CompositeMap: a Novel Framework for Music Similarity Measure

Bingjun Zhang[1], Jialie Shen[2], Qiaoliang Xiang[1], Ye Wang[1]
[1]School of Computing, National University of Singapore
[2]School of Information Systems, Singapore Management University
[1]{bingjun,xiangqiaoliang,wangye}@comp.nus.edu.sg, [2]jlshen@smu.edu.sg

ABSTRACT

With the continuing advances in data storage and communication technology, there has been an explosive growth of music information from different application domains. As an effective technique for organizing, browsing, and searching large data collections, music information retrieval is attracting more and more attention. How to measure and model the similarity between different music items is one of the most fundamental yet challenging research problems. In this paper, we introduce a novel framework based on a multimodal and adaptive similarity measure for various applications. Distinguished from previous approaches, our system can effectively combine music properties from different aspects into a compact signature via supervised learning. In addition, an incremental Locality Sensitive Hashing algorithm has been developed to support efficient retrieval processes with different kinds of queries. Experimental results based on two large music collections reveal various advantages of the proposed framework including effectiveness, efficiency, adaptiveness, and scalability.

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, Similarity Measure, Personalization, Browsing, Search, Recommendation

1. INTRODUCTION

Over the past decade, empowered by advances in networking, data compression and digital storage, modern information systems dealt with ever-increasing amounts of music data from various domain applications. Consequently, the development of advanced Music Information Retrieval (MIR) techniques have gained great momentum as a means to facilitate effective music organization, browsing, and searching. One of the typical examples is that an end user might issue a text-based query to search for music records performed by a particular artist.

As one of the most fundamental components for MIR applications, how to measure and model similarity between music items is an important yet challenging research question [5]. This is because music information can contain rich semantics and the related representations of low-level features are high-dimensional in nature. There has been intense research in this field and the solutions proposed so far can be generally classified into three independent families:

Metadata-based similarity measure (MBSM) - Text retrieval techniques are used to compare the similarity between the input keywords and the metadata around music items [1, 2]. The keywords could include the title, author, genre, performer's name, etc. The main disadvantage is that high-level domain knowledge is essential for creating the metadata and music facet (timbre, rhythm, melody, etc.) identification. It would be very expensive and difficult to represent this information using human languages.

Content-based similarity measure (CBSM) - Extracting temporal and spectral features from music items for use as content descriptors has a relatively long history. It can be used as musical content representation to facilitate applications [8, 11, 20] for searching similar music recordings in a database by content-related queries (audio clips, humming, tapping, etc.). However, the previous research on music content similarity measures focused mainly on a single aspect similarity measure or a holistic similarity measure. In single aspect similarity, only limited retrieval options are available. With this paradigm, end users have less flexibility to describe their information need. On the other hand, for the holistic similarity measure [8], high dimensional feature space results in slow nearest neighbor finding or complex probability model comparison (Gaussian Mixture Models, etc.). This is impractical for a commercial size database containing millions of songs. In addition, either the single aspect or holistic similarity is not flexible enough to adapt with the users' evolving music information needs or retrieval context. Even worse, no personalization of the similarity measure is allowed.

Semantic description-based similarity measure (SDSM) - It is a proposed paradigm originally developed for image and video retrieval [17]. The basic idea is to annotate each music item in a collection using a vocabulary of predefined words. Music can be represented as a se-

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

Table 1: Summary of the main categories for music similarity measure.

Type of Measure	Physical Representation	Semantic Related	Computational Metric	Indexing Structure	Personalization
MBSM	Textual keywords	Yes	Inner product of document vectors	Inverted list, Hashing	No
CBSM	Feature vector, probability models	No	Mahalanobis, Euclidean, KL divergence, etc.	High-dimensional indexing tree, linear search	No
SDSM	Multinomial distribution of a bag of keywords	Yes	KL divergence, etc.	Linear search	No
CompositeMap	FMSV + DV	Yes	Inner produce of document vectors, Euclidean	Hybrid: inverted list + iLSH	Yes

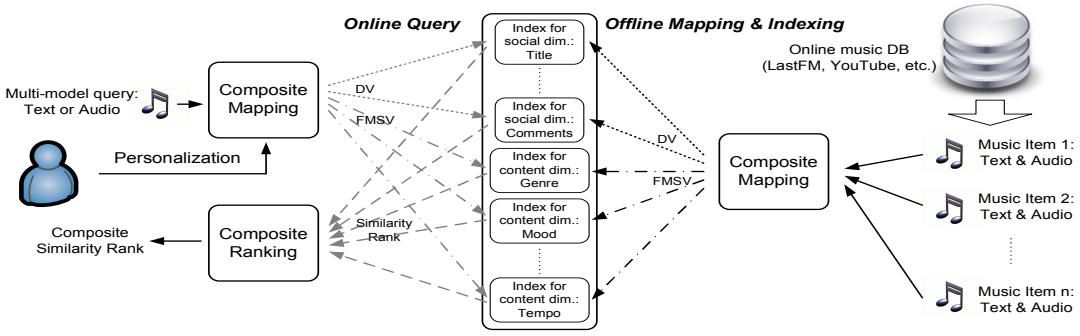

Figure 1: The conceptual framework of CompositeMap for effective multi-model music similarity measure.

mantic multinomial distribution over the vocabulary. The Kullback-Leibler (KL) divergence [17] is used to measure the distance between the multinomial distributions of the query and a music record. The same problem of limited description capability of human languages also exists in SDSM, since a limit number of keywords are used to describe music content. The large vocabulary (easily hundreds of keywords) results in low efficient indexing and ranking, thus unaffordable response time.

