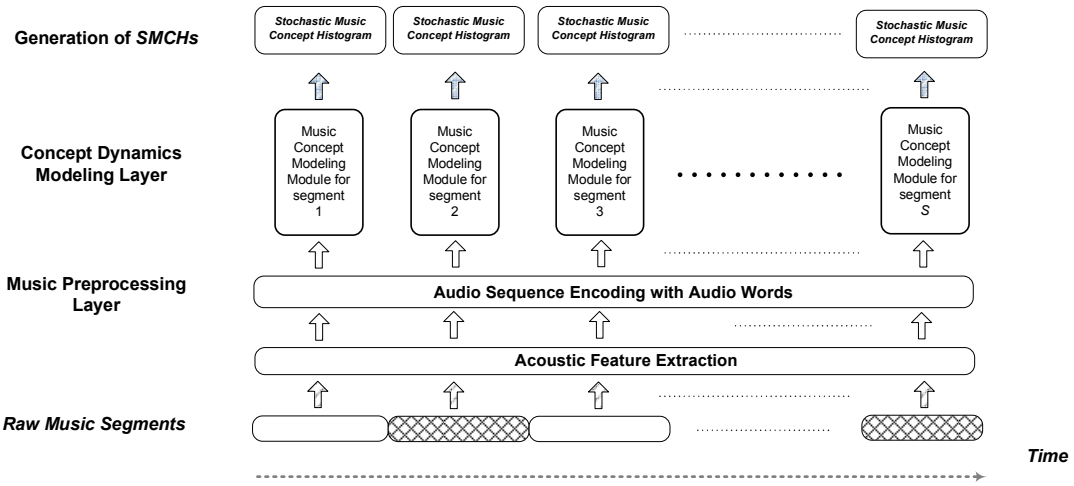

Figure 2: Architecture of the stochastic music concept modeling framework (DMCM). The output of this framework is a set of stochastic music concept histograms ($SMCH$s).

word from the codebook. Upon receiving the raw music signal, our system first decomposes it into multiple short segments with equal duration for the purpose of feature extraction. The number of segments S is preconfigured as a system parameter. Then, for each of the segments, based on four different kinds of features extracted, we find the most similar entry in the codebook and replace the index accordingly.

After this process, each input music is transformed into a sequence of indices or one-dimensional codes of audio words. The music document is now similar to a text document, which is a list of keywords. Under this paradigm, each music can be re-constructed by using the codebook of audio words.

4.2 Concept Dynamics Modeling Layer

The second layer of our system contains multiple music concept profiling modules developed based on generative learning principle. Each module corresponds to one music segment and outputs a stochastic music concept histogram - $SMCH$, describing the "probabilistic association" between the music and concepts at different levels. The set of $SMCH$s generated from this layer serves as signature of the input music for different MIR or content analysis applications. Before reviewing the architecture of this layer, we introduce the $SMCH$ music signature scheme.

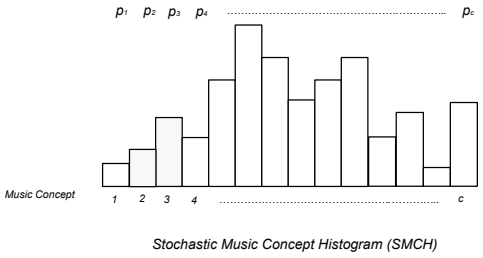

Figure 3: The structure of stochastic music concept histogram - $SMCH$

4.2.1 Stochastic Music Concept Histogram

Music concept representation is a crucial component of many music data management applications, including indexing, browsing, mining and retrieval. Here, we propose a unified scheme - stochastic music concept histogram ($SMCH$) as probabilistic representation of music documents. As demonstrated in Figure 3, it models the probability distribution of a music or a music segment over a set of predefined music concepts represented by audio words,

$$SMCH = \{p_1, p_2, ..., p_C\} \tag{4}$$

where $\sum_{c=1}^{C} p_c = 1$, C is the total number of musical concepts related to the music sequences, and p_c represents the probability of a given music object assigned to concept c. Since each $SMCH$ is a real-valued vector in metric feature space $\Re^C$, the distance between two music sequences ms_i and ms_j can be calculated using the normalized linear metric:

$$msim(SMCH_i, SMCH_j) = \frac{\sum_{c=1}^{C} |p_{c,i} - p_{c,j}|}{C} \tag{5}$$

where $msim(SMCH_i, SMCH_j) \in [0, 1]$. The distance function for $SMCH$ enjoys the following properties,

- Positive definite: Given two music sequences ms_i and ms_j, $msim(SMCH_i, SMCH_j) \geq 0$.
- Symmetry: Given two music sequences ms_i and ms_j, $msim(SMCH_i, SMCH_j) = msim(SMCH_j, SMCH_i)$.
- Reflexivity: Given a music sequence ms_j, $vsim(SMCH_j, SMCH_j) = 0$.
- Triangle inequality: Given three music sequences ms_i, ms_j and ms_l, $msim(SMCH_i, SMCH_j) \leq msim(SMCH_i, SMCH_l) + msim(SMCH_j, SMCH_l)$.

