Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 24602, 10 pages
doi:10.1155/2007/24602

*Research Article*

# A Model-Based Approach to Constructing Music Similarity Functions

**Kris West[1] and Paul Lamere[2]**

[1] *School of Computer Sciences, University of East Anglia, Norwich NR4 7TJ, UK*
[2] *Sun Microsystems Laboratories, Sun Microsystems, Inc., Burlington, MA 01803, USA*

Received 1 December 2005; Revised 30 July 2006; Accepted 13 August 2006

Recommended by Ichiro Fujinaga

Several authors have presented systems that estimate the audio similarity of two pieces of music through the calculation of a distance metric, such as the Euclidean distance, between spectral features calculated from the audio, related to the timbre or pitch of the signal. These features can be augmented with other, temporally or rhythmically based features such as zero-crossing rates, beat histograms, or fluctuation patterns to form a more well-rounded music similarity function. It is our contention that perceptual or cultural labels, such as the genre, style, or emotion of the music, are also very important features in the perception of music. These labels help to define complex regions of similarity within the available feature spaces. We demonstrate a machine-learning-based approach to the construction of a similarity metric, which uses this contextual information to project the calculated features into an intermediate space where a music similarity function that incorporates some of the cultural information may be calculated.



## 1. INTRODUCTION

The rapid growth of digital media delivery in recent years has led to an increase in the demand for tools and techniques for managing huge music catalogues. This growth began with peer-to-peer file sharing services, internet radio stations, such as the Shoutcast network, and online music purchase services such as Apple's iTunes music store. Recently, these services have been joined by a host of music subscription services, which allow unlimited access to very large music catalogues, backed by digital media companies or record labels, including offerings from Yahoo, RealNetworks (Rhapsody), BTOpenworld, AOL, MSN, Napster, Listen.com, Streamwaves, and Emusic. By the end of 2006, worldwide online music delivery is expected to be a $2 billion market (http://blogs.zdnet.com/ITFacts/?p=9375).

All online music delivery services share the challenge of providing the right content to each user. A music purchase service will only be able to make sales if it can consistently match users to the content that they are looking for, and users will only remain members of music subscription services while they can find new music that they like. Owing to the size of the music catalogues in use, the existing methods of organizing, browsing, and describing online music collections are unlikely to be sufficient for this task. In order to implement intelligent song suggestion, playlist generation and audio content-based search systems for these services, efficient and accurate systems for estimating the similarity of two pieces of music will need to be defined.

### 1.1. Existing work in similarity metrics

A number of methods for estimating the similarity of pieces of music have been proposed and can be organized into three distinct categories; methods based on metadata, methods based on analysis of the audio content, and methods based on the study of usage patterns related to a music example.

Whitman and Lawrence [1] demonstrated two similarity metrics, the first based on the mining of textual music data retrieved from the web and Usenet for language constructs, the second based on the analysis of user's music collection cooccurrence data downloaded from the OpenNap network. Hu et al. [2] also demonstrated an analysis of textual music data retrieved from the Internet, in the form of music reviews. These reviews were mined in order to identify the genre of the music and to predict the rating applied to the piece by a reviewer. This system can be easily extended to estimate the similarity of two pieces, rather than the similarity of a piece to a genre.

The commercial application Gracenote Playlist [3] uses proprietary metadata, developed by over a thousand in-house editors, to suggest music and generate playlists. Systems based on metadata will only work if the required metadata is both present and accurate. In order to ensure this is the case, Gracenote uses waveform fingerprinting technology, and an analysis of existing metadata in a file's tags, collectively known as Gracenote MusicID [4], to identify examples allowing them to retrieve the relevant metadata from their database. However, this approach will fail when presented with music that has not been reviewed by an editor (as will any metadata-based technique), fingerprinted, or for some reason fails to be identified by the fingerprint (e.g., if it has been encoded at a low bit rate, as part of a mix or from a noisy channel). Shazam Entertainment [5] also provides a music fingerprint identification service, for samples submitted by mobile phone. Shazam implements this content-based search by identifying audio artefacts that survive the codecs used by mobile phones, and by matching them to fingerprints in their database. Metadata for the track is returned to the user along with a purchasing option. This search is limited to retrieving an exact recording of a particular piece and suffers from an inability to identify similar recordings.

Logan and Salomon [6] present an audio content-based method of estimating the "timbral" similarity of two pieces of music based on the comparison of a signature for each track, formed by clustering of Mel-frequency cepstral coefficients (MFCCs) calculated for 30-millisecond frames of the audio signal, with the $K$-means algorithm. The similarity of the two pieces is estimated by the Earth mover's distance (EMD) between the signatures. Although this method ignores much of the temporal information in the signal, it has been successfully applied to playlist generation, artist identification, and genre classification of music.

Pampalk et al. [7] present a similar method applied to the estimation of similarity between tracks, artist identification and genre classification of music. The spectral feature set used is augmented with an estimation of the fluctuation patterns of the MFCC vectors. Efficient classification is performed using a nearest neighbour algorithm also based on the EMD. Pampalk et al. [8] demonstrate the use of this technique for playlist generation, and refine the generated playlists with negative feedback from user's "skipping behaviour."