Table 1 summarizes the existing work for music similarity measures. We can see that the similarity between two musical items can be measured from multiple dimensions in terms of title, author, genre, melody, rhythm, tempo, instrumentation, etc. These dimensions are not independent. Different emphasis on each dimension will result in different similarity between the same two music items [5, 7].

Motivated by the above observations, we propose a novel framework for multifaceted music similarity measure. The key innovation of this study is to design and develop a comprehensive representation of music items called CompositeMap. Using CompositeMap, music content-related dimensions (genre, mood, tempo, melody, etc.) are modeled as Fuzzy Music Semantic Vectors (FMSVs) and social information-related dimensions are described as Document Vectors (DVs). Adaptive similarity between music items can be measured using each individual musical dimension, or by any combination of those dimensions based on user's preferred music information need in each search process. To the best of our knowledge, this is the first method to seamlessly integrate the metadata, content, and semantic description-based similarity measure into a single framework. Moreover, personalization of music similarity can be easily enabled in related applications, where end users with certain information needs in a particular context are able to specify their desirable dimensions to retrieve similar music items. By better modeling users' search targets based on personalized music dimensions, we can create more comprehensive similarity measures and improve the music retrieval accuracy. Compared with SDSM, high-level semantic concepts of a common music facet are grouped into a single music dimension. For example, tens of genre classes are grouped into a genre dimension. Therefore, each music dimension contains a many fewer components than the whole vocabulary in SDSM. This advantage can provide more efficient music query and ranking in large databases. In addition, we also developed an indexing structure based on the LSH algorithm [3] to further improve the efficiency of the retrieval process. We implemented a showcase system of keyword and content-based music searching based on YouTube music data. Evaluation results based on two large-scale data sets collected from YouTube demonstrate the various advantages of the proposed scheme for music similarity measure.

The remainder of this paper is organized as follows. In Sec. 2, we give a detailed introduction of the proposed framework. Sec. 3 describes the experimental setup. Evaluation results are discussed in Sec. 4, which is followed by our conclusion in Sec. 5.

2. THE FRAMEWORK

To address the problem raised in Sec. 1, a novel framework is developed to facilitate effective and flexible music information retrieval. As illustrated in Fig. 1, this multilayer structure consists of two major functionality modules: music signature generation and indexing. In this approach, we propose a compact music signature, called Fuzzy Music Semantic Vector (FMSV). FMSV can explicitly describe each music content-related dimension in a structured and human-understandable way. A conceptual diagram is presented in Fig. 2. By further representing the social information dimensions as Document Vectors (DVs) [13], a novel scheme called CompositeMap is proposed to map multiple and cross-modal music dimensions into a unified representa-

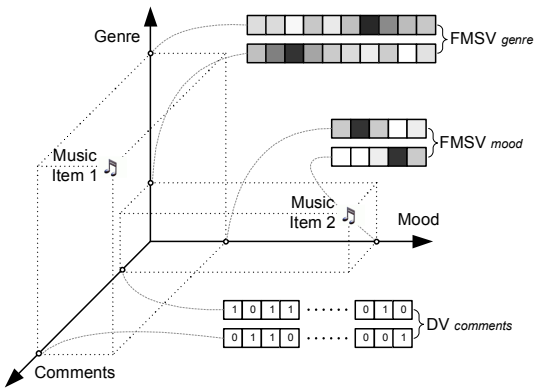

Figure 2: Illustration of music space with exemplar music dimensions: genre, mood, and comments.

tion. These music dimensions further span a music space, in which adaptive music similarity can be measured between any two music items. Each dimension can be indexed separately using incremental Locality Sensitive Hashing (iLSH) or inverted list in the indexing module. This framework facilitates flexible retrieval by involving user's personalization of preferred musical facets.

2.1 Fuzzy Music Semantic Vector - FMSV

To represent each music content-related dimension, we design a new representation - Fuzzy Music Semantic Vector (FMSV). We define the i-th music dimension as a FMSV, $\mathbf{f}_i = [f_{i,1} \ ... \ f_{i,N_i}]^T$, $0 \leq f_{i,j} \leq 1$, $1 \leq j \leq N_i$. For music dimension related to classification (genre, mood, etc.), N_i is the number of classes in the i-th music dimension and $f_{i,j}$ indicates the probability that the music item belongs to the j-th class of the i-th music dimension. For other content-related music dimensions (tempo, melody, etc.), N_i is the number of normalized values, $f_{i,j}$, of that music dimension[1]. We further employ Document Vectors (DVs) [13], $\mathbf{d} = [d_{i,1} \ ... \ d_{i,N_i}]^T$, to model each social information-related music dimension, where $d_{i,j} \in \{0, 1\}$ and $d_{i,j} = 1$ indicates the j-th word in a dictionary exists in the i-th music dimension. All music dimensions are represented as real vectors with different number of components (we notate both FMSV and DV by $\mathbf{f}$ from here). Based on FMSV and DV, a music item can be represented as the set of all music dimensions, $\mathcal{M} = \{\mathbf{f}_i | 1 \leq i \leq N\}$. Examples of FMSVs and DVs for different music dimensions are illustrated in Fig. 2, in which the positions on genre, mood or comments axis illustrate the different vector values of FMSVs or DVs.

As discussed in [5, 6, 12], music semantic concepts are usually represented by rigid human labels, e.g., classical for a genre type. However, music concepts are fuzzy in nature. Humans do not always agree on a single label for the same music item. Besides, human labels may be too broad to compare the similarity between two music items. These observations imply that human labels are not good representations of musical semantics when measuring music similarity.

