

Music Clustering With Features From Different Information Sources

Tao Li, Mitsunori Ogihara, Wei Peng, Bo Shao, and Shenghuo Zhu

Abstract—Efficient and intelligent music information retrieval is a very important topic of the 21st century. With the ultimate goal of building personal music information retrieval systems, this paper studies the problem of identifying “similar” artists using features from diverse information sources. In this paper, we first present a clustering algorithm that integrates features from both sources to perform bimodal learning. We then present an approach based on the generalized constraint clustering algorithm by incorporating the instance-level constraints. The algorithms are tested on a data set consisting of 570 songs from 53 albums of 41 artists using artist similarity provided by All Music Guide. Experimental results show that the accuracy of artist similarity identification can be significantly improved.

Index Terms—Clustering, different information sources, machine learning, music information retrieval.

I. INTRODUCTION

A. Music Clustering

FOR those who listen to music through the Internet, how to navigate in the ocean of online music is an important issue. Nowadays, everything about music is on the web—audio, lyrics, artist discographies, artist biographies, reviews, and discussions. This raises an issue of whether the online music data can be efficiently accessed so that the user can benefit from the existence of such large volumes of data. A solution to the issue can be given by developing efficient music information retrieval programs, which integrate techniques for analyzing, summarizing, indexing, classifying, and grouping music data.

Two fundamental problems in dealing with music data are classification and clustering. While classification aims at assigning predefined class labels to the data, clustering aims at dividing the data into classes based on their similarity without predefined class labels. Since it requires user input for training, the former problem is called *supervised learning*. In contrast the latter problem does not require user input, and thus is called *unsupervised learning*. While there is a vast literature on music classification, the problem of music clustering is much less explored, but interesting pieces of work do exist [10], [27], [32].

Manuscript received November 29, 2007; revised September 30, 2008. First published March 04, 2009; current version published March 18, 2009. The work of T. Li was supported in part by NSF Career Award IIS-054680. This work was supported in part by the Open Research Fund of the Lab of Spatial Data Mining and Information Sharing of Ministry of Education of China at Fuzhou University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Horace Ho-Shing Ip.

T. Li, W. Peng, and B. Shao are with the School of Computer Science, Florida International University, Miami, FL 33199 USA.

M. Ogihara is with the Department of Computer Science, University of Miami, Coral Gables, FL 33146 USA.

S. Zhu is with NEC Laboratories America, Cupertino, CA 95014 USA.

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

Digital Object Identifier 10.1109/TMM.2009.2012942

B. Semi-Supervised Learning From Different Information Sources

In music information retrieval, the data are naturally multimodal, in the sense that they are represented by multiple sets of features. For example, the representation of a song has three modes: 1) the personnel (the producer, the director, the editor, the scenario writer, the music composer, the cast, etc.), 2) the lyric features, and 3) the acoustic features (which summarize the voice and the background audio). Since the proportion of predefined class labels available as part of input is 0% for clustering and 100% for classification, one naturally wonders about the special cases of these two fundamental problems in which only a part of the data has predefined labels.

The problem, whether it may be classification or clustering, in such a situation is called *semi-supervised learning*. The main question in semi-supervised learning is whether it is possible to use the unlabeled data to produce something better than the one produced using only the labeled data. In particular, for semi-supervised learning of multimodal data, i.e., data with heterogeneous sets of features, a natural question is whether multimodality can be effectively utilized in learning and, if so, whether such multimodal learning methods produce better results than unimodal methods.

The celebrated paper of Blum and Mitchell [8] is the first to address formally this question. In this paper, Blum and Mitchell study the problem of incorporating unlabeled data in building classifiers in the presence of two feature sets. In particular, they propose a strategy for constructing classifiers called *co-training* for the purpose of making use of unlabeled data. The co-training algorithm proceeds in rounds in the following way: In each round a classifier is built on each of the two feature sets using the current training set, which is initially set to the set of data whose labels are given as input. Then, for each feature set, the point among the unlabeled data for which the classifier with respect to the feature set provides the most confident assertion is selected and is added to the training set of the other feature set along with the assertion. (Note that the two classifiers may select an identical point and disagree on its class label). Blum and Mitchell show that under a certain “independence” assumption about the joint distribution of the feature sets their co-training algorithm converges in the sense of PAC-learning. Many research efforts have been done for the purpose of extending and generalizing the idea of co-training [2], [11], [18], [25], [29].

It is also possible to design an interactive (or ensemble)¹ learning algorithm (that exploits interactions among classifiers

¹In the literature, the word “interactive” is used often in the case of bimodal learning and “ensemble” in the case of learning with more than two modes. Here we use “interactive” throughout, even to mean learning of data with more than two modes.

to improve accuracy) for supervised learning (that is, all the data are already labeled). For example, the *co-boosting* algorithm of Collins and Singer [9] uses the individual boosting of the feature sets with the weight adjustments influenced by the labeling of the other classifier(s). The approach can be used not only for supervised learning but for semi-supervised learning (indeed co-boosting algorithm was originally conceived for semi-supervised learning). Although such algorithms may fall into pitfalls due to the highly simple mutual boosting structure, Collins and Singer point out, such multimodal learning can be very powerful and thus is worth while.

The work of Blum and Mitchell and that of Collins and Singer study the design of effective algorithms multimodality through interaction for semi-supervised learning and for supervised learning, respectively. This naturally leads to the question of whether multimodal interactive methods can be more powerful than unimodal methods in the case of unsupervised learning, namely, clustering.