Aucouturier and Pachet [9] describe a content-based method of similarity estimation also based on the calculation of MFCCs from the audio signal. The MFCCs for each song are used to train a mixture of Gaussian distributions which are compared by sampling in order to estimate the "timbral" similarity of two pieces. Objective evaluation was performed by estimating how often pieces from the same genre were the most similar pieces in a database. Results showed that performance on this task was not very good, although a second subjective evaluation showed that the similarity estimates were reasonably good. Aucouturier and Pachet also report that their system identifies surprising associations between certain pieces often from different genres of music, which they term the "Aha" factor. These associations may be due to confusion between superficially similar timbres of the type described in Section 1.2, which we believe are due to a lack of contextual information attached to the timbres. Aucouturier and Pachet define a weighted combination of their similarity metric with a metric based on textual metadata, allowing the user to increase or decrease the number of these confusions. Unfortunately, the use of textual metadata eliminates many of the benefits of a purely content-based similarity metric.

Ragno et al. [10] demonstrate a different method of estimating similarity based on ordering information in what they describe as expertly authored streams (EAS), which might be any published playlist. The ordered playlists are used to build weighted graphs, which are merged and traversed in order to estimate the similarity of two pieces appearing in the graph. This method of similarity estimation is easily maintained by the addition of new human-authored playlists but will fail when presented with content that has not yet appeared in a playlist.

### 1.2. Common mistakes made by similarity calculations

Initial experiments in the use of the aforementioned content-based "timbral" music similarity techniques showed that the use of simple distance measurements between sets of features, or clusters of features, can produce a number of unfortunate errors, despite generally good performance. Errors are often the result of confusion between superficially similar timbres of sounds, which a human listener might identify as being very dissimilar. A common example might be the confusion of a classical lute timbre, with that of an acoustic guitar string that might be found in folk, pop, or rock music. These two sounds are relatively close together in almost any acoustic feature space and might be identified as similar by a naïve listener, but would likely be placed very far apart by any listener familiar with western music. This may lead to the unlikely confusion of rock music with classical music, and the corruption of any playlist produced.

It is our contention that errors of this type indicate that accurate emulation of the similarity perceived between two examples by human listeners, based directly on the audio content, must be calculated on a scale that is nonlinear with respect to the distance between the raw vectors in the feature space. Therefore, a deeper analysis of the relationship between the acoustic features and the "ad hoc" definition of musical styles must be performed prior to estimating similarity.

In the following sections, we explain our views on the use of contextual or cultural labels such as genre in music description, our goal in the design of a music similarity estimator, and use detail existing work in the extraction of cultural metadata. Finally, we introduce and evaluate a content-based method of estimating the "timbral" similarity of musical audio, which automatically extracts and leverages cultural metadata in the similarity calculation.

### 1.3. Human use of contextual labels in music description

We have observed that when human beings describe music, they often refer to contextual or cultural labels such as membership of a period, genre, or style of music; with reference to similar artists or the emotional content of the music. Such content-based descriptions often refer to two or more labels in a number of fields, for example the music of Damien Marley has been described as "a mix of original dancehall reggae with an R&B/hip hop vibe,"[1] while "Feed me weird things" by Squarepusher has been described as a "jazz track with drum'n'bass beats at high bpm."[2] There are few analogies to this type of description in existing content-based similarity techniques. However, metadata-based methods of similarity judgement often make use of genre metadata applied by human annotators.

### 1.4. Problems with the use of human annotation

There are several obvious problems with the use of metadata labels applied by human annotators. Labels can only be applied to known examples, so novel music cannot be analyzed until it has been annotated. Labels that are applied by a single annotator may not be correct or may not correspond to the point of view of an end user. Amongst the existing sources of metadata there is a tendency to try and define an "exclusive" label set (which is rarely accurate) and only apply a single label to each example, thus losing the ability to combine labels in a description, or to apply a single label to an album of music, potentially mislabelling several tracks. Finally, there is no degree of support for each label, as this is impossible to establish for a subjective judgement, making accurate combination of labels in a description difficult.

### 1.5. Design goals for a similarity estimator

Our goal in the design of a similarity estimator is to build a system that can compare songs based on content, using relationships between features and cultural or contextual information learned from a labelled data set (i.e., producing greater separation between acoustically similar instruments from different contexts or cultures). In order to implement efficient search and recommendation systems, the similarity estimator should be efficient at application time, however, a reasonable index building time is allowed.

The similarity estimator should also be able to develop its own point of view based on the examples it has been given. For example, if fine separation of classical classes is required (baroque, romantic, late romantic, modern), the system should be trained with examples of each class, plus examples from other more distant classes (rock, pop, jazz, etc.) at coarser granularity. This would allow definition of systems for tasks or users, for example, allowing a system to mimic a user's similarity judgements, by using their own music collection as a starting point. For example, if the user only listens to dance music, they will care about fine separation of rhythmic or acoustic styles and will be less sensitive to the nuances of pitch classes, keys, or intonations used in classical music.