For the properties shown above, the feature space based on $SMCH$ is a C-dimensional metric space $\Re^C$, and each $SMCH$ is a numeric vector in this space.

4.2.2 Generative Model for Modelling Music Concepts

The main goal of the multiclass probability estimation module is to derive a probabilistic distribution between a set of predefined

Algorithm 1 Generative Process to Learn Concept Mapping M.

Description:

1: Obtain prior about matching via sampling an n-to-1 mapping M;
2: **for** each matched edge (n, c) in graph G, where $n = 1, ..., N$ and c = $1, ..., C$:
3: Sample latent concept: $z_c \sim N(0, I_d)$, min$\{d_A, d_T\} \geq d \geq 1$
4: Sample features for audio words
5: $\phi_A(aw_n) \sim N(W_A z_c + \mu_A, \Psi_A)$
6: Sample features for textual words
7: $\phi_T(t_c) \sim N(W_T z_c + \mu_T, \Psi_T)$
8: **for** each unmatched audio word n:
9: $\phi_A(av_n) \sim N(0, \sigma^2 I_{d_A})$
10: **for** each unmatched textual word c:
11: $\phi_T(t_c) \sim N(0, \sigma^2 I_{d_T})$

concepts (keywords) and a given music encoded with the audio codebook. In this study, we develop a generative model to facilitate the mapping process M based on canonical correlation analysis (CCA) [15].

As illustrated in Figure 4, the mappings M are treated as a weighted bipartite graph $G = (A, T, M, P)$. A and T are the feature vectors to describe audio words and textual keywords. For each audio word, we calculate the contextual feature - the co-occurrence counts of the word in neighboring music segments for audio words. Thus, we can generate a set of audio word feature vectors: $A = (av_1, ..., av_n)$, where $av_n \in \mathbb{R}^{d_A}$ for all n. To capture the text feature, the co-occurrence of different textual words is counted over the corpus. Hence we obtain a set of feature vectors $T = (t_1, ..., t_C)$ for C different music concept keywords. P is a set of the related probabilities (weights) estimated using the generative model. Given T and A, the learning process aims to derive an optimal mapping to translate words from the audio domain to a textual keyword (concept). It inputs feature matrices of the audio and textual words. **Algorithm 1** describes each step of the whole training process and the Figure 4 illustrates the abstract of the projection.

Different from the approach presented in [14], our algorithm samples an n-to-1 matching M from the prior mappings. We also restrict each audio word to occur only once but each text word can appear in multiple mapping pairs (line 1). The priors are assumed to be uniformly distributed. The approach underlying the learning algorithm is based on the probabilistic interpretation of CCA. It can be proved that the canonical correlation must be largest for the maximum likelihood estimates. Hence, a latent concept $z_c \sim N(0, I_d)$ can be generated for each matched edge (n, c) in the mapping M via sampling (line 3). I_d is a $d \times d$ identity matrix. Given the latent concept, samples of the audio words' feature vectors $\phi_A(aw)$ are drawn from a multidimensional Gaussian model with mean $W_A z_c + \mu_A$ and covariance Ψ_A (line 4 and line 5). W_A is a $d_A \times d$ matrix projecting concept z_c to the feature vector of audio words. A similar process is applied to the textual words (line 6 and line 7). Ψ_A and Ψ_T are covariance matrices to measure variations in two different domains. For unmatched words, we assume that the background distribution of audio and textual words can be used for mapping (line 8 - line 11).

4.2.3 Training Algorithm

Our training algorithm is developed on the principle of expectation-maximization (EM). It aims to find the maximum likelihood estimate via an iterative process. Given the statistical model introduced above, the log-likelihood of the training data is,

$$l(\theta) = log \sum_{M} p(av_n, t_c, M_{nc}|\theta) \tag{6}$$

where θ = (W_A, W_T, Ψ_A, Ψ_T) is a set of model parameters. The learning algorithm consists of two main steps,

- **E-step**: The posterior of all mappings is estimated by identifying the optimal n-to-1 mapping M in the graph G with weights P.
- **M-step**: Calculate the expected values of log-likelihood over all the possible pairs in the graph G. The weights P are updated with kernel Canonical Correlation Analysis (kCCA), which is trained on the best b mappings previously. In the first iteration of the EM estimation process, we do M-step computation using training matching pairs.