C. Contributions of the Paper

This paper addresses the issue of clustering pop music into groups with respect to the artists from diverse information sources and we develop algorithms to improving the clustering by integrating different information sources. We first develop a new bimodal music clustering algorithm for integrating the features based on minimizing disagreement. To apply the bimodal music clustering, we need to have a complete feature representation, i.e., we need to know the content and lyrics information for each song. However, situations exist we might not be able to have the complete feature representation. For example, sometimes the lyrics may not be available for all the songs in our study. In addition, in many cases, some data sources may not be as informative as other data sources. For example, the lyrics may not be able to provide the same level of details of genre/style information as the acoustic features. This motivates us to study music clustering with constraints: One data source is chosen as primary information source. The other data sources are treated as secondary information and are used as constraints to improve the clustering results based on the primary source. In summary, we study the following two related problems in the paper.

- **Bimodal music clustering:** Note that in music information retrieval, the personnel feature set of the representation of music, is significantly smaller than that of movies, since many music artists produce, compose, and perform themselves. This compels one to take the standpoint that the representation of popular music is bimodal, consisting of the acoustic features, which summarize the sound, and the text features, which summarize the words put into the music. To apply the bimodal music clustering, we need to have a complete feature representation, i.e., we need to know the content and lyrics information for each song. We of course anticipate that bimodal clustering techniques can be naturally extended to general multimodal clustering.
- **Music clustering with constraints:** In practice, bimodal clustering might not be plausible for the following two scenarios: 1) The feature set from some information source might not be sufficient enough to represent the music (e.g.,

the personnel features described above); 2) We may not always have the complete feature representation. For instance, sometimes we only have the lyrics information or meta-data information of a small number of songs. To utilize these partial or incomplete information from diverse information sources, we represent it as instance-level constraints (e.g., two artists share similar lyrics or personnel features) and study the problem of music clustering with constraints [26].

In this paper, we first present a bimodal clustering framework for integrating the features based on minimizing disagreement. It is known that in bimodal learning minimizing disagreement between two classifiers can improve the performance of learning [5], [14]. In our framework, minimizing disagreement can be considered as a simple common theme of multimodal information retrieval: individual feature sets interact to help each other by reducing disagreement among their outputs. We present a bimodal clustering algorithm based on the common theme—initialize the cluster layout using the output of its counterpart and try to minimize the disagreement between these two modes.

We then investigate the problem of content-based music clustering with instance-level constraints generated from diverse information sources. The use of instance-level constraints as the background information to improve data clustering has been widely studied in machine learning in the past few years. Instance-level constraints are generally pairwise and they are of two types: the *positive constraint* is one that specifies that two instances must belong to the same cluster, and the *negative constraint* is one that specifies that two instances must belong to different clusters. These instance-level constraints have been used in learning distance/dissimilarity measures [4], [6], [13], [35], modifying objective criteria for cluster evaluation [1], and improving optimization procedures [3], [33]. In particular, we adapt a generalized constraint clustering algorithm based on K-means and discuss approaches of automatically generating constraints.

Finally, we evaluate our algorithms on a data set consisting of 53 albums covering 41 popular artists. The “correctness” of the clusters generated is tested using artist similarity provided by All Music Guide. Experimental results show the effectiveness of our approaches. A preliminary version of this work has appeared in [21]. The rest of the paper is organized as follows: Section II presents the bimodal clustering algorithm. In particular, Section II-A introduces the underlying principle of minimizing the disagreement and Section II-B describes the clustering procedure of utilizing the general principle. Section III presents the algorithm for constraint-based clustering. Section IV describes feature extraction approaches and constraints generation procedure from different data sources. Section V reports the results of experiments. Finally Section VI concludes.

II. BIMODAL CLUSTERING

A. Minimizing the Disagreement

1) *Theoretical Underpinnings:* In this section, we introduce the basic principle of minimizing disagreement, i.e., minimizing the disagreement between two individual models could lead to the improvement of learning performance of individual models.

Our data are bimodal: let X_1 and X_2 be the space of the first mode and the space of the second mode, respectively. Let $X = (X_1, X_2)$ be the product space of X_1 and X_2 . Let 0 and 1 be the class labels of these data, which we will often denote by Y . For each $u \in \{0, 1\}$, we use \bar{u} to denote its opposite class label, that is, $1 - u$. Suppose that the data in X is subject to a distribution D . Let f be our class label function and let f_1 and f_2 be our class label functions based on the first mode and on the second mode, respectively. The (x) in f and Y are often omitted—we will write $f = u$ to mean $f(x) = u$ and $Y = u$ to mean $Y(x) = u$, etc.

Definition 1: We say that f is a good classifier if for all $u \in \{0, 1\}$

$$\Pr(f(x) = u|Y(x) = u) > \Pr(f(x) = \bar{u}|Y(x) = u)$$

where the probability is subject to D .

In [8], it is assumed that x_1 and x_2 are conditionally independent given the labels, i.e.,

$$\Pr(x_1 = x'_1|x_2 = x'_2) = \Pr(x_1 = x'_1|f_2(x_2) = f_2(x'_2)).$$

The independence assumption is rather strong, but has been used by many successful applications. Suppose we build hypotheses f'_1 on X_1 and f'_2 on X_2 . Thus, if x_1 and x_2 are conditional independent given the labels, then f'_1 and f'_2 are also conditional independent. The conditional independence of f'_1 and f'_2 can be interpreted as follows:

$$\Pr(f'_1(x_1) = u|f'_2(x_2) = v, Y = y) = \Pr(f'_1(x_1) = u|Y = y)$$

where $u, v, y \in \{0, 1\}$. In other words, *The conditional independence* implies that 1) for all $S_1 \subseteq X_1$ such that the probability of (S_1, X_2) is nonzero, the distribution of X_2 in which the first mode is restricted to S_1 is identical to the distribution of X_2 with no restriction; and that 2) for all $S_2 \subseteq X_2$ such that the probability of (X_1, S_2) is nonzero, the distribution of X_1 in which the first mode is restricted to S_2 is identical to the distribution of X_1 with no restriction.