## 2. LEARNING MUSICAL RELATIONSHIPS

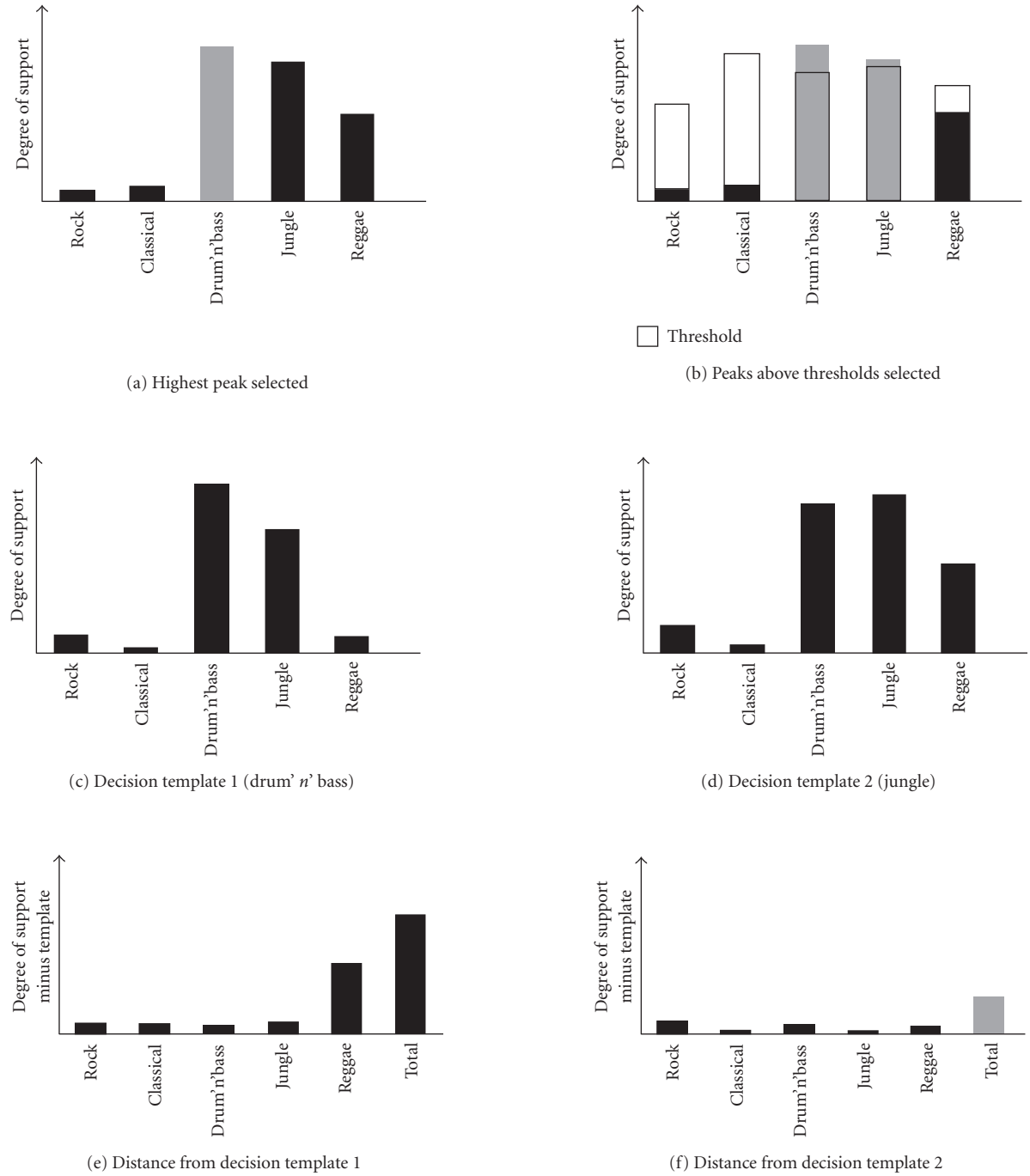
Many systems for the automatic extraction of contextual or cultural information, such as genre or artist metadata, from musical audio have been proposed, and their performances are estimated as part of the annual Music Information Retrieval Evaluation eXchange (MIREX) (see Downie et al. [11]). All of the content-based music similarity techniques, described in Section 1.1, have been used for genre classification (and often the artist identification task) as this task is much easier to evaluate than the similarity between two pieces, because there is a large amount of labelled data already available, whereas music similarity data must be produced in painstaking human listening tests. A full survey of the state of the art in this field is beyond the scope of this paper; however, the MIREX 2005 Contest results [12] give a good overview of each system and its corresponding performance. Unfortunately, the tests performed are relatively small and do not allow us to assess whether the models overfitted an unintended characteristic making performance estimates overoptimistic. Many, if not all of these systems, could also be extended to emotional content or style classification of music; however, there is much less usable metadata available for this task and so few results have been published.

Each of these systems extracts a set of descriptors from the audio content, often attempting to mimic the known processes involved in the human perception of audio. These descriptors are passed into some form of machine learning model which learns to "perceive" or predict the label or labels applied to the examples. At application time, a novel audio example is parameterized and passed to the model, which calculates a degree of support for the hypothesis that each label should be applied to the example.

The output label is often chosen as the label with the highest degree of support (see Figure 1(a)); however, a number of alternative schemes are available as shown in Figure 1. Multiple labels can be applied to an example by defining a threshold for each label, as shown in Figure 1(b), where the outline indicates the thresholds that must be exceeded in order to apply a label. Selection of the highest-peak abstracts information in the degrees of support which could have been used in the final classification decision. One method of leveraging this information is to calculate a "decision template" (see Kuncheva [13, pages 170–175]) for each class of audio (Figures 1(c) and 1(d)), which is normally an average profile for examples of that class. A decision is made by calculating the distance of a profile for an example from the available "decision templates" (Figures 1(e) and 1(f)) and by selecting the closest. Distance metrics used include the Euclidean and Mahalanobis distances. This method can also be used to combine the output from several classifiers, as the "decision

[1] http://cd.ciao.co.uk/Welcome_To_Jamrock_Damian_Marley__Review_5536445.