We propose FMSV to represent each high-level music dimension. It represents the probabilities that a music item belongs to each class of that dimension or the most probable values that dimension has. It reveals the fuzzy nature (uncertainty) of human's perception, which is a more accurate representation of human's musical opinions. FMSVs are well structured and human understandable, which allows direct interaction between users and the music signature. FMSVs are efficient to compute, as the FMSV of each music dimension has many fewer components (e.g., ≈ 10 in genre [8]) than existing audio features (e.g., ≈ 100 in Sec. 2.3.1). The human-understandable nature allows FMSVs to be customized to represent different sets of classes in various applications. These properties not only make FMSVs effective to represent music but also flexible to use and efficient to index in music retrieval applications.

[1]Lower-case bold letters notate column vectors. Italic letters notate scalars. Calligraphic upper-case letters notate sets.

2.2 Adaptive Music Similarity Measure

With the above description, we can see that FMSVs and DVs satisfy properties of Euclidean metric, i.e., symmetry, and triangle inequality. The distance between two music items $\mathcal{M}^j$ and $\mathcal{M}^k$ in the i-th music dimension $\mathbf{f}_i$ can be measured by the normalized Euclidean metric as:

$$\text{dis}(\mathbf{f}_i^j, \mathbf{f}_i^k) = \sqrt{\frac{1}{N_i} \sum_{l=1}^{N_i} (f_{i,l}^j - f_{i,l}^k)^2} \qquad (1)$$

where N_i is the number of components in the i-th music dimension, and $\text{dis}(\mathbf{f}_i^j, \mathbf{f}_i^k) \in [0, 1]$.

With all the N music dimensions, we can span a music space in which musical items can be characterized by clear and musically meaningful concepts. The music space can be personalized by users into a subspace, $\mathcal{P} = \{(p_i, w_{p_i}) | 1 \leq p_i \leq N, 1 \leq i \leq N_\mathcal{P}, N_\mathcal{P} \leq N\}$, by choosing the most interesting dimensions p_i and specifying their preferred weights, $w_{p_i} \in [0, 1]$. In $\mathcal{P}$, a personalized music similarity measure between two music items $\mathcal{M}^j$ and $\mathcal{M}^k$ is defined as:

$$\text{Sim}(\mathcal{M}^j, \mathcal{M}^k; \mathcal{P}) = \sum_{i=1}^{N_\mathcal{P}} \frac{w_{p_i} \cdot \alpha}{1 + \exp\left(\text{dis}(\mathbf{f}_i^j, \mathbf{f}_i^k)\right)} - \beta \qquad (2)$$

where α and β are normalizing factors. If $\alpha = 2\frac{e+1}{e-1}$ and $\beta = \frac{2}{e-1}$, $\text{Sim}(\mathcal{M}^j, \mathcal{M}^k; \mathcal{P}) \in [0, 1]$.

2.3 CompositeMap: From Rigid Acoustic Features to Adaptive FMSVs

In order to map low-level acoustic features into FMSVs for content related music dimensions (Fig. 3) and to map text information into DVs for social information related music dimensions [13], a supervised learning based scheme, called CompositeMap, is developed to generate a new feature space. During the mapping of FMSVs, the most effective heuristic feature sets are selected to ensure reasonable prediction accuracy. Then a feature selection algorithm is applied to reduce dimensionality. Efficient multi-class probability estimation is then conducted to generate FMSVs. For the mapping of non-classification related FMSVs, we directly calculate their most probable values. For example, for tempo and melody we compute the beat histogram and pitch histogram as their FMSVs, respectively.

2.3.1 Audio Feature Extraction and Selection

In this framework, we consider various audio features. Based on their musical meanings, we categorized the employed features as follows:

Timbral features represent the timbral texture of musical sounds. Timbral features are calculated based on the

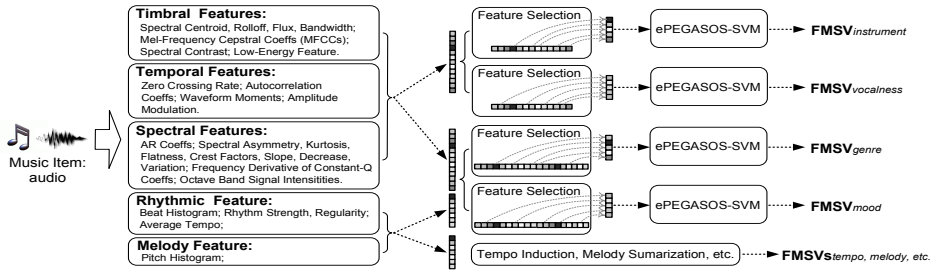

Figure 3: CompositeMap: from rigid acoustic features to adaptive FMSVs.

magnitude spectrum of short time Fourier transform (STFT) and include: *Spectral Centroid*, *Rolloff*, *Flux*, *Low-Energy feature* [19]; *Spectral Contrast* [12]; *Mel-Frequency Cepstral Coefficients*(MFCCs) [10]. The total dimensionality is 20.

Temporal features represent musical properties based on time domain signals. They include: *Zero Crossing Rate*; *Autocorrelation Coefficients*; *Waveform Moments*; *Amplitude Modulation* [12]. The total dimensionality is 15.

Spectral features complement timbral features in representing musical characteristics by spectra. They include *Auto-regressive (AR) features*; *Spectral Asymmetry, Kurtosis, Flatness, Crest Factors, Slope, Decrease, Variation*; *Frequency Derivative of Constant-Q Coefficients*; *Octave Band Signal Intensities* [12]. The total dimensionality is 20.