M-step: The goal of M-step is to optimize the values of θ so as to maximize the log-likelihood of all the mapping words from two different domains,

$$\max_{\theta} \sum_{(n,c) \in M} logp(v_n, t_c, M_{nc}|\theta) \tag{7}$$

This learning goal is equivalent to maximizing the likelihood of the probabilistic KCCA model [3]. For two sets of samples, CCA seeks to find basis vectors which can (i) map the elements of the samples into the multidimensional space and (ii) maximize the correlation between sets of the projections. Because CCA is effective only for linear relationships, kCCA is applied to first project each data point into a multidimensional space with higher dimensionality, and CCA analysis is then carried out in the new feature space. Similar to kernel PCA [26], two kernels K_A and K_T are defined over A and T. With the kernels, the related function that we need to optimize is given by,

$$\max_{w_a, w_t} \frac{w_a^T K_A K_T w_t}{\sqrt{w_a^T K_A^2 K_T^2 w_t}} \tag{8}$$

In order to avoid trivial learning, the process is regularized by introducing controlling the flexibility of the project. The partial least squares (PLS) is applied to penalized the norms of associated weights. Based on [4], a standard eigen problem can be obtained

$$\max_{w_a, w_t} \frac{w_a^T K_A K_T w_t}{\sqrt{w_a^T K_A^2 K_T^2 w_t}} \tag{9}$$

E-step aims to calculate the expected value (posterior) over all the possible matching pairs in the graph G. It is easy to prove that the related computing process is P-complete [38]. Since the hard EM only needs to compute the b best mapping under the current model, we have:

$$M^{'} = \arg\max_{M_{1:b}} \log p(V, T, M|\theta) \tag{10}$$

where θ is current model parameter estimated for kCCA. The optimization process is casted as a maximum weighted bipartite matching problem. The goal is to project the vectors in two domains (audio and text) onto the latent space. Our system uses the Euclidean distance function to quantify similarity - $p_{a,t}$ between the audio word a and textual word t. Meanwhile, $p_{a,t}$ is interpreted as the matching probability or edge weight in the graph G for different pairs (a,t).

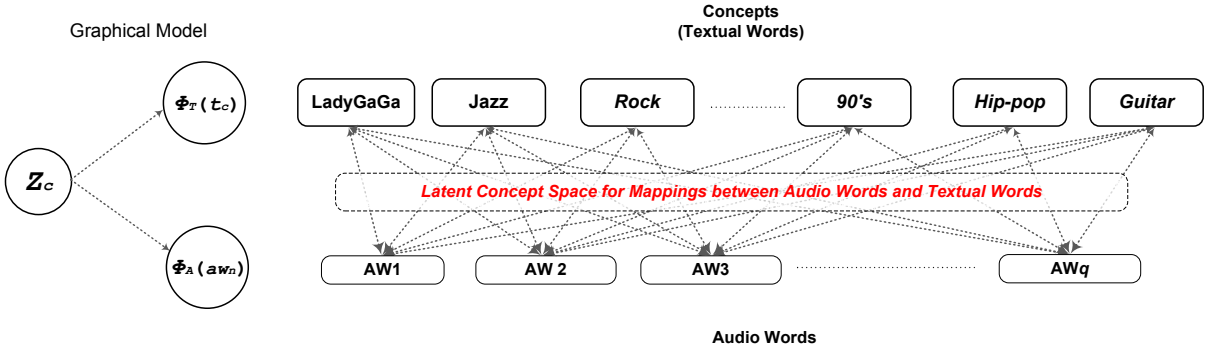

Figure 4: The goal of generative model is to derive the optimal mappings between audio words and various music concepts represented by textual words.

Algorithm 2 $SMCH$s Computation

Description:

1: Divide the input song to S segments
2: Feature extraction over each music segment
3: Encode music with codebook
4: Calculate $SMCH$ for each segment using music concept modeling module in concept dynamics modeling layer
5: Output a set of $SMCH$s

$$p_{a,t} = \|K_A(v_a)w_a - K_T(v_t)w_t\| \quad (11)$$

Using the equation, we calculate the matching probabilities for all possible mappings and find the best b pairs. Once the process is completed, we add the results into the kCCA learning examples and re-run the M-step. Our EM training procedure is similar to bootstrapping. Through iterations, the number of edges in graph G increases gradually. When the EM process stops, the matching probabilities obtained are used to compute $SMCH$.

4.3 Music Signature Generation with DMCM

The goal of $DMCM$ is to derive a set of $SMCH$s to serve as signature of a given music. The $SMCH$s can be applied in different MIR tasks. Using the system architecture introduced in the previous sections, we start with a training database where the codebook of audio words is generated. In the training stage, we process the songs in the database, extract features for each music and calculate the audio word codebook. Then, using **Algorithm 1** and the EM training procedure introduced above, we construct the concept dynamics modeling layer in our framework.