[2] http://www.bbc.co.uk/music/experimental/reviews/squarepusher_go.shtml.




(a) Highest peak selected

(b) Peaks above thresholds selected

(c) Decision template 1 (drum' *n*' bass)

(d) Decision template 2 (jungle)

(e) Distance from decision template 1

(f) Distance from decision template 2

Figure 1: Selecting an output label from continuous degrees of support.

template" can be very simply extended to contain a degree of support for each label from each classifier. Even when based on a single classifier, a decision template can improve the performance of a classification system that outputs continuous degrees of support, as it can help to resolve common confusions where selecting the highest peak is not always correct. For example, drum and bass tracks always have a similar degree of support to jungle music (being very similar types of music); however, jungle can be reliably identified if there is also a high degree of support for reggae music, which is uncommon for drum and bass profiles.

## 3. MODEL-BASED MUSIC SIMILARITY

If comparison of degree of support profiles can be used to assign an example to the class with the most similar average profile in a decision template system, it is our contention that the same comparison could be made between

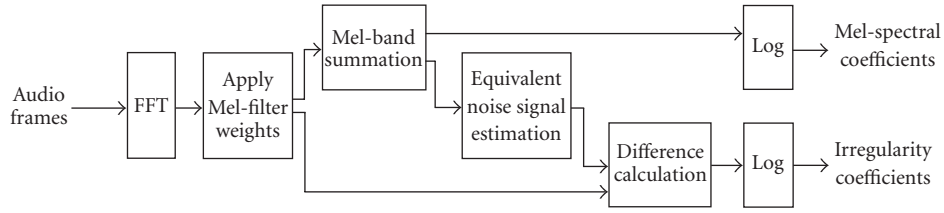



Figure 2: Spectral irregularity calculation.

two examples to calculate the distance between their contexts (where the context might include information about known genres, artists, or moods etc.). For simplicity, we will describe a system based on a single classifier and a "timbral" feature set; however, it is simple to extend this technique to multiple classifiers, multiple label sets (genre, artist, or mood), and feature sets/dimensions of similarity.

Let $P_x = \{c_0^x, \ldots, c_n^x\}$ be the profile for example $x$, where $c_i^x$ is the probability returned by the classifier that example $x$ belongs to class $i$, and $\sum_{i=1}^{n} c_i^x = 1$, which ensures that similarities returned are in the range $[0:1]$. The similarity $S_{A,B}$ between two examples $A$ and $B$ is estimated as one minus the Euclidean distance between their profiles $P_A$ and $P_B$ and is defined as follows:

$$S_{A,B} = 1 - \sqrt{\sum_{i=1}^{n} \left(c_i^A - c_i^B\right)^2}. \quad (1)$$

The contextual similarity score $S_{A,B}$ returned may be used as the final similarity metric or may form part of a weighted combination with another metric based on the similarity of acoustic features or textual metadata. In our own subjective evaluations, we have found that this metric gives acceptable performance when used on its own.

### 3.1. Parameterization of musical audio

In order to train the genre classification models used in the model-based similarity metrics, the audio must be preprocessed and a set of descriptors extracted. The audio signal is divided into a sequence of 50% overlapping, 23 millisecond frames, and a set of novel features collectively known as Mel-frequency spectral irregularities (MFSIs) are extracted to describe the timbre of each frame of audio. MFSIs are calculated from the output of a Mel-frequency scale filter bank and are composed of two sets of coefficients, half describing the spectral envelope and half describing its irregularity. The spectral features are the same as Mel-frequency cepstral coefficients (MFCCs) without the discrete cosine transform (DCT).

The irregularity coefficients are similar to the octave-scale spectral contrast feature as described by Jiang et al. [14], as they include a measure of how different the signal is from white noise in each band. This allows us to differentiate frames from pitched and noisy signals that may have the same spectrum, such as string instruments and drums. Our contention is that this measure comprises important psychoacoustic information which can provide better audio modelling than MFCCs. In our tests, the best audio modelling performance was achieved with the same number of bands of irregularity components as MFCC components, perhaps because they are often being applied to complex mixes of timbres and spectral envelopes. MFSI coefficients are calculated by estimating the difference between the white noise FFT magnitude coefficients that would have produced the spectral coefficient in each band, and the actual coefficients that produced it. Higher values of these coefficients indicate that the energy was highly localized in the band and therefore would have sounded more pitched than noisy.

The features are calculated with 16 filters to reduce the overall number of coefficients. We have experimented with using more filters and a principal components analysis (PCA) or DCT of each set of coefficients, to reduce the size of the feature set, but found performance to be similar using less filters. This property may not be true in all models as both the PCA and DCT reduce both noise within and covariance between the dimensions of the features as do the transformations used in our models (see Section 3.2), reducing or eliminating this benefit from the PCA/DCT.

An overview of the spectral irregularity calculation is given in Figure 2.

As a final step, an onset detection function is calculated and used to segment the sequence of descriptor frames into units corresponding to a single audio event, as described by West and Cox in [15]. The mean and variance of the descriptors are calculated over each segment, to capture the temporal variation of the features. The sequence of mean and variance vectors is used to train the classification models.