Rhythmic features represent musical timing characteristics of a music item. They include: *Beat Histogram* [19]; *Rhythm Strength*, *Regularity* and *Average Tempo* [12]. The total dimensionality is 12.

Melody features summarize the melody content of a music item. We employ *Pitch Histogram* proposed in [19] as melody features. The total dimensionality is 48.

As noticed, low-level audio features contain many more components (115) than FMSVs. High dimensionality of existing audio features has restricted the applicability of content-based music retrieval in large collections. A feature selection algorithm (Alg. 1) based on localized prediction error [14] is applied to reduce the dimensionality of the combined features while maintaining relatively good prediction accuracy. In Alg. 1, t_e is the stopping threshold of the decrease in prediction accuracy. Feature selection can significantly reduce the complexity of on-line prediction at an affordable cost of higher off-line computation.

Algorithm 1: Feature selection algorithm.

Input: Initial feature set, $\mathcal{F} = \{c_i | 1 \leq i \leq N_d\}$; training and testing databases, DB_{tr} and DB_{te};
Output: Selected feature set, $\mathcal{F}^s = \{c_i^s | 1 \leq i \leq N_d^s\} \subset \mathcal{F}$;
Description:
1: Train SVM using ePEGASOS on DB_{tr} with features $\mathcal{F}$;
2: Compute the localized prediction error e_o on DB_{te};
3: Let $\mathcal{F}^s := \mathcal{F}$;
4: **repeat**
5: Train the classifier on DB_{tr} with feature set $\mathcal{F}^s$;
6: **for** $i = 1$ to N_d^s **do**
7: Compute the localized prediction error, e_i, by keeping c_i^s constant as its mean on DB_{te}; [14]
8: **end for**
9: Set $r := \arg\min_i \{e_i | 0 \leq i \leq N_d^s\}$;
10: Set $\mathcal{F}^s := \mathcal{F}^s \setminus \{c_r^s\}$;
11: **until** $e_r - e_o > t_e$
12: **return** $\mathcal{F}^s$.

2.3.2 Multi-class Probability Estimation

In this study, Support Vector Machines (SVMs) are used for the purpose of multi-class probability estimation. Based on an efficient SVM training algorithm, PEGASOS [16], for binary classification problems with only binary label output, we propose an extended version, ePEGASOS, to support multi-class SVMs with probability estimates. The running time of PEGASOS has inverse dependency on the training dataset. Based on our experimental results, we show that ePEGASOS reveals the same desirable property: training a better generalized SVM with less run time on a large database.

PEGASOS is an iterative algorithm for optimizing SVM $\mathbf{w}$ on a given training set $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$. Each iteration involves a stochastic gradient descent step and a projection step. By giving T, the number of iterations, and k, the number of samples used for calculating sub-gradients at each iteration, PEGASOS optimizes the following unconstrained training error function with a penalty term for the norm of SVM being learned:

$$f(\mathbf{w}; \mathcal{A}_t) = \frac{\lambda}{2} \parallel \mathbf{w} \parallel^2 + \frac{1}{k} \sum_{(\mathbf{x},y) \in \mathcal{A}_t} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle\} \quad (3)$$

where $\mathcal{A}_t \subset \mathcal{S}$ is formed by k samples selected i.i.d. from $\mathcal{S}$ at each iteration t. $\mathbf{w}$ is initialized as zero vector and is updated at each iteration t as follows:

$$\mathbf{w}_{t+\frac{1}{2}} = (1 - \eta_t \lambda)\mathbf{w}_t + \frac{\eta_t}{k} \sum_{(\mathbf{x},y) \in \mathcal{A}_t^+} y\mathbf{x} \quad (4)$$

$$\mathbf{w}_{t+1} = \min\{1, \frac{1/\sqrt{\lambda}}{\parallel \mathbf{w}_{t+\frac{1}{2}} \parallel}\}\mathbf{w}_{t+\frac{1}{2}} \quad (5)$$

where $\eta_t = 1/(\lambda t)$ is the learning rate, $\mathcal{A}_t^+$ is the set of samples on which $\mathbf{w}$ has non-zero training error. To train kernel SVMs, $\mathbf{w}_t$ can be calculated as $\mathbf{w}_t = \sum_{i \in \mathcal{I}_t} \alpha_i \mathbf{x}_i$, where $\mathcal{I}_t \subset \{1, ..., m\}$. Then $\langle \mathbf{w}_t, \mathbf{x}_t \rangle = \sum_{i \in \mathcal{I}_t} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_t \rangle$ and $\parallel \mathbf{w}_t \parallel^2 = \sum_{i,j \in \mathcal{I}_t} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Once $\mathbf{w}$ is trained, we employ the method proposed in [15] to estimate the probability that a unknown sample belongs to class $y = 1$ as:

$$r^+ = \frac{1}{1 + \exp(A\langle \mathbf{w}, \mathbf{x} \rangle + B)} \quad (6)$$

where A and B are estimated scalars by minimizing the error function using the training data and their decision values.

Based on the above binary class SVM with a probability estimate, we further employ the generalized Bradley-Terry model [9] to extend binary-probability PEGASOS to support multi-class probability estimate. In K class classifica-

Table 2: The hierarchy of the tunable database, including 3020 music items. The number of collected music items is indicated after each class label. Some music items are shared by multiple music dimensions.