After the training phase, we proceed to compute the music signatures. The basic procedure is shown in **Algorithm 2** and consists of five main steps. For a given music item, the system initially partitions the input song into S segments (line 1). After that, the feature extraction procedure generates different kinds of features using the techniques described in Section 3 (line 2). Next, we apply the features to encode the input music sequence (line 3). A set of $SMCH$s are calculated using the music concept modeling module in the second layer, one per generative model:

$$MSIG = \{SMCH_1, SMCH_2, ..., SMCH_S\} \quad (12)$$

where $SMCH_s$ is a set of the probabilities (weights) - P learned with the generative model for segment s.

5. EXPERIMENTAL CONFIGURATION

This section describes in detail our experiment configuration to facilitate large scale performance evaluation and comparison. First, an introduction of the evaluation metrics and methodology is given in Section 5.1. Next, Section 5.2 presents details of two different testbeds used in the empirical study. After that, the competing methods that are included in the performance comparison are introduced in Section 5.3. All the music descriptor generation schemes evaluated here have been fully implemented and tested on a Pentium (R) D, 3.20GHz, 1.98 GB RAM PC running the Windows XP operating system. The number of music segments in $DCMC$ is set to 20.

5.1 Evaluation Metrics and Methodology

Music signature generation is one of the most fundamental components in various kinds of MIR applications. In order to conduct a comprehensive performance comparison of different schemes, our proposed system and the competitors are tested and compared on three MIR related tasks. They are,

- Task I - Music tagging: For a given music sequence, how accurately different systems determine a set of recommended tags. We examine the quality of the tag sets with different number of tags (top 5 tags, top 10 tags, top 15 tags and top 20 tags). When applied to Task I, our proposed $DMCM$ first outputs a set of $SMCHs$, then the concept keywords (tags) with top k weights are selected as the tagging result.

- Task II - Music search (Content-based Music Retrieval): Using the music signature extracted from a query example, the system retrieves a list of music pieces from the database that are most similar. Here, the Euclidean distance function is used to quantify the similarity between two pieces of music. We examine the list of top-k results. k is set to 10 and 20.

- Task III - Music clustering: Based on the signature assigned to music clips, automatically cluster music documents. We apply the k-Means clustering algorithm due to its simplicity and efficiency [11, 22].

Two evaluation metrics are used for the music annotation task (Task I). They are mean per-tag precision and recall. The top 5, 10, 15, 20 and 25 tags are generated by the various systems for comparison. The per-tag recall and per-tag precision are formally given by

$$Precision = \frac{|t_{TP}|}{|t_{GT}|} \qquad Recall = \frac{|t_{TP}|}{|t_A|} \tag{13}$$

where $|t_{GT}|$ is the number of songs annotated with the tags in the human-generated "ground truth" annotation and t_{TP} is the number of music annotated correctly with the tags. A detailed explanation on how the "ground truth" information is generated will be given shortly.

To measure the performance of the various systems for music search (Task II), the mean average precision (MeanAP) and the area under the receiver operating characteristic curve (AROC) are adopted as assessment metrics. For comparing the performance of various clustering methods (Task III), the metric we used is the F-Measure F. The formula is given by

$$F = \frac{P \times R}{P + R} \tag{14}$$

where,

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \tag{15}$$

TP is the true positive ratio, FP is the false positive ratio, and FN is the number of false negatives. In addition, to ensure result quality, tenfold cross validation is used to calculate the classification and clustering accuracy. This means the whole dataset is divided into ten disjoint subsets of (approximately) equal size. For testing, we train the algorithms on nine of the ten subsets and the remaining one is used for testing, each time leaving out a different subset. The process above is repeated for each system tested in our experiment study.

5.2 Music Testbeds

The test collections play a very important role in the empirical comparison study for MIR research. To ensure accuracy and fairness of the empirical results, we carefully develop or select two different test collections. Their details are as follows.

- TSI - This is the Computer Audition Lab 500-Song (CAL 500) data set developed by the CAL group [34, 35]. The collection consists of 500 modern western music documents performed by 500 different artists. Overall there are 174 'musically-relevant' tags categorized into six different semantic groups. They are mainly about instrumentation, vocal characteristics, genre, emotion, solo and usage terms. For this dataset, we use those six groups as high level musical concepts to train our statistical model. The text corpus used can be found in [1].
- TSII - As the size of the CAL500 data set is relatively small, we develop another test collection called TSII. It contains 6000 popular music items downloaded from Youtube or selected from the first author's CD collection. They are performed by 110 different singers, including 55 females and 55 males (such as Van Morrison, Michael Jackson, Elton John, Kylie Minogue, Madonna, Jennifer Lopez, Faith Hill, Lady Gaga etc.). We extract the lyrics of the songs and articles on the singers from Wikipedia and Last.fm as text corpus. 12 amateur musicians who are familiar with various music taxonomy and concepts were hired to create the ground truth for the tags in this collection. The ground truth was generated by attaching a tag to a music item if at least three people assigned the tag to the song. In cases where the respondents could not reach an agreement on a tag assignment, a similar resolution methodology as the one used in generating CAL500 is applied. At the end of the process, we obtain a total of 320 tags across 8 different categories. They are instrumentation, emotion, country, time, genre, vocal characteristics, speed, solo and usage terms.