The Marsyas [16] software package, a free software framework for the rapid deployment and evaluation of computer audition applications, was used to parameterise the music audio for the Marsyas-based model. A single 30-element summary feature vector was collected for each song. The feature vector represents timbral texture (19 dimensions), rhythmic content (6 dimensions), and pitch content (5 dimensions) of the whole file. The timbral texture is represented by means and variances of the spectral centroid, rolloff, flux and zero crossings, the low-energy component, and the means and variances of the first five MFCCs (excluding the DC component). The rhythmic content is represented by a set of six features derived from the beat histogram for the piece. These include the period and relative amplitude

of the two largest histogram peaks, the ratio of the two largest peaks, and the overall sum of the beat histogram (giving an indication of the overall beat strength). The pitch content is represented by a set of five features derived from the pitch histogram for the piece. These include the period of the maximum peak in the unfolded histogram, the amplitude and period of the maximum peak in the folded histogram, the interval between the two largest peaks in the folded histogram, and an overall confidence measure for the pitch detection. Tzanetakis and Cook [17] describe the derivation and performance of Marsyas and this feature set in detail.

### 3.2. Candidate models

We have evaluated the use of a number of different models, trained on the features described above, to produce the classification likelihoods used in our similarity calculations, including Fisher's criterion linear discriminant analysis (LDA) and a classification and regression tree (CART) of the type proposed by West and Cox in [15] and West [18], which performs a multiclass linear discriminant analysis and fits a pair of single Gaussian distributions in order to split each node in the CART tree. The performance of this classifier was benchmarked during the 2005 Music Information Retrieval Evaluation eXchange (MIREX) (see Downie et al. [11]) and is detailed by Downie in [12].

The similarity calculation requires each classifier to return a real-valued degree of support for each class of audio. This can present a challenge, particularly as our parameterization returns a sequence of vectors for each example and some models, such as the LDA, do not return a well-formatted or reliable degree of support. To get a useful degree of support from the LDA, we classify each frame in the sequence and return the number of frames classified into each class, divided by the total number of frames. In contrast, the CART-based model returns a leaf node in the tree for each vector and the final degree of support is calculated as the percentage of training vectors from each class that reached that node, normalized by the prior probability for vectors of that class in the training set. The normalization step is necessary as we are using variable-length sequences to train the model and cannot assume that we will see the same distribution of classes or file lengths when applying the model. The probabilities are smoothed using Lidstone's law [19] (to avoid a single spurious zero probability eliminating all the likelihoods for a class), the log taken and summed across all the vectors from a single example (equivalent to multiplication of the probabilities). The resulting log likelihoods are normalized so that the final degrees of support sum to 1.

### 3.3. Similarity spaces produced

The degree-of-support profile for each song in a collection, in effect, defines a new intermediate feature set. The intermediate features pinpoint the location of each song in a high-dimensional similarity space. Songs that are close together in this high-dimensional space are similar (in terms of the model used to generate these intermediate features), while songs that are far apart in this space are dissimilar. The intermediate features provide a very compact representation of a song in similarity space. The LDA- and CART-based features require a single floating-point value to represent each of the ten genre likelihoods, for a total of eighty bytes per song which compares favourably to the Marsyas feature set (30 features or 240 bytes), or MFCC mixture models (typically on the order of 200 values or 1600 bytes per song).

A visualization of this similarity space can be a useful tool for exploring a music collection. To visualize the similarity space, we use a stochastically based implementation [20] of multidimensional scaling (MDS) [21], a technique that attempts to best represent song similarity in a low-dimensional representation. The MDS algorithm iteratively calculates a low-dimensional displacement vector for each song in the collection to minimize the difference between the low-dimensional and the high-dimensional distances. The resulting plots represent the song similarity space in two or three dimensions. In the plots in Figure 3, each data point represents a song in similarity space. Songs that are closer together in the plot are more similar according to the corresponding model than songs that are further apart in the plot.

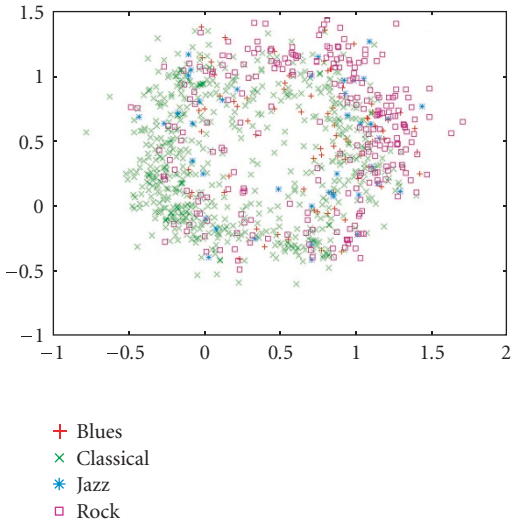
For each plot, about one thousand songs were chosen at random from the test collection. For plotting clarity, the genres of the selected songs were limited to one of "rock," "jazz," "classical," and "blues." The genre labels were derived from the ID3 tags of the MP3 files as assigned by the music publisher.

Figure 3(a) shows the 2-dimensional projection of the Marsyas feature space. From the plot, it is evident that the Marsyas-based model is somewhat successful at separating classical from rock, but is not very successful at separating jazz and blues from each other or from rock and classical genres.