Genre - 1163		Mood - 778	Vocalness - 1968	Instrument - 1392			
Classical - 112	Jazz - 125	Passionate - 156	Nonvocal - 493	Brass - 310	Woodwinds - 382	Percussion - 356	Strings - 344
Country - 118	Rock - 122	Rollicking - 158	Male - 494	Trombone - 103	Flute - 124	Piano - 129	Violin - 111
Baroque - 121	Romantic - 124	Literate - 160	Female - 484	Trumpet - 104	Clarinet - 125	Snare - 100	Cello - 100
Electronic - 130	Blues - 105	Humorous - 152	Mixed - 497	Tuba - 103	Saxophone - 133	DrumKit - 127	Guitar - 133
HipHop - 106	Metal - 100	Aggressive - 152					

tion problems, *one-against-the-rest* scheme is employed to decouple the multi-class problem into K binary classification problems. The Bradley-Terry model is formulated as:

$$\begin{aligned} &\min_p \ -\sum_{i=1}^{K} (r_i^+ \log(\tfrac{p_i}{\sum_{j=1}^{K} p_j}) + r_i^- \log(\tfrac{\sum_{j=1,\neq i}^{K} p_j}{\sum_{j=1}^{K} p_j})) \\ &\text{subject to } \sum_{j=1}^{K} p_j = 1, 0 \leq p_j, j = 1, ..., K. \end{aligned} \quad (7)$$

to derive probability $p_j, j = 1, ..., K$, that a unknown sample belongs to the j-th class. Then the FMSV is formed as $\mathbf{f} = [p_1 \ ... \ p_K]^T$.

2.4 iLSH Indexing Structure

Inspired by the inverted index used in text retrieval, we develop a hybrid indexing framework to index each music dimension separately by its most suitable algorithm in order to build an overall efficient index for the whole music space.

Music dimensions represented by FMSVs are indexed by a proposed incremental Locality Sensitive Hashing (iLSH). The original LSH was proposed in [3]. It supports fast nearest neighbor search in high dimensional space with sub-linear time, which is critical for large music database of millions of tracks. To better suit our indexing solution to real application scenarios, such as on YouTube or Last.fm, where new music samples are periodically added into existing indexes, we propose an iLSH algorithm (Alg. 2) to efficiently update the existing index structure without the need to recompute the whole index from scratch. iLSH is desirable especially in a large database. In Alg. 2, the difference function for two sets of parameters is defined as:

$$\text{dif}(\Theta, \Theta') = |\frac{k - k'}{k}| + |\frac{L - L'}{L}| \quad (8)$$

where $\Theta = \{k, L\}$; k is the number of hashing functions chosen to construct a hash table; L is the number of hash tables [3]. γ and t_Θ are two update thresholds. Inverted list [13] is used for music dimensions represented by DVs.

Based on the above indexing approach, the time complexity of online query is sub-linear, $O(N_\mathcal{P} \cdot N_d \cdot n^{1/c^2})$, where $N_\mathcal{P}$ is the number of personalized music dimensions, N_d is the highest number of components in all those music dimensions, n is the total number of music items in the database, and c is the factor for approximate nearest neighbor finding in iLSH. In a commercial system, $N_\mathcal{P}$ and N_d will be small (≈ 10), while n is over a million. $c > 1$, can be tuned to trade off between query accuracy and efficiency.

2.5 Composite Ranking

Based on users' personalization input $\mathcal{P}$ discussed in 2.2, nearest music items $\mathcal{M}^{r_i}$ to the query $\mathcal{M}^q$ are retrieved by iLSH in each of the personalized dimensions, $(p, w) \in \mathcal{P}$. The adaptive music similarity measure, $\text{Sim}(\mathcal{M}^q, \mathcal{M}^{r_i}; \mathcal{P})$, is then used to rank all the returned items. As $\text{Sim}(\mathcal{M}^q, \mathcal{M}^{r_i}; \mathcal{P})$ is of accumulative nature, music items that are near to the query in more music dimensions are more likely ranked top.

Algorithm 2: Incremental Locality Sensitive Hashing.

Input: Initial set of samples $\mathcal{S}$;
Additional online sets of samples $\mathcal{S}_i$, $1 \leq i \leq N_s$;
Output: Index $\mathcal{H}$ for all samples;
Description:
1: Compute the parameter set Θ of the hashing structure $\mathcal{H}$ based on $\mathcal{S}$;
2: Hash $\mathcal{S}$ into $\mathcal{H}$ [3];
3: Set the last update position of $\mathcal{H}$, $s := 1$;
4: **for** $i = 1$ to N_s **do**
5: **if** $|\cup_{j=s}^{i} \mathcal{S}_j| < \gamma \cdot |\mathcal{S}|$ **then**
6: Hash $\mathcal{S}_i$ into $\mathcal{H}$ and continue;
7: **else**
8: Compute the new parameter set Θ' of the hashing structure $\mathcal{H}'$ based on $\mathcal{S} \cup (\cup_{j=s}^{i} \mathcal{S}_j)$;
9: **if** $\text{dif}(\Theta, \Theta') < t_\Theta$ **then**
10: Hash $\mathcal{S}_i$ into $\mathcal{H}$ and continue;
11: **else**
12: Set $\mathcal{H} := \mathcal{H}'$, re-hash $\mathcal{S} \cup (\cup_{j=s}^{i} \mathcal{S}_j)$ into $\mathcal{H}$;
13: Set $s := i$;
14: **end if**
15: **end if**
16: **end for**
17: **return** $\mathcal{H}$.

3. EXPERIMENTAL CONFIGURATION

A music search system on top of YouTube APIs and Marsyas [18] was implemented as an exemplar application of the proposed framework. In this section, we give an introduction on experimental configuration for empirical study. Sec. 3.1 describes query design and two music test collections. Sec. 3.2 details the methodology for the experimental study, hardware configuration and evaluation metric.