One can show the following [19].

Theorem 1: Under conditional independence assumption, the disagreement upper bounds the misclassification error for the good classifiers.

In essence, this indicates that, under certain conditions, the disagreement upper bounds the misclassification error. Thus, minimizing disagreement will ideally decrease the upper bound on the misclassification error and could bootstrap the learning algorithm. It should be pointed out that although the principle was originally proved in the context of supervised learning [14], it can be regarded as a simple common theme of multimodal information retrieval: individual feature sets interact to help each other by reducing disagreement among their outputs.

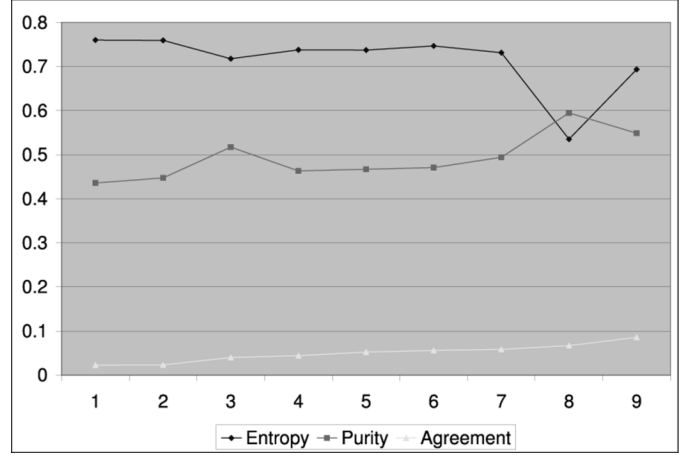


Fig. 1. Relationships between clustering performance and agreements. Each unit on the X-axis represents five iterations of the algorithm and the Y-axis shows the performance value.

B. Bimodal Clustering Algorithm

In this section, we present a clustering algorithm that integrates different features based on the principle of minimizing disagreements.

1) *Measuring Agreements Between Clusterings:* Let $D = \{d_1, d_2, \dots, d_n\}$ be a set of n data points. Suppose we are given two clusterings P_1 and P_2 with each consists of a set of clusters

$$P_i = \{C_i^1, C_i^2, \dots, C_i^{k_i}\}, \quad i = 1, 2$$

where k_i is the number of clusters for clustering P_i , and $D = \bigcup_{j=1}^{k_i} C_i^j$. The first question is how to measure the agreements between two clusterings.

We use Adjusted Rand Index to compute the agreement between clusterings. Adjusted Rand Index is a statistic to assess the clustering quality compared against assigned known classes. The Rand Index is defined as the number of pairs of objects which are both located in the same cluster and the same class, or both in different clusters and different classes, divided by the total number of objects [28]. Adjusted Rand Index which adjusts Rand Index is set between $[0, 1]$ [22]. The higher the Adjusted Rand Index, the more resemblance between the two clusterings. Formally, the adjusted Rand index, *ARI*, is defined as

$$\frac{\sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \binom{n_{ij}}{2} - \sum_{i=1}^{k_1} \binom{n_{i\cdot}}{2} \sum_{j=1}^{k_2} \binom{\binom{n_{\cdot j}}{2}}{\binom{n}{2}}}{\left(\sum_{i=1}^{k_1} \binom{n_{i\cdot}}{2} + \sum_{j=1}^{k_2} \binom{n_{\cdot j}}{2} \right)}}, \frac{2 - \sum_{i=1}^{k_1} \binom{n_{i\cdot}}{2} \sum_{j=1}^{k_2} \binom{\binom{n_{\cdot j}}{2}}{\binom{n}{2}}}{\binom{n}{2}}$$

Here n_{ij} denotes the number of objects belonging to both C_i^1 and C_j^2 , $n_{i\cdot} = \sum_{j=1}^{k_2} n_{ij}$, and $n_{\cdot j} = \sum_{i=1}^{k_1} n_{ij}$.

2) *Clustering Procedure:* Based on the principle of minimizing the disagreement, we now present a bimodal clustering approach. This algorithm is an extension of the EM method [15]. In each iteration of algorithm, an EM type procedure is

TABLE I
LIST OF NOTATIONS

n	Number of Songs
$s_i = (s_i^1, s_i^2)$	s_i has two modes: content s_i^1 and lyrics s_i^2
$S = (s_1, \dots, s_n)$	A collection of songs
K	Number of clusters
$\Lambda^1 = (\lambda_1^1, \dots, \lambda_K^1)$	Modal 1 model parameters
$\Lambda^2 = (\lambda_1^2, \dots, \lambda_K^2)$	Modal 2 model parameters
$Y = (y_1, \dots, y_n)$ $y_n \in \{1, \dots, K\}$	Cluster assignment vector
$s \in S$	s represents a song from S
$y_s = k$	Song s is in k -th cluster

employed to bootstrap the model by starting with the cluster assignments obtained in the previous iteration. Upon convergence, the two individual models are used to construct the final cluster assignment. Table I lists the notions used for the algorithm and Fig. 1 presents the algorithm procedure.