Both TSI and TSII are used for testing the performance of different systems. TSII is applied for studying the performance of various signature generation schemes on music clustering and music search. For the purpose of acoustic feature extraction, we extract the audio tracks and convert them to 22050Hz, 16-bit, mono MP3 audio documents to ensure recording quality. The average length of the music clips is 120 seconds, the maximum length is 200 seconds and the shortest is about 100 seconds.

5.3 Competitors for Performance Comparison

For task I (music tagging), we compare the performance of our system against three state-of-the-art approaches including MMTagger [29], Autotagger [12, 5] and MSML [35, 34]. Acoustic feature considered by MSML is Mel-frequency cepstral coefficient (MFCC). Autotagger is evaluated based on three feature sets including MFCC delta, afeats and bfeats [2]. For MMTagger, we consider five low level feature configurations (timber features denoted by TF, rhythm features denoted by RF, spectral features denoted by SF, melody features denoted by MF and timber features+rhythm features+spectral features+melody features denoted by ALL.). Autotagger(MFCC delta), Autotagger(afeats) and Autotagger(bfeats) denote Autotagger with MFCC delta, afeats and bfeats respectively. MMTagger(TF), MMTagger(SF), MMTagger(MF), MMTagger(RF), MMTagger(ALL) denote our proposed model with timbral features, spectral features, rhythmic features, melody features and the combination of all four musical features.

To demonstrate the effectiveness of the music annotation generated by our system in search and clustering (Task II and Task III), we examine a wide range of methods for generating music descriptors, including CBIF [3], InMAF [30], DWCHs [16] and MARSYAS (denoted by MAR) [36].

6. EXPERIMENT RESULTS AND ANALYSIS

The music signatures generated by $DMCM$ are applicable to a wide range of MIR applications. This section presents a set of experiment studies to test and compare the performance of different systems on three MIR tasks including music tagging, music search and music clustering. How the systems are able to perform in a noisy environment is also evaluated.

6.1 On Music Tagging

In the first study, we test the performance of various tagging systems and $DMCM$ on the music tagging task. Table 2 and 3 report the empirical results of the systems with different feature configurations for the two testbeds. In this study, we test MMTagger and Autotagger with five and three feature settings respectively. The size of the tag set generated is configured to 10. The first row in both tables indicate how MSML performs on the tagging task using MFCC. Among the four systems tested, MSML demonstrates the lowest effectiveness. We believe there are two main reasons for the poor accuracy. Firstly, since music signals contain a rich set of acoustic features, effective music classification or annotation

[2] [35] gives a detailed description of the feature sets.

[3] CBIF denotes the method proposed in [31].

cannot be achieved by considering only a single low-level acoustic feature. Moreover, the system does not consider temporal dynamics inside the music signal. Compared to MSML, Autotagger demonstrates a performance improvement though the accuracy gain is very limited. It is interesting to observe that MMTagger(SF) and MMTagger(MF) based on our learning framework could suffer from lower accuracies than Autotagger with certain acoustic feature configurations. This suggests the importance of music signature quality. On the other hand, the performance of MMTagger(ALL) is better than Autotagger using any feature combinations. A significant effectiveness gain ranging from 5% to 15% can be observed as different features are gradually integrated. The results clearly demonstrate that it is critical to combine features properly to achieve tagging effectiveness. The results obtained with both datasets clearly show that for the music tagging task, $DMCM$ significantly outperforms all the existing systems. For example, Table 2 shows that in comparison to MMTagger(ALL), our system achieves a precision gain of 0.351 to 0.413 for the CAL500 dataset, and 0.327 to 0.352 for the TSII collection. We also obtain similar improvements with the other two evaluation metrics.

Model	Precision	Recall
MSML	0.144	0.064
Autotagger(MFCC delta)	0.281	0.131
Autotagger(afeats)	0.266	0.094
Autotagger(bfeats)	0.291	0.153
MMTagger(ALL)	0.351	0.291
MMTagger(TF)	0.256	0.141
MMTagger(SF)	0.241	0.137
MMTagger(MF)	0.226	0.131
MMTagger(RF)	0.289	0.150
$DMCM$	**0.413**	**0.326**

Table 2: Comparison of tagging accuracy on test collection CAL500(TSI).