3.1 Design of Database and Query

By crawling the audio stream of music videos on YouTube, we built a tunable test collection (TS1 with 3020 music items) with YouTube social text information and manually labeled content related tags (Table 2 shows its hierarchy). TS1 is labeled and cross checked by multiple amateur musicians to ensure the validity of the ground truth. TS1 is intended to evaluate the effectiveness of FMSV generation and compare the retrieval precision of FMSV with other audio signatures. A large scale test collection (TS2 with 100,000 music items) with YouTube social text information and the built FMSV description[2] was built to evaluate the effectiveness of FMSV on large scale collections and the scalability of the proposed framework.

To simulate the realistic music search behavior, we design music queries with different levels of *complexity in musical information need*. Audio queries were designed to allow personalization of any single music dimension or any combination of music dimensions. Some examples of the designed

[2]The YouTube ID lists, human labeled tags and audio features of both test collections can be obtained by emailing the first author.

Table 3: Examples of designed queries to evaluate the example system for personalized music search.

No.	Genre	Mood	Vocalness	Instrument	Comments
1	Country	Passionate	Male		"Thank God I'm a Country Boy" by John Denver
2	Country	Humorous	Female		"Landslide" by Dixie Chicks
3	HipHop	Aggressive	Male		"Till I Collapse" by Eminem
4	Classical		Nonvocal	Violin	"Partita No. 3" in E by Bach
5	Baroque	Rollicking	Nonvocal	Piano	"First Impressions" by William Goldstein
6	Romantic	Rollicking	Nonvocal	Guitar	"Another Day" by Dream Theater
7	Metal	Aggressive			"The Metal Lyrics" by Tenacious D
...	.....	.....	.....	.....	.....
23	Baroque	Rollicking	Nonvocal	Piano	"Aria" by Daniel Barenboim
24			Nonvocal	Snare	"Krystal Klear" Snare Drum Solo by Scott Fairdosi
25		Passionate	Nonvocal	DrumKit	"Travis Barker Superbowl Drum Remix" by Haven Lamoureux

queries are listed in Table 3. Each query is associated with different music dimensions, which simulates the search situation that different users may want to search similar music to the query based on its different music aspects, i.e., genre, mood, etc. Users can form *low complex queries* (personalize one music dimension) or *high complex queries* (personalize more dimensions) to search for their wanted music.

3.2 Methodology

24 subjects volunteered for the evaluation. 10 of them are amateur musicians, familiar with various music styles and taxonomy. The other 14 are music hobbyists. It is noted that for each audio query, the class labels of each music dimension only serve as a reference. The subjects do not need to know the actual meaning of all the class labels in order to judge the similarity of the returned results. They just need to distinguish different music dimensions.

For each test collection, the same methodology was applied to conduct experiments. A briefing was conducted before the experiment to make sure subjects understood the experimental procedure and were familiar with the music dimensions to be used. Firstly, subjects were asked to do searches with low complex queries by randomly selecting an audio query of one personalized music dimension. For each search task, subjects needed to judge whether each of the first 30 returned results was similar to the query in the personalized music dimension. For a complete trial, each subject repeated this with each of the music dimensions personalized and an audio query randomly selected. With this procedure, we guaranteed that over each music dimension, the same number of searches were performed and the selected queries for each dimension were uniformly distributed among all the designed queries. Secondly, high complex queries were used for searches by subjects. We followed the methodology described above to ask each subject to conduct at least one complete trail over all combinations of the music dimensions. When more music dimensions were personalized, the returned result is considered relevant as long as it was similar to the query in any of the music dimensions.

Precision@n is used as the metric to evaluate the retrieval effectiveness. It is defined as the percentage of the relevant results in the top n returned ones. The average precision@{5-30} was measured for search tasks of both low and high complex queries. The *average runing/response time* were employed to evaluate the system efficiency. All experiments were conducted on a DELL PowerEdge 2970 workstation with 2 CPUs (each is a Quad-Core Intel Xeon E5420, 2x6MB cache CPU) and 32GB memory (DDR-2 667MHz).

4. RESULT ANALYSIS

In this section, we study the proposed framework from two main aspects - effectiveness and efficiency.

4.1 Effectiveness Study

4.1.1 Effectiveness of FMSV generation

Effective FMSV generation plays a very important role on the final performance of the whole system. For music dimensions, such as genre, mood, instrument, and vocalness, multi-class SVMs were trained using randomly selected 50% of music items in each class and evaluated using the rest on TS1. 10 evaluation trials were conducted. The average classification accuracy and standard deviation are listed in Table 4. These accuracies of our approach are comparable to the state of the art performances [8]. The high quality FMSV generation is the foundation of accurate music retrieval.

Table 4: Effectiveness of generating FMSV.

Genre	Mood	Vocal	Instrument
61.0±1.4	70.7±0.6	71.6 ± 2.3	75.9 ± 3.4

4.1.2 Effectiveness of search

Based on TS1, we compare the retrieval effectiveness of FMSV with other audio signatures: existing audio features (AF), described in Sec. 2.3.1, and the transformed audio features by principal component analysis (AFPCA). For AF, all the 115 features components were combined as a music signature for genre/mood dimensions, and 55 feature components (without rhythmic and melody features) were combined for instrument/vocalness. For AFPCA, 95% data variance was retained during PCA, which corresponds to 18 and 12 feature components for genre/mood and instrument/vocalness, respectively. 50% of data were used to train FMSV and AFPCA, the rest were used for testing.