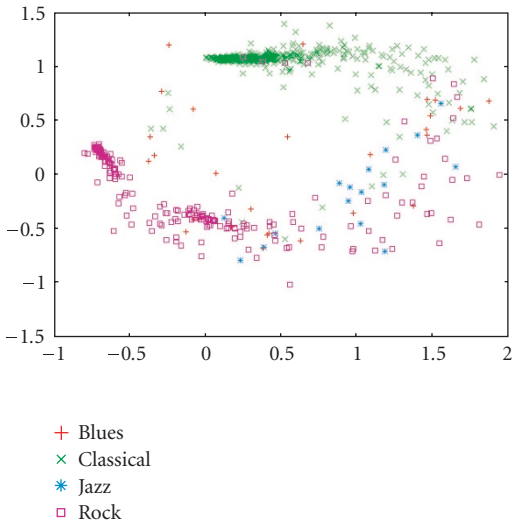
Figure 3(b) shows the 2-dimensional projection of the LDA-based genre model similarity space. In this plot we can see that the separation between classical and rock music is much more distinct than with the Marsyas model. The clustering of jazz has improved, centering in an area between rock and classical. Still, blues has not separated well from the rest of the genres.

Figure 3(c) shows the 2-dimensional projection of the CART-based genre model similarity space. The separation between rock, classical, and jazz is very distinct, while blues is forming a cluster in the jazz neighbourhood and another smaller cluster in a rock neighbourhood. Figure 4 shows two views of a 3-dimensional projection of this same space. In this 3-dimensional view, it is easier to see the clustering and separation of the jazz and the blues data.

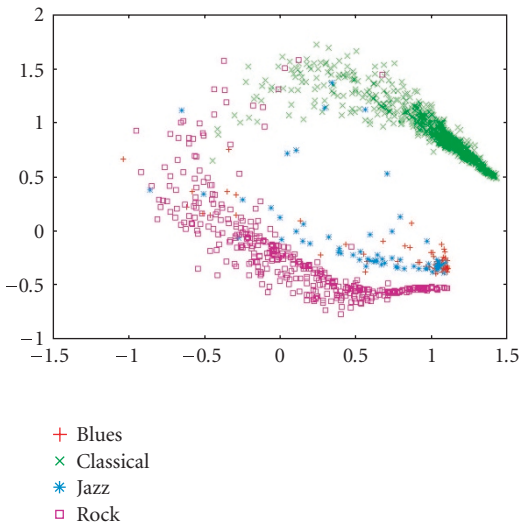
An interesting characteristic of the CART-based visualization is that there is spatial organization even within the genre clusters. For instance, even though the system was trained with a single "classical" label for all Western art music, different "classical" subgenres appear in separate areas within the "classical" cluster. Harpsichord music is near other harpsichord music while being separated from choral and string quartet music. This intracluster organization is a key attribute of a visualization that is to be used for music collection exploration.




(a)

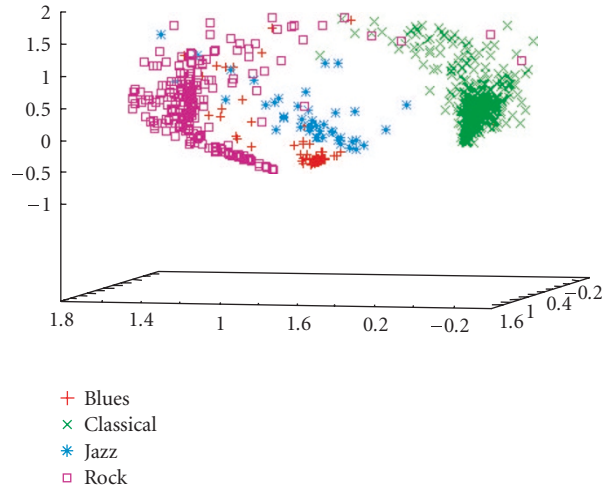



(b)

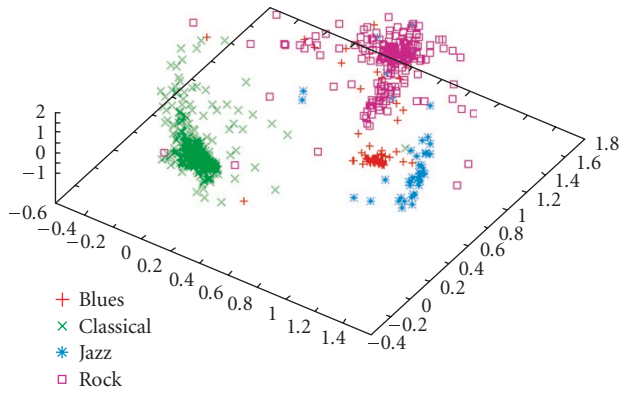



(c)

Figure 3: Similarity spaces produced by (a) Marsyas features, (b) an LDA genre model, and (c) a CART-based model.




(a)




(b)

Figure 4: Two views of a 3D projection of the similarity space produced by the CART-based model.

## 4. EVALUATING MODEL-BASED MUSIC SIMILARITY

### 4.1. Challenges

The performance of music similarity metrics is particularly hard to evaluate as we are trying to emulate a subjective perceptual judgement. Therefore, it is both difficult to achieve a consensus between annotators and nearly impossible to accurately quantify judgements. A common solution to this problem is to use the system one wants to evaluate to perform a task, related to music similarity, for which there already exists ground-truth metadata, such as classification of music into genres or artist identification. Care must be taken in evaluations of this type as overfitting of features on small test collections can give misleading results.