Algorithm 1: Bimodal Clustering

Input: S, K

Output: Cluster assignment Y and the trained model structure

1) **Initialization:** Initialize the model structure (Λ^1, Λ^2) as well as the cluster assignment Y

2) **while** the stopping criterion does not meet **do**

3) **Step I:**

Randomly pick a different data source $i \in \{1, 2\}$

4) **Step II:**

Model re-estimation for source i : for each cluster k , the model parameters, λ_k^i , are re-estimated as

$$\lambda_k^i = \operatorname{argmax}_{\lambda} \sum_{s: s \in S, y_s = k} \log P(s^i | \lambda_k^i)$$

5) **Step III:**

Sample reassignment: for each data sample $s \in S$, set

$$y_s = \operatorname{argmax}_k \log P(s^i | \lambda_k^i)$$

6) **Step IV:**

Measure the agreement between two sources. If the agreement increases, go to Step I. Otherwise, go to Step II.

7) **end while**

8) Return Y as well as the trained models (Λ^1, Λ^2)

We assume parameterized models, one for each cluster. Typically, all the models are from the same family, e.g., multivariate Gaussian. The algorithm described above is a variant of the EM algorithm. It performs an iterative optimization process for each

data source by using the cluster assignments (possibly from another data source). Note that in each iteration, one data source is picked and every data point is reassigned to one of the clusters based on information from that data source and on its previous assignment. At the end of each iteration, the algorithm explicitly checks whether the agreement between two clusterings (one clustering from each data source) has been improved. If it is improved, the algorithm then continues to iterate. Otherwise, the algorithm will go back to the allocation step and hopefully get a new clustering.

III. CONSTRAINT-BASED CLUSTERING

This section provides some background on the K-means algorithm and then discusses the constraint-based clustering algorithm following the exposition in [13].

A. K-Means Clustering

K-means is a popular clustering algorithm where the input data set is partitioned into K groups, where the number K is specified by the user. The quality of partition into K clusters can be viewed as the *quantization error* described in the following:

$$E = \frac{1}{2} \sum_{j=1}^K \sum_{s \in C_j} (\bar{c}_j - s)^2. \quad (1)$$

Here C_1, \dots, C_K are the K clusters and $\bar{c}_1, \dots, \bar{c}_K$ their centroids. The goal of K-means is to minimize this quantization error, which is accomplished by iteratively alternating between the allocation step and the evaluation step. In the former each data point is allocated to the cluster whose centroid is the closest to it so as to minimize the quantization error with respect to the current centroids, while in the latter, the centroid of each cluster is updated based on the new allocation.

B. Constraint-Based Clustering

Following [13] we define the concept of constraint-based clustering for music similarity. We modify the objective function so that penalty is added for each constraint that is not satisfied. For a positive constraint (s_i, s_j) the penalty (in the case where they go to different clusters) is the squared distance between their cluster centroids. For a negative constraint (s_i, s_j) the penalty (in the case where they go to the same clusters) is the squared distance between the centroids that are the closest and the second closest to either s_i or s_j . In both cases, we use the centroids to determine the penalty so as to treat constraint violations equally within a cluster, and we use squared distance since the quantization error is based on squared distance.

The formula for the objective function is given in the following:

$$CE = \frac{1}{2} (E + PM + PC) \quad (2)$$

$$= \frac{1}{2} \left(\sum_{j=1}^K \sum_{s \in C_j} (\bar{c}_j - s)^2 + PM + PC \right) \quad (3)$$

$$PM = \sum_{(s_i, s_j) \in M} p_{ij}^m (1 - \Delta(y(s_i), y(s_j))) \quad (4)$$

$$PC = \sum_{(s_i, s_j) \in C} p_{ij}^c \Delta(y(s_i), y(s_j)) \quad (5)$$

$$p_{ij}^m = (\bar{c}_{y(x_i)} - \bar{c}_{y(x_j)})^2 \quad (6)$$

$$p_{ij}^c = (\bar{c}_{y(x_i)} - \bar{c}_{ij}^*)^2. \quad (7)$$

Here M and C respectively represent the set of positive constraints and the set of negative constraints, p_{ij}^m and p_{ij}^c are respectively penalty parameters for the positive and negative constraints, and the value of $y(s_i)$ is the index of the cluster to which the data point s_i belongs. Also, Δ is the Kronecker delta function defined by: $\Delta(x, y) = 1$ if $x = y$ and 0 otherwise. That is, the penalty p_{ij}^m is added only if $(s_i, s_j) \in M$ but s_i and s_j belong to different clusters; and the penalty p_{ij}^c is added only if $(s_i, s_j) \in C$ but s_i and s_j belong to the same cluster. Furthermore, \bar{c}_{ij}^* is the centroid that is the next closest to either s_i or s_j .

Like K-means, the constraint-based clustering algorithm is iterative, alternating between the allocation step and the centroid update step. In the allocation step, the goal is to minimize the generalized constrained vector quantization error in (3). This is achieved by assigning instances so as to minimize the proposed error term. For pairs of instances in the constraint set, the quantization error CE is calculated for each possible combination of cluster assignments, and the instances are assigned to the clusters so that CE is minimized. In the update step, the centroids are cluster centroids. As in K-means, the first order partial derivatives of CE with respect to each centroid is evaluated and the solution that makes all these derivatives equal to zero is obtained.

IV. FEATURE EXTRACTION AND CONSTRAINT GENERATION

In this section, we describe the feature sets extracted from the lyrics and the acoustic content and we also discuss various approaches to generate constraints in music applications.