Fig. 4 shows the precision@{5-30} of searches using FMSV, AF and AFPCA for low complex queries. In each of the four music dimensions, FMSV clearly outperforms AF and AFPCA with statistically significant improvement. Fig. 5 illustrates their retrieval precision for high complex queries. It is noted that when personalizing more music dimensions, search precision consistently gets better than personalizing one music dimension. With high complex queries, FMSV still performs the best. In some queries with genre+instrument or genre+vocalness personalized, FMSV reveals more improvement than with low complex queries. Those results imply that music content representation based on FMSV carries more useful information and enjoy superior discrimination capability. It leads to better search accuracy.

Fig. 6 illustrates the average precision of FMSV for low/high complex queries on TS2. One thing worth noting is that while the size of test collection becomes larger, FMSV still can sustain superior retrieval accuracy. This result demonstrates the robustness of FMSV from another perspective.

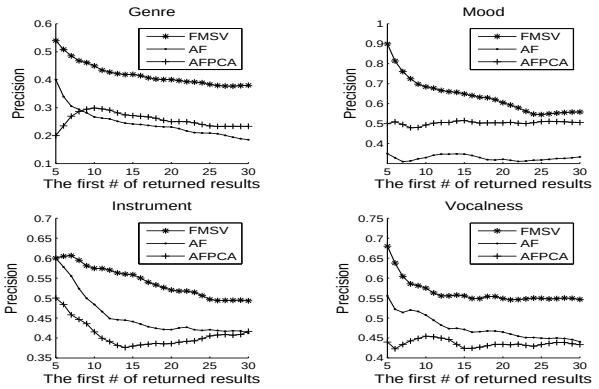

Figure 4: Average precision@{5-30} comparison for low complex queries on TS1.

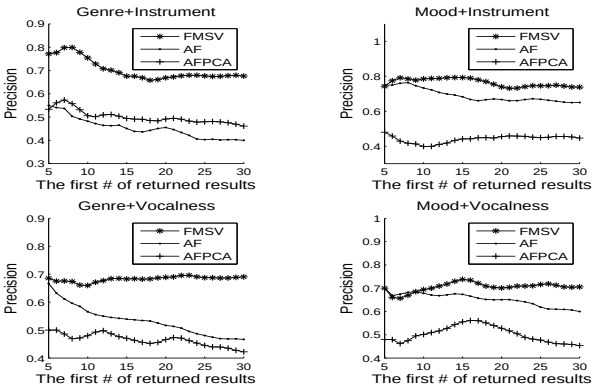

Figure 5: Average precision@{5-30} comparison for high complex queries on TS1.

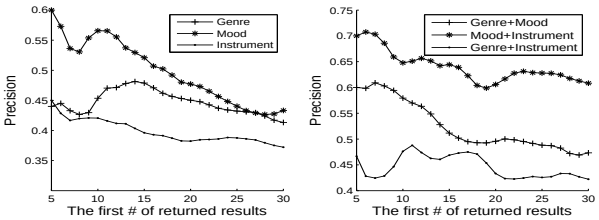

Figure 6: Average precision@{5-30} of FMSV for both low and high complex queries on TS2.

4.2 Efficiency Study

4.2.1 *Efficiency of FMSV generation*

Training SVMs could be a very time consuming process. In the first set of experiment, we evaluate ePEGASOS over a large data set, the Reuters CCAT[3]. The main purpose of this study is to show that using the proposed algorithm, SVM training time has inverse dependency on the size of training data, provided that the same generalization error is maintained. The left sub-figure of Fig. 7 shows the average running time of ePEGASOS training a multi-class SVM on CCAT. It is noted that the running time decreases when more and more training data are provided. On large data

[3]CCAT consists of 804,414 samples with 47,236 components.

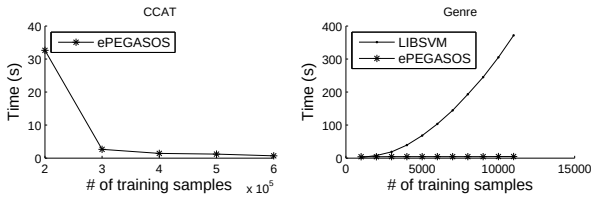

Figure 7: The average running time of SMO and ePEGASOS in training multi-class SVMs with probability estimate on different sized datasets.

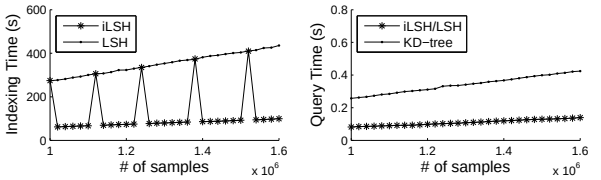

Figure 8: The indexing and query time comparison in incremental indexing scenario.

sets, this is desirable to train SVMs with less running time and better generalization performance.

We further compare the average running time of ePEGASOS and SMO[4] on a smaller scale genre feature set to show its efficiency. As shown in the right sub-figure of Fig. 7, the running time of ePEGASOS almost stays the same as more training data are added, while the running time of SMO increases dramatically. Due to the much smaller scale of the genre feature set compared with CCAT, the running time of ePEGASOS is already very low and does not decrease as dramatically as on CCAT.

In the feature selection algorithm (Alg. 1), the stopping threshold t_e was set as 0.03. With this setting, 30 out of 115 features were selected for genre/mood dimensions and 20 out of 55 features were selected for instrument/vocalness. The average FMSV generation time for a 3-minute music item is reduced from 1.561 to 1.303 seconds and from 1.334 to 1.127 seconds, respectively. The 0.2 seconds improvement is significant as it constitutes more than 10% of the total response time ($\approx$ 1.7 seconds), described in Sec. 4.2.3.