#### 4.1.1. Data set

The algorithms presented in this paper were evaluated using MP3 files from the Magnatune collection [22]. This collection consists of 4510 tracks from 337 albums by 195 artists

Table 1: Genre distribution for Magnatune data set.

| Genre | Number | Genre | Number |
|---|---|---|---|
| Acid | 9 | Other | 8 |
| Ambient | 156 | Pop | 42 |
| Blues | 113 | Punk | 101 |
| Celtic | 24 | Punk Rock | 37 |
| Classical | 1871 | Retro | 14 |
| Electronic | 529 | Rock | 486 |
| Ethnic | 600 | Techno | 10 |
| Folk | 71 | Trance | 9 |
| Hard rock | 52 | Trip-Hop | 7 |
| Industrial | 29 | Unknown | 17 |
| Instrumental | 11 | New Age | 202 |
| Jazz | 64 | Metal | 48 |

Table 2: Genre distribution used in training models.

| Genre | Training instances |
|---|---|
| Ambient | 100 |
| Blues | 100 |
| Classical | 250 |
| Electronic | 250 |
| Ethnic | 250 |
| Folk | 71 |
| Jazz | 64 |
| New age | 100 |
| Punk | 100 |
| Rock | 250 |

representing twenty four genres. The overall genre distributions are shown in Table 1.

The LDA and CART models were trained on 1535 examples from this database using the 10 most frequently occurring genres. Table 2 shows the distribution of genres used in training the models. These models were then applied to the remaining 2975 songs in the collection in order to generate a degree-of-support profile vector for each song. The Marsyas model was generated by collecting the 30 Marsyas features for each of the 2975 songs.

### 4.2. Evaluation metric

#### 4.2.1. Distance measure statistics

We first use a technique described by Logan and Salomon [6] to examine some overall statistics of the distance measure. Table 3 shows the average distance between songs for the entire database of 2975 songs. We also show the average distance between songs of the same genre, songs by the same artist, and songs on the same album. From Table 3 we see that all three models correctly assign smaller distances to songs in the same genre, than the overall average distance, with even smaller distances assigned for songs by the same artist on the same album. The LDA- and CART-based models assign significantly lower genre, artist, and album distances compared to the Marsyas model, confirming the impression given in Figure 2 that the LDA- and CART-based models are doing a better job of clustering the songs in a way that agrees with the labels and possibly human perceptions.

Table 3: Statistics of the distance measure.

| Average distance between songs | | | | |
|---|---|---|---|---|
| Model | All songs | Same genre | Same artist | Same album |
| Marsyas | 1.17 | 1.08 | 0.91 | 0.84 |
| LDA | 1.22 | 0.72 | 0.59 | 0.51 |
| CART | 1.21 | 0.60 | 0.48 | 0.38 |

Table 4: Average number of closest songs with the same genre.

| Model | Closest 5 | Closest 10 | Closest 20 |
|---|---|---|---|
| Marsyas | 2.57 | 4.96 | 9.53 |
| LDA | 2.77 | 5.434 | 10.65 |
| CART | 3.01 | 6.71 | 13.99 |

Table 5: Average number of closest songs with the same artist.

| Model | Closest 5 | Closest 10 | Closest 20 |
|---|---|---|---|
| Marsyas | 0.71 | 1.15 | 1.84 |
| LDA | 0.90 | 1.57 | 2.71 |
| CART | 1.46 | 2.60 | 4.45 |

Table 6: Average number of closest songs occurring on the same album.

| Model | Closest 5 | Closest 10 | Closest 20 |
|---|---|---|---|
| Marsyas | 0.42 | 0.87 | 0.99 |
| LDA | 0.56 | 0.96 | 1.55 |
| CART | 0.96 | 1.64 | 2.65 |

#### 4.2.2. Objective relevance

We use the technique described by Logan and Salomon [6] to examine the relevance of the top $N$ songs returned by each model in response to a query song. We examine three objective definitions of relevance: songs in the same genre, songs by the same artist, and songs on the same album. For each song in our database, we analyze the top 5, 10, and 20 most similar songs according to each model.

Tables 4, 5, and 6 show the average number of songs returned by each model that has the same genre, artist, and album label as the query song. The genre for a song is determined by the ID3 tag for the MP3 file and is assigned by the music publisher.

Table 7: Time required to calculate two-million distance.

| Model | Time |
|---|---|
| Marsyas | 0.77 seconds |
| LDA | 0.41 seconds |
| CART | 0.41 seconds |

#### 4.2.3. Runtime performance

An important aspect of a music recommendation system is its runtime performance on large collections of music. Typical online music stores contain several million songs. A viable song similarity metric must be able to process such a collection in a reasonable amount of time. Modern, high-performance text search engines such as Google have conditioned users to expect query-response times of under a second for any type of queries. A music recommender system that uses a similarity distance metric will need to be able to calculate on the order of two-million-song distances per second in order to meet the user's expectations of speed. Table 7 shows the amount of time required to calculate two million distances. Performance data was collected on a system with a 2 GHz AMD Turion 64 CPU running the Java HotSpot(TM) 64-Bit Server VM (version 1.5).

These times compare favourably to stochastic distance metrics such as a Monte Carlo sampling approximation. Pampalk et al. [7] describe a CPU performance-optimized Monte Carlo system that calculates 15554 distances in 20.98 seconds. Extrapolating to two-million-distance calculations yields a runtime of 2697.61 seconds or 6580 times slower than the CART-based model.