A. Feature Extraction

1) *Content-Based Features*: In our study, we use timbral features along with wavelet coefficient histograms. The feature set consists of the following three parts and totals 35 features.

a) *Mel-Frequency Cepstral Coefficients (MFCC)*: MFCC is a feature set popular in speech processing and is designed to capture short-term spectral-based features. To obtain the feature, we first compute, for each frame, the logarithm of the amplitude spectrum based on short-term Fourier transform, where the frequencies are divided into thirteen bins using the Mel-frequency scaling. After taking the logarithm of the amplitude spectrum, the frequency bins are grouped and smoothed according to Mel-frequency scaling, which is design to agree with perception. MFCC features are generated by decorrelating the Mel-spectral vectors using discrete cosine transform. In this study, we use the first five bins, and compute the mean and variance of each over the frames.

b) *Short-Term Fourier Transform Features (FFT)*: This is a set of features related to timbral textures and is not captured using MFCC. It consists of the following five types: Spectral

Centroid, Spectral Rolloff, Spectral Flux, Zero Crossings, and Low Energy. More detailed descriptions of STFT can be found in [31].

c) *Daubechies Wavelet Coefficient Histograms (DWCH)*: The Daubechies Wavelet Coefficient Histograms, proposed in [20], are features extracted in the following manner: First, the Daubechies-8 (db8) filter with seven levels of decomposition (or seven subbands) is applied to 30 s of monaural audio signals. Then, the histogram of the wavelet coefficients is computed at each subband. Then the first three moments of a histogram, i.e., the average, the variance, and the skewness, are calculated from each subband. In addition, the subband energy, defined as the mean of the absolute value of the coefficients, is computed from each subband. More details of DWCH can be found in [20].

2) *Text-Based Style Features*: To account for the characteristics of the lyrics, our text-based feature extraction consists of four components: bag-of-words features, part-of-speech statistics, lexical features, and orthographic features.

- *Bag-of-words*: We compute the TF-IDF measure for each words and select the top 200 words as our features. We do not apply stemming operations.
- *Part-of-speech statistics*: We also use the output of Brill's part-of-speech (POS) tagger [7] as the basis for feature extraction. POS statistics usually reflect the characteristics of writing. There are 36 POS features extracted for each document, one for each POS tag expressed as a percentage of the total number of words for the document.
- *Lexical Features*: By lexical features, we mean features of individual word-tokens in the text. The most basic lexical features are lists of 303 generic function words taken from [23], which generally serve as proxies for choice in syntactic (e.g., preposition phrase modifiers versus adjectives or adverbs), semantic (e.g., usage of passive voice indicated by auxiliary verbs), and pragmatic (e.g., first-person pronouns indicating personalization of a text) planes. Function words have been shown to be effective style markers.
- *Orthographic features*: We also use orthographic features of lexical items, such as capitalization, word placement, and word length distribution as our features. Word orders and lengths are very useful since the writing of lyrics usually follows certain melody.

B. Constraints Generation

The constraints come naturally in the context of music applications. Constraints can be generated from the background knowledge. If we already know that two songs are of the same styles, or formally, if we know two songs have the same cluster labels, then they must be in the same cluster (e.g., a positive constraint). Similarly, if it is known that two songs are of different styles, then they should be in different clusters (e.g., a negative constraint).

In our study, constraints can be generated from complementary and diverse music information sources. For example, if two piece of music have the same personnel-related features or lyrics, then they can be considered to be similar based on content.

TABLE II
CLUSTER MEMBERSHIPS

Clusters	Members
No. 1	{ <i>Fleetwood Mac, Yes, Utopia, Elton John, Genesis, Steely Dan, Peter Gabriel</i> }
No. 2	{ <i>Carly Simon, Joni Mitchell, James Taylor, Suzanne Vega, Ricky Lee Jones, Simon & Garfunkel</i> }
No. 3	{ <i>AC/DC, Black Sabbath, ZZ Top, Led Zeppelin, Grand Funk Railroad, Derek & The Dominos</i> }
No. 4	All the remaining artists

V. EXPERIMENTS

A. Data Description

Our experiments are performed on the dataset consisting of 570 songs from 53 albums of a total of 41 artists. The sound recordings and the lyrics from them are obtained. To obtain the ground truth of song styles, we choose to use similarity information between artists available at All Music Guide artist pages (<http://www.allmusic.com>), assuming that this information is the reflection of multiple individual users. By examining All Music Guide artist pages, if the name of an artist X appears on the list of artists similar to Y, it is considered that X is similar to Y. We select artists having a large number of neighbors. There are three of them, Fleetwood Mac, Yes, and Utopia. These three are neighbors to one another, so we select the neighbors of these three as a cluster. Of the remaining nodes we identify two other clusters in a similar manner. All the remaining artists are in a separated cluster. The clusters are listed in Table II. Our goal is to identify the song styles using both content and lyrics, i.e., cluster the 570 songs into the four different clusters. We use the cluster information of the artists as the labels for their songs.

B. Evaluation Measures

As discussed above, we use the cluster structures obtained from All Music Guide as labels to evaluate the clustering performance. We use Purity, Entropy, and Accuracy [16], [36] as our performance measures. We expect these measures would provide us with good insights on how our algorithm works.

Purity measures the extent to which each cluster contains data points from primarily one class [36]. In general, the larger the values of purity, the better the clustering solution is. Entropy measures how classes distributed on various clusters. The smaller the entropy value, the better the clustering quality is.

Accuracy discovers the one-to-one relationship between clusters and classes, therefore measure the extent to which each cluster contains data points from the corresponding class [16]. It sums up the whole matching degree between all pair class-clusters. The larger accuracy usually means the better clustering performance.