4.2.2 *Efficiency of index construction and query processing*

For large MIR systems, economic maintenance cost is another important concern. In this study, we compare the average index construction time of iLSH and LSH in the following scenario: firstly index a static data set, which contains 1,000,000 data samples of 15 components with value ranging from 0 to 1 to simulate FMSVs; then update the index structure when 20,000 new samples are added into the data set at regular time instances. The size of the initial static data set is at the comparable order of commercial music databases, such as YouTube and Last.fm. The number of samples added at each time instance simulates the music items uploaded by users on YouTube or created by new artists on Last.fm in a period of time. This scenario considers the need of incremental indexing in real life applications.

Fig. 8 shows the average index construction time of 10 runs with $\gamma = 0.1$ and $t_\Theta = 0.08$. iLSH performs signif-

[4]SMO is used in LIBSVM, an efficient SVM implementation package.

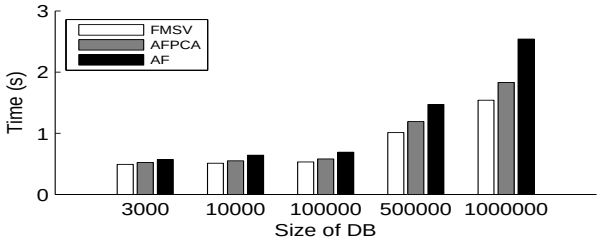

Figure 9: The average response time of search in single music dimension on various data set scales.

icantly better than LSH at most of the time instances, as iLSH only updates the index structure instead of re-indexing from scratch like LSH. At the time instances when iLSH performs a complete update (re-indexing), its running time is the same as LSH.

The average top-100 query time of iLSH/LSH was compared with KD-tree [4]. The average query time of 100 queries is illustrated in Fig. 8. It is noted that iLSH and LSH has the same query time ($\approx$ 100 ms in a data set of 1.6 million samples), as they follow the same procedure to search nearest neighbors. Their query time is significantly lower than KD-tree over all sized data sets.

4.2.3 *Efficiency of search*

In Fig. 9, we compared the average top-100 response time of a search process including query upload, music signature generation, query, and ranking for a single music dimension. Different music signatures (FMSV, AF, and AFPCA) on various sized data sets were evaluated. Since FMSV has many fewer components ($\approx$ 10) than AF (115), the response time using FMSV is significantly less than using AF, especially on large data set. After applying PCA on AF (AFPCA), the response time is reduced compared with AF. However, due to the concern of retrieval effectiveness, enough features must be retained in PCA (could be > 10). This adds unpredictable factors to the response time, as different feature sets need to keep different number of components in PCA. In our system, as AFPCA has more features than FMSV, its response time is longer. As the data set gets larger, the response time of FMSV remains acceptable (≈ 0.5 seconds on the data set with 3000 samples and ≈ 1.7 seconds on the data set with 1 million samples). The flexible indexing approach of the framework allows easy parallel implementation of music search over multiple music dimensions. Therefore, the above response time is illustrative even when searching is with multiple music dimensions.

With fast response time in each music dimension and efficient parallel computation for multiple dimensions, the proposed framework scales well on large databases.

5. CONCLUSIONS

We have presented CompositeMap, a novel framework of multimodal music similarity measure to facilitate various music retrieval tasks such as organizing, browsing, and searching in a large data set. We have detailed the FMSV which can map any existing audio features into high-level concepts such as genre, mood, etc. CompositeMap has unified content-based, metadata-based, and semantic description-based music retrieval approaches. It combines different music facets into a compact signature which can enable personalized services for users with different information needs, background knowledge, and expectations.

For a case study, we have employed CompositeMap in a music search engine to evaluate its effectiveness, efficiency, adaptiveness and scalability using two separate large scale music collections extracted from YouTube. Our objective evaluation and user study show the clear advantages of the proposed framework. Furthermore, our project has led to several innovations including an efficient SVM training algorithm with multi-class probability estimates and an incremental Locality Sensitive Hashing algorithm.

6. REFERENCES

[1] `http://www.last.fm`.
[2] `http://www.YouTube.com`.
[3] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS'06*, 2006.
[4] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 1975.
[5] A. Berenzweig, B. Logan, D. Eills, and B. Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Comput. Music J.*, 2004.
[6] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: current directions and future challenges. *Proc. of the IEEE*, 2008.
[7] S. J. Downie. The scientific evaluation of music information retrieval systems: Foundations and future. *Computer Music Journal*, 2004.
[8] S. J. Downie. The music information retrieval evaluation exchange (2005 - 007): A window into music information retrieval research. *Acoustical Science and Technology*, 2008.
[9] T. Huang, R. Weng, and C. Lin. Generalized bradley-terry models and multi-class probability estimates. *J. Mach. Learn. Res.*, 2006.
[10] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. of the ISMIR*, 2000.
[11] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *Proc. of IEEE ICME*, 2001.
[12] L. Lu, D. Liu, and H. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Trans. Acoust., Speech, Signal*, 2006.
[13] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
[14] W. Ng, D. Yeung, M. Firth, E. Tsang, and X. Wang. Feature selection using localized generalization error for supervised classification problems using RBFNN. *Pattern Recognition*, 2008.
[15] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 2000.
[16] S. Shwartz and N. Srebro. SVM optimization: inverse dependence on training set size. In *ICML'08*, 2008.
[17] 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.
[18] G. Tzanetakis and P. Cook. marsyas a framework for audio analysis. *Organized Sound*, 2000.
[19] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Speech and Audio Proc.*, 2002.
[20] J. You, S. Park, and I. Kim. An efficient frequent melody indexing method to improve the performance of query-by-humming systems. *Journal of Information Science*, 2008.