Model	Precision	Recall
MSML	0.121	0.043
Autotagger(MFCC delta)	0.257	0.102
Autotagger(afeats)	0.239	0.073
Autotagger(bfeats)	0.268	0.139
MMTagger(ALL)	0.327	0.241
MMTagger(TF)	0.231	0.117
MMTagger(SF)	0.220	0.116
MMTagger(MF)	0.207	0.103
MMTagger(RF)	0.262	0.125
$DMCM$	**0.352**	**0.278**

Table 3: Tagging accuracy on test collection TSII.

6.2 On Music Retrieval

Effective music retrieval is important in coping with the fast growth of online music data. The next study is to evaluate the performance of $DMCM$ and the competitors on the music search task. For a given music query, we use different methods to extract descriptors (feature vectors) and the system being evaluated retrieves a list of music pieces from the database that are most similar; the similarity is quantified by the distance between two descriptors with the Euclidean (l_2) distance function. We use MeanAP and MeanAROC as metrics to compare the rank list.

One of our key hypotheses for this study is that the more discriminative the information that is packed into the music signature, the more accurate the retrieval results will be. The experiments here are intended to validate the hypothesis. Table 4 summarizes the experiment results with TSII. It is clearly shown that MARSYAS which combines the feature vectors linearly demonstrates the worst performance on both metrics. Meanwhile, because DWCHs captures only low level acoustic musical properties, the related performance improvement is very marginal. Compared to DWCHs, both InMAF and CBIF take into account multiple kinds of acoustic features, which brings about substantial improvement in search effectiveness. Specifically, we observe an increase of at least 20.1% in MeanAP and 23.4% in MeanAROC. The results also show clearly that $DMCM$ leads to much better performance than the competitors, delivering around 10.9% and 23.3% improvement in MeanAP over CBIF and InMAF. The findings suggest that it is impossible to achieve accurate MIR without effectively combining musical features.

Model	MeanAP	MeanAROC
MARSYAS	0.204	0.381
DWCHs	0.267	0.423
InMAF	0.321	0.522
CBIF	0.357	0.549
$DMCM$	**0.396**	**0.591**

Table 4: Music retrieval accuracy comparison on test collection TSII.

Model	F-Measure
MARSYAS	0.314
DWCHs	0.367
InMAF	0.409
CBIF	0.433
$DMCM$	**0.524**

Table 5: Music clustering accuracy comparison on test collection TSII (Task - Artist based clustering).

Model	MeanAP			MeanAROC		
	CL	NO	DR	CL	NO	DR
MARSYAS	0.204	0.161	21.3%	0.381	0.294	22.7%
DWCHs	0.267	0.213	20.4%	0.423	0.332	21.3%
InMAF	0.321	0.275	14.2%	0.522	0.441	15.5%
CBIF	0.357	0.308	13.5%	0.549	0.473	13.7%
$DMCM$	**0.396**	**0.355**	**10.2%**	**0.591**	**0.524**	**11.3%**

Table 6: Music retrieval robustness comparison on test collection TSII. Noise type - 50% volume amplification.

6.3 On Music Clustering

Accurate music clustering process has become increasingly important for different MIR applications. Our main objective in the third study is to examine the accuracy of clustering based on the music descriptors generated by $DMCM$ versus the other approaches. We perform the performance comparison with artist based music clustering.

Table 5 shows how the various systems perform on the clustering task. Clearly, our proposed $DMCM$ significantly outperforms all the other approaches in effectiveness. In particular, the results reveal that $DMCM$ enjoys an 21% increase in F-Measure over

Model	MeanAP			MeanAROC		
	CL	NO	DR	CL	NO	DR
MARSYAS	0.204	0.171	20.3%	0.381	0.292	23.1%
DWCHs	0.267	0.221	19.4%	0.423	0.335	20.7%
InMAF	0.321	0.287	15.3%	0.522	0.448	14.1%
CBIF	0.357	0.312	14.5%	0.549	0.482	12.1%
$DMCM$	**0.396**	**0.356**	**10.0%**	**0.591**	**0.533**	**9.7%**

Table 7: Music retrieval robustness comparison on test collection TSII. Noise type - 50% volume deamplification.

Model	MeanAP			MeanAROC		
	CL	NO	DR	CL	NO	DR
MARSYAS	0.204	0.154	24.3%	0.381	0.298	21.9%
DWCHs	0.267	0.205	22.9%	0.423	0.327	22.7%
InMAF	0.321	0.269	16.2%	0.522	0.446	14.6%
CBIF	0.357	0.302	15.5%	0.549	0.478	12.9%
$DMCM$	**0.396**	**0.350**	**11.7%**	**0.591**	**0.539**	**10.9%**

Table 8: Music retrieval robustness comparison on test collection TSII. Noise type - 10 second cropping.