C. Experimental Results on Bimodal Clustering

1) *Performance Comparison:* We compare the results of the bimodal clustering algorithm with the results obtained when the clustering is applied on the two sources of data separately.

We also compare the bimodal clustering algorithm with the following clustering strategies on integrating different informa-

TABLE III
PERFORMANCE COMPARISON. NUMBERS ARE OBTAINED BY AVERAGING OVER TEN TRIALS

Feature Set(s)	Purity	Entropy	Accuracy
Content-only	0.436	0.731	0.438
Lyrics-only	0.444	0.728	0.402
Feature-Level Integration	0.425	0.729	0.380
Cluster Integration	0.465	0.725	0.423
Sequential Integration I	0.431	0.724	0.434
Sequential Integration II	0.438	0.734	0.407
bimodal Clustering	0.471	0.697	0.453

tion sources: 1) Feature-level integration: Feature-level integration performs K-means clustering after simply concatenating the features obtained from the two data sources. 2) Cluster integration: Cluster integration refers to the procedure of obtaining a combined clustering from multiple clusterings of a dataset [17], [24], [30]. Formally, let $C_1^1, \dots, C_1^{k_1}$ denote the clusters obtained from source 1, and $C_2^1, \dots, C_2^{k_2}$ denote the clusters obtained from source 2. Each point d_i can be represented as a $(k_1 + k_2)$ -dimensional vector

$$d_i = (d_{i11}, \dots, d_{i1k_1}, \dots, d_{i21}, \dots, d_{i2k_2})$$

$$d_{ijl} = \begin{cases} 1 & d_i \in C_j^{k_j} \text{ for } 1 \leq j \leq 2. \\ 0 & \text{otherwise,} \end{cases}$$

A combined clustering can be found by applying the K-means algorithm on the new representation. 3) Sequential integration: Sequential integration is an intermediate approach of combining different information sources. It first performs clustering on one data source and obtains a clustering assignment, say, C^1, \dots, C^{k_1} . We can represent each point d_i as a k_1 -dimensional vector using the similar idea in cluster integration. Then we can combine the new representation with another data source using feature integration. Clustering can thus be performed on the new concatenated vectors. Depending on the order of the two sources, we have two sequential integration strategies: a) Sequential integration I: firstly cluster based on content, then integrate with lyrics; b) Sequential integration II: firstly cluster based on lyrics, then integrate with content.

2) *Analysis of the Results:* We compare the results of bimodal clustering with the results obtained when clustering is applied on content and lyrics separately, and with the results of other integration strategies. Table III presents the experimental results.

From the table, we observe the following.

- The performance of purity, entropy, and accuracy relative to the other is not always consistent in our comparison, i.e., higher purity values do not necessarily correspond to lower entropy values, or to higher accuracy values. This is because different evaluation measures consider different aspects of the clustering results. For example, the entropy measure takes into account the entire distribution of the data in a particular cluster and not just the largest class as in the computation of the purity. The accuracy considers the relationships among all pair class-clusters. We compare these three different measures and hope they would provide enough insights for our experiments.

- The purity and accuracy of feature-level integration are worse than those of content-only and lyric-only clustering methods, while their entropy values are fairly close. This shows that even though the joint feature space is often more informative than that available from individual sources, naive feature integration tends to generalize poorly [34].
- Cluster integration: The cluster integration performs better than content-only and lyrics-only: cluster integration has higher purity and accuracy values and lower entropy values than those of content-only and lyrics-only. This actually conforms to the results in [17]: cluster aggregation would usually provide better clustering results.
- Sequential integration: the results of sequential integration are generally better than feature-level integration, and they are comparable with those of content-only and lyrics-only.
- Our bimodal clustering outperforms all other methods in all three performance measures. The bimodal clustering algorithm can be thought as a kind of *semantic* integration of data from different information sources. The performance improvements show that bimodal clustering has advantages over cluster integration. The bimodal clustering aims to minimize the disagreements between different sources and it can implicitly learn the correlation structure between different sets of features.

Experimental comparisons show that our bimodal clustering can efficiently identify song styles. For example, in our experiments, two songs from the album *Utopia/Anthology: Overture Mountain Top And Sunrise Communion With The Sun* and *The Very Last Time* would be put into two different clusters based on their contents or lyrics only. However, using both the content and lyrics, our bimodal clustering algorithm identifies them to be in the same cluster with similar styles. Similarly, bimodal clustering identifies two songs from the album *Peter-Gabriel/Peter Gabriel: Excuse Me* and *Solsbury Hill* to be in the same cluster while other methods do not. In our experiments, we have identified around 50 such pairs and they give good anecdotal evidence that our bimodal clustering algorithm can efficiently identify song styles.

3) *Minimizing Disagreement*: To investigate the relationship between the clustering performance and the agreement with respect to the two sources, we take a closer look at our experiments. Fig. 1 shows the cluster performance (entropy and purity values) and the (dis)agreements between two sources in a trial. Each unit on the X-axis represents five iterations of the algorithm and the Y-axis shows the performance value. We can observe from Fig. 1 that as the agreement between the two sources increases, the clustering quality also tends to increase (i.e., entropy is generally decreasing while purity is increasing).

D. Experimental Results on Constraint-Based Clustering

Thirty constraints (including ten positive constraints and 20 negative constraints) are randomly generated from the cluster labels. We compare the results of constraint clustering with the results obtained when clustering is applied on content without any constraints. Table IV presents the experimental results over ten independent trials.

We observe that constraint-based clustering achieves better performance (i.e., higher purity and accuracy values and lower

TABLE IV
PERFORMANCE COMPARISON. NUMBERS ARE OBTAINED BY AVERAGING OVER TEN TRIALS

Measurement	Purity	Entropy	Accuracy
Without Constraints	0.436	0.731	0.438
With Constraints	0.471	0.723	0.472

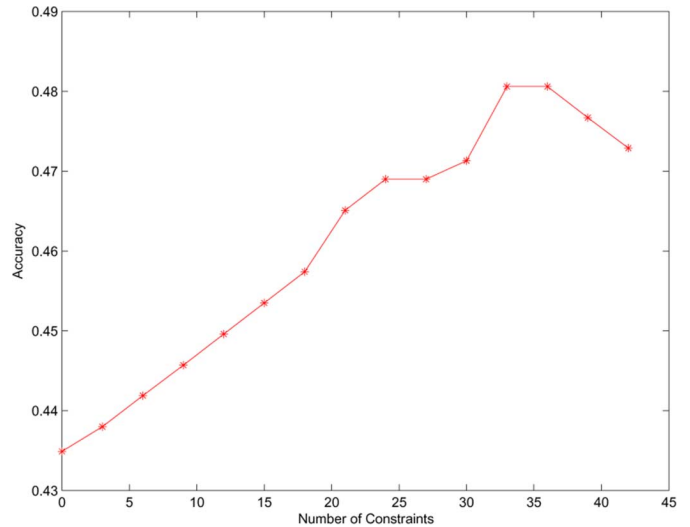


Fig. 2. Comparisons of the clustering accuracy as a function of constraint size.

entropy values) than clustering without any constraints, and that the performance of purity, entropy, and accuracy relative to the other is consistent in our comparison, i.e., higher purity values correspond to lower entropy values, and to higher accuracy values. Note that different evaluation measures consider different aspects of the clustering results. For example, the entropy measure takes into account the entire distribution of the data in a particular cluster and not just the largest class as in the computation of the purity. The accuracy considers the relationships among all pair class-clusters. We hope that these different measures would provide enough information to understand the results of our experiments.

Fig. 2 illustrates the effects of the constraint size. The X-axis of figure shows the number of constraints while the Y-axis shows the clustering accuracy. Here different constraint sizes are tested to investigate the effect of the size of the constraint on the overall clustering performance. An approximate 1:2 ratio of the number of positive constraints to the number of negative constraints is maintained throughout the experiment. We observe that as the constraint set size increases, the accuracy measures steadily improves and flattens out after 40. Then, after that, it looks as if the accuracy was to decrease. This may suggest that too many constraints may force our clustering algorithm to over-fit.

The total number of constraints to specify relations between data elements in an N -element data set is $N(N+1)/2$. In our case, $N = 300$ so the number is 44 850. The number of constraints we used is less than 0.1% of this and thus may look very small. However, the total number of class relations for a K -class data set is K^2 , which is in our case 16. Thus, with 40 constraints we can expect that the class relations are represented at least twice on average. The conspicuous decline in accuracy may suggest that adding more than three constraints per class relation can lower the performance.

VI. CONCLUSION

In this paper, we study the problem on whether multimodal interactive methods can be more powerful than unimodal methods in the case of clustering. In particular, we present a bi-clustering framework for integrating the features based on minimizing disagreement, and also provide a constraint-based clustering framework for clustering music songs in the presence of constraints. Experimental results on a data set consisting of 570 songs from 41 artists of 53 albums show the effectiveness of our approaches.