Model	MeanAP			MeanAROC		
	CL	NO	DR	CL	NO	DR
MARSYAS	0.204	0.142	25.3%	0.381	0.298	22.9%
DWCHs	0.267	0.201	24.9%	0.423	0.333	21.2%
InMAF	0.321	0.272	15.2%	0.522	0.440	15.7%
CBIF	0.357	0.312	12.5%	0.549	0.471	14.2%
$DMCM$	**0.396**	**0.358**	**9.7%**	**0.591**	**0.531**	**10.1%**

Table 9: Music retrieval robustness comparison on test collection TSII. Noise type - 35dB SNR mean background noise.

Model	MeanAP			MeanAROC		
	CL	NO	DR	CL	NO	DR
MARSYAS	0.204	0.139	24.1%	0.381	0.299	21.5%
DWCHs	0.267	0.189	22.9%	0.423	0.336	20.5%
InMAF	0.321	0.242	14.2%	0.522	0.447	14.3%
CBIF	0.357	0.310	13.1%	0.549	0.472	13.9%
$DMCM$	**0.396**	**0.354**	**10.5%**	**0.591**	**0.533**	**9.7%**

Table 10: Music retrieval robustness comparison on test collection TSII. Noise type - 35dB SNR white background noise.

CBIF. The gain of $DMCM$ over InMAF is even more pronounced, averaging around 24%. We believe the performance gains arise because $DMCM$ generates an effective combination of information on different acoustic features and musical dynamics, resulting in discriminative and informative signatures for the music signals.

6.4 Robustness against Audio Noise

The human hearing system possesses the robust capability to sense and identify music even in a noisy environment. This capability is very useful for real life MIR applications, where the music signals may be mixed with noise signals. Typical examples include music recorded at live concerts or at other outdoor environments. However, very few existing schemes are designed to deal with inputs that are accompanied by distortions. Thus it is important for us to evaluate the robustness of the various systems against noises. We also wish to examine how different types of audio noise may influence the music retrieval process supported by $DMCM$ and other music signature generation methods. For this study, we pollute each query music with different kinds of audio distortions. Then we carry out experiments to test the corresponding retrieval performance of our system and the competitors. We run same set of tests in Section 6.2 on TSII. In this study, we consider five different distortion cases including 50% volume amplification, 50% volume deamplification, 10 second cropping, 35dB SNR mean background noise and 35dB SNR white background noise [4].

Tables 6-10 summarize the search accuracy of the five systems for the various audio distortion cases. CL and NO denote the search accuracies obtained under clean input and noisy input, respectively. DR is the search accuracy drop ratio between clean music input and input containing noise. From the results, we observe that generally all the systems suffer certain levels of accuracy loss with noisy input. However, $DMCM$ performs more robustly than the competitors over all the noise cases. For example, $DMCM$'s MeanAP experiences a 10.2% drop when the music inputs are polluted with 50% volume amplification. In comparison, the MeanAP ratio of InMAF and CBIF decrease about 14.2% and 13.5% respectively. Moreover, in the case of 35dB SNR mean background noise, $DMCM$ only loses 10.1% in MeanAROC whereas performance degradation of 15.7% and 14.2% are observed for InMAF and CBIF. For MARSYAS and DWCHs, the performance gaps are even much wider. Thus, we conclude that $DMCM$ emerges as the more robust scheme against noise and acoustic distortion.

7. CONCLUSION

As an enabling technology for large scale MIR, music signature generation has received a lot of attention in recent years, with many different proposed approaches. Notwithstanding that, the technology is still in its infancy as the reported effectiveness are existing schemes are generally poor. The main reasons for this stagnation include 1) the lack of advanced techniques to intelligently combine multimodal and temporal information and 2) the unavailability of a comprehensive classification scheme to systematically bridge the gap between different levels of semantics. In this paper, we report a novel framework called $DMCM$ that incorporates an advanced feature extraction scheme and a composite system architecture for comprehensive music signature generation. Our system architecture contains two basic modules - 1) music preprocessing module and 2) concept dynamics modelling module in the form of an advanced two-layer classification scheme. A comprehensive empiri-

[4]We use $SNR_{dB} = 10log_{10}\frac{Si}{No}$ to calculate the signal-to-noise ratio SNR_{dB}, where Si denotes the signal power, and No denotes the noise power in dB.

cal study involving two large music collections has been conducted to compare our framework with competing methods on various MIR tasks. The experiment results show that our framework leads to significant improvements in accuracy, robustness and scalability on the MIR tasks.

The current study can be extended in several directions for further investigation: One of biggest advangtages enjoyed by $DMCM$ is the use of multiple feature about both acoustic characteristics and temporal dynamics from music. This results in more comprehensive statistical models and hence a better modelling effectiveness. One question naturally raised is how much each of these factors contributes towards accuracy and robustness improvement. In the future, we plan to have a detailed study. Further, applying the method on data from other application domains would be also very interesting.