There are several natural avenues for future research. The first natural direction is on music annotation. How can we automatically and efficiently generate music style or similarity information? Note we did not agree completely with the artist similarity obtained from All Music Guide, but nonetheless used it as the ground truth to evaluate our algorithms in the experiments. Can we incorporate the opinions from music experts or take into account the views from individual users? Second, it would also be interesting to extend the bimodal algorithm by using statistical inference techniques to adaptively weight different data sources during the clustering process. Third, can we incorporate the opinions from music experts or take into account the views from individual users? Fourth, it would also be interesting to evaluate the quality of the generated constraints and explore other constrain-based clustering algorithms. Finally, we would also like to explore the applications of multi-view algorithms in other domains.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] K. P. Bennett, A. Demiriz, and M. J. Embrechts, "Semi-supervised clustering using genetic algorithms," *Artif. Neural Netw. Eng. (ANNIE-99)*, pp. 809–814, 1999.
- [2] S. Abeny, "Bootstrapping," in *Proc. 40th Annu. Meeting Association for Computational Linguistics*, 2002, pp. 360–367.
- [3] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised clustering by seeding," in *Proc. 19th Int. Conf. Machine Learning*, 2002, pp. 27–34.
- [4] S. Basu, M. Bilenko, and R. J. Mooney, "A probabilistic framework for semi-supervised clustering," in *KDD '04: Proc. 2004 ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, New York, 2004, pp. 59–68.
- [5] S. Becker, "Mutual information maximization: Models of cortical self-organization," *Netw.: Comput. Neural Syst.*, vol. 7, no. 1, pp. 7–31, Feb. 1996.
- [6] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proc. Int. Conf. Machine Learning*, 2004.
- [7] E. Bill, "Some advances in transformation-based parts of speech tagging," in *Proc. 12th Nat. Conf. Artificial Intelligence*, 1994, vol. 1, pp. 722–727.
- [8] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Computational Learning Theory (COLT'98)*, 1998, pp. 92–100.
- [9] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proc. Joint SIGDAT Conf. Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [10] R. Cilibrasi, P. Vitányi, and R. De Wolf, "Algorithmic clustering of music based on string compression," *Comput. Music J.*, vol. 28, no. 4, pp. 49–67, 2004.
- [11] S. Dasgupta, M. L. Littman, and D. McAllester, "PAC generalization bounds for co-training," in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: The MIT Press, 2002, pp. 375–382.
- [12] A. David and S. Panchanathan, "Wavelet-histogram method for face recognition," *J. Electron. Imag.*, vol. 9, no. 2, pp. 217–225, 2000.
- [13] I. Davidson and S. S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm," in *Proc. SIAM Int. Conf. Data Mining*, 2005.
- [14] V. R. De Sa and D. Ballard, "Category learning through multi-modality sensing," *Neural Comput.*, vol. 10, no. 5, pp. 1097–1117, 1998.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statist. Soc.*, vol. 39, no. 1, p. 1, 38, 1977.
- [16] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *KDD '06: Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, New York, 2006, pp. 126–135.
- [17] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ICDE*, pp. 341–352, 2005.
- [18] S. Goldman and Y. Zhou, "Enhancing supervised learning with unlabeled data," in *Proc. 17th Int. Conf. Machine Learning (ICML'00)*, San Francisco, CA, 2000, pp. 327–334.
- [19] T. Li and M. Ogihara, "Semi-supervised learning from different information sources," *Knowl. Inf. Syst. J.*, vol. 7, no. 3, pp. 289–309, 2005.
- [20] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proc. 26th Annu. ACM Conf. Research and Development in Information Retrieval (SIGIR 2003)*, 2003, pp. 282–289.
- [21] T. Li, M. Ogihara, and S. Zhu, "Integrating features from different sources for music information retrieval," in *Proc. 2006 IEEE Int. Conf. Data Mining*, 2006, pp. 372–381.
- [22] G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivar. Behav. Res.*, vol. 21, pp. 846–850, 1986.
- [23] R. Mitton, "Spelling checkers, spelling correctors and the misspellings of poor spellers," *Inf. Process. Manage.*, vol. 23, no. 5, pp. 103–209, 1987.
- [24] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data," *Mach. Learn. J.*, vol. 52, no. 1–2, pp. 91–118, 2003.
- [25] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proc. 2000 ACM CIKM Int. Conf. Information and Knowledge Management (CIKM'00)*, 2000, pp. 86–93.
- [26] W. Peng, T. Li, and M. Ogihara, "Music clustering with constraints," in *Proc. 8th Int. Conf. Music Information Retrieval (ISMIR 2007)*, 2007, pp. 27–32.
- [27] E. Pampalk, A. Rauber, and D. Merkl, "Content-based organization and visualization of music archives," *Proc. ACM Multimedia 2002 (ACM MM2002)*, pp. 570–579, 2002.
- [28] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, pp. 846–850, 1971.
- [29] D. Roth and D. Zelenko, "Toward a theory of learning coherent concepts," in *Proc. 17th Nat. Conf. Artificial Intelligence and 12th Conf. Innovative Applications of Artificial Intelligence (AAAI/IAAI'00)*, 2000, pp. 639–644.
- [30] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.
- [31] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.
- [32] W. H. Tsai, D. Rodgers, and H. M. Wang, "Blind clustering of popular music recordings using singer voice characteristics," *Comput. Music J.*, vol. 28, no. 3, pp. 68–78, 2006.
- [33] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained k-means clustering with background knowledge," in *Proc. Int. Conf. Machine Learning*, 2001, pp. 577–584.
- [34] L. Wu, S. L. Oviatt, and P. R. Cohen, "Multimodal integration—A statistical view," *IEEE Trans. Multimedia*, vol. 1, no. 4, pp. 334–341, Dec. 1999.
- [35] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," *Adv. Neural Inf. Process. Syst.* 15, pp. 505–512, 2003.
- [36] Y. Zhao and G. Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Mach. Learn.*, vol. 55, no. 3, pp. 311–331, 2004.



Tao Li received the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 2004.

He is currently an Assistant Professor in the School of Computer Science at Florida International University, Miami. His primary research interests are data mining, machine learning, bioinformatics, and music information retrieval.

Dr. Li is a recipient of NSF CAREER Award in 2006 and IBM Faculty Research Awards (2005, 2007, and 2008).



Bo Shao received the B.S. degree in mining engineering from Northeastern University, Shenyang, China, in 1992, and M.S. degree in computer sciences and applications from Southeast University, Nanjing, China, in 1995. He is currently pursuing the Ph.D. degree at the School of Computing and Information Sciences at Florida International University, Miami.

His primary research interests are music information retrieval and data mining.



Mitsunori Ogihara received the Ph.D. degree in information sciences from Tokyo Institute of Technology, Tokyo, Japan, in 1993.

He is currently a Professor of computer science at the University of Miami, Coral Gables, FL, and Director of Data Mining in the Center for Computational Science at the university.

Dr. Ogihara is a Distinguished Scientist member of the Association for Computing Machinery. He is a recipient of NSF CAREER Award in 1997. He is on the editorial board for the journals *Theory of Computing Systems* and *International Journal of Foundations of Computer Science*.



Shenghuo Zhu received the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 2003.

He is currently a research staff member with NEC Laboratories America, Cupertino, CA. His primary research interests include information retrieval, machine learning, and data mining.



Wei Peng received the B.S. degree in computer science from Xian Polytechnic University, Xian, China, in 2002 and the Ph.D. degree in computer science from Florida International University, Miami, in 2008.

She is currently a member of the research and technical staff in Xerox Innovation Group of Xerox Corporation, Rochester, NY. Her primary research interests are data mining, information retrieval, machine learning, and bioinformatics.