8. REFERENCES

[1] Cal500 data set annotation, 2007. `http://cosmal.ucsd.edu/cal/pubs/annotations.txt`.

[2] Nielsen company & billboard's 2011 music industry report. *Business Wire*, 5 January 2012.

[3] F. Bach and M. I. Jordan. *A Probabilistic Interpretation of Canonical Correlation Analysis*. Technical Report 688, Department of Statistics, University of California, Berkeley, 2005.

[4] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[5] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2), 2008.

[6] S. Bhattacharjee, R. D. Gopal, K. Lertwachara, and J. R. Marsden. Consumer search and retailer strategies in the presence of online music sharing. *J. of Management Information Systems*, 23(1), 2006.

[7] S. Bhattacharjee, R. D. Gopal, K. Lertwachara, J. R. Marsden, and R. Telang. The effect of digital sharing technologies on music markets: A survival analysis of albums on ranking charts. *Management Science*, 53(9), 2007.

[8] E. Coviello, A. B. Chan, and G. Lanckriet. Time series models for semantic music annotation. *IEEE Trans. on Audio, Speech & Language Processing*, 19(5), 2011.

[9] L. Daudet. Transients modeling by pruned wavelet trees. In *Proc. of International Computer Music Conference*, 2001.

[10] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2), 2003.

[11] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[12] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Proc. of NIPS*, 2007.

[13] H. Green. Kissing off the big music labels. *Businessweek*, 2004.

[14] A. Haghighi, P. Liang, T. Berg-Kirkpatrick, and D. Klein. Learning bilingual lexicons from monolingual corpora. In *Proc. of ACL*, 2008.

[15] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. *Canonical Correlation Analysis; An Overview with Application to Learning Methods*. Technical Report CSD-TR-03-02, Computer Science Dept. Royal Holloway, University of London, 2003.

[16] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proc. of ACM SIGIR*, 2003.

[17] W. Li, Y. Liu, and X. Xue. Robust audio identification for mp3 popular music. In *Proc. of ACM SIGIR*, 2010.

[18] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. of ISMIR*, 2000.

[19] L. Lu, S. H. Li, and J. Zhang. Content-based audio segmentation using support vector machines. In *Proc. of IEEE ICME*, 2001.

[20] L. Lu, D. Liu, and H. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Trans. Acoust., Speech, Signal*, 2006.

[21] S. Mallat. *A Wavelet Tour of Signal Processing*. Acadamic PressAcademic Press, 3rd edition, 2008.

[22] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[23] R. Miotto and N. Orio. A probabilistic model to combine tags and acoustic similarity for music retrieval. *ACM Trans. Inf. Syst.*, 30(2), May 2012.

[24] U. Nam and J. Berger. Addressing the same but different-different but similar problem in automatic music classification. In *Proc. of ISMIR*, 2001.

[25] C. Sanden and J. Zhang. Enhancing multi-label music genre classification through ensemble techniques. In *Proc. of ACM SIGIR*, 2011.

[26] B. Scholkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[27] A. Sheh and D. Ellis. Chord segmentation and recognition using em-trained hidden markov models. In *Proc. of ISMIR*, 2003.

[28] J. Shen, B. Cui, J. Shepherd, and K. Tan. Towards efficient automated singer identification in large music databases. In *Proc. of ACM SIGIR*, 2006.

[29] J. Shen, W. Meng, S. Yan, H. Pang, and X. Hua. Effective music tagging through advanced statistical modeling. In *Proc. of ACM SIGIR*, 2010.

[30] J. Shen, J. Shepherd, and A. H. H. Ngu. Towards effective content-based music retrieval with multiple acoustic feature combination. *IEEE Trans. on Multimedia*, 8(6), 2006.

[31] Y. Song and C. Zhang. Content-based information fusion for semi-supervised music genre classification. *IEEE Trans. on Multimedia*, 10(1), 2008.

[32] D. Turnbull, L. Barrington, and G. Lanckriet. Modeling music and words using a multi-class naïve bayes approach. In *Proc. of ISMIR*, 2006.

[33] D. Turnbull, L. Barrington, G. R. G. Lanckriet, and M. Yazdani. Combining audio content and social context for semantic music discovery. In *Proc. of ACM SIGIR*, 2009.

[34] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proc. of ACM SIGIR*, 2007.

[35] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. on Audio, Speech & Language Processing*, 16(2), 2008.

[36] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Processing*, 2002.

[37] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 2003.

[38] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8, 1995.

[39] B. Zhang, J. Shen, Q. Xiang, and Y. Wang. Compositemap: a novel framework for music similarity measure. In *Proc. of ACM SIGIR*, 2009.