Another use for a song similarity metric is to create playlists on handheld music players such as the iPod. These devices typically have slow CPUs (when compared to desktop or server systems), and limited memory. A typical hand held music player will have a CPU that performs at one hundredth the speed of a desktop system. However, the number of songs typically managed by a handheld player is also greatly reduced. With current technology, a large-capacity player will manage 20 000 songs. Therefore, even though the CPU power is one hundred times less, the search space is one hundred times smaller. A system that performs well indexing a 2 000 000 song database with a high-end CPU should perform equally well on the much slower handheld device with the correspondingly smaller music collection.

## 5. CONCLUSIONS

We have presented improvements to a content-based, "timbral" music similarity function that appears to produce much better estimations of similarity than existing techniques. Our evaluation shows that the use of a genre classification model, as part of the similarity calculation, not only yields a higher number of songs from the same genre as the query song, but also a higher number of songs from the same artist and album. These gains are important as the model was not trained on this metadata, but still provides useful information for these tasks.

Although this is not a perfect evaluation, it does indicate that there are real gains in accuracy to be made using this technique, coupled with a significant reduction in runtime. An ideal evaluation would involve large-scale listening tests. However, the ranking of a large music collection is difficult and it has been shown that there is large potential for overfitting on small test collections [7]. At present, the most common form of evaluation of music similarity techniques is the performance on the classification of audio into genres. These experiments are often limited in scope due to the scarcity of freely available annotated data and do not directly evaluate the performance of the system on the intended task (genre classification being only a facet of audio similarity). Alternatives should be explored for future work.

Further work on this technique will evaluate the extension of the retrieval system to likelihoods from multiple models and feature sets, such as a rhythmic classification model, to form a more well-rounded music similarity function. These likelihoods will either be integrated by simple concatenation (late integration) or through a constrained regression on an independent data set (early integration) [13].

## ACKNOWLEDGMENTS

The experiments in this document were implemented in the M2K framework [23] (developed by the University of Illinois, the University of East Anglia, and Sun Microsystems Laboratories), for the D2K Toolkit [24] (developed by the Automated Learning Group at the NCSA) and were evaluated on music from the Magnatune Label [22], which is available on a Creative Commons License that allows academic use.

## REFERENCES

[1] B. Whitman and S. Lawrence, "Inferring descriptions and similarity for music from community metadata," in *Proceedings of the International Computer Music Conference (ICMC '02)*, pp. 591–598, Göteborg, Sweden, September 2002.

[2] X. Hu, J. S. Downie, K. West, and A. F. Ehmann, "Mining music reviews: promising preliminary results," in *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR '05)*, pp. 536–539, London, UK, September 2005.

[3] Gracenote, Gracenote Playlist. 2005. http://www.gracenote.com/gn_products/.

[4] Gracenote, Gracenote MusicID. 2005. http://www.gracenote.com/gn_products/.

[5] A. Wang, "Shazam Entertainment," ISMIR 2003 - Presentation. http://ismir2003.ismir.net/.

[6] B. Logan and A. Salomon, "A music similarity function based on signal analysis," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '01)*, pp. 745–748, Tokyo, Japan, August 2001.

[7] E. Pampalk, A. Flexer, and G. Widmer, "Improvements of audio-based music similarity and genre classificaton," in *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR '05)*, pp. 628–633, London, UK, September 2005.

[8] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic playlist generation based on skipping behavior," in *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR '05)*, pp. 634–637, London, UK, September 2005.

[9] J.-J. Aucouturier and F. Pachet, "Music similarity measures: what's the use?" in *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR '02)*, Paris, France, October 2002.

[10] R. Ragno, C. J. C. Burges, and C. Herley, "Inferring similarity between music objects with application to playlist generation," in *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, Singapore, Republic of Singapore, November 2005.

[11] J. S. Downie, K. West, A. F. Ehmann, and E. Vincent, "The 2005 music information retrieval evaluation exchange (MIREX 2005): preliminary overview," in *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR '05)*, pp. 320–323, London, UK, September 2005.

[12] J. S. Downie, "MIREX 2005 Contest Results," http://www.music-ir.org/evaluation/mirex-results/.

[13] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, New York, NY, USA, 2004.

[14] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '02)*, vol. 1, pp. 113–116, Lausanne, Switzerland, August 2002.

[15] K. West and S. Cox, "Finding an optimal segmentation for audio genre classification," in *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR '05)*, pp. 680–685, London, UK, September 2005.

[16] G. Tzanetakis, "Marsyas: a software framework for computer audition," October 2003, http://marsyas.sourceforge.net/.

[17] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[18] K. West, "MIREX Audio Genre Classification," 2005, http://www.music-ir.org/evaluation/mirex-results/articles/audio_genre/.

[19] G. J. Lidstone, "Note on the general case of the bayeslaplace formula for inductive or a posteriori probabilities," *Transactions of the Faculty of Actuaries*, vol. 8, pp. 182–192, 1920.

[20] M. Chalmers, "A linear iteration time layout algorithm for visualising high-dimensional data," in *Proceedings of the 7th IEEE Conference on Visualization*, San Francisco, Calif, USA, October 1996.

[21] J. B. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.

[22] Magnatune, "Magnatune: MP3 music and music licensing (royalty free music and license music)," 2005, http://magnatune.com/.

[23] J. S. Downie, "M2K (Music-to-Knowledge): a tool set for MIR/MDL development and evaluation," 2005, http://www.music-ir.org/evaluation/m2k/index.html.

[24] National Center for Supercomputing Applications, ALG: D2K Overview. 2004. http://alg.ncsa.uiuc.edu/do/tools/d2k.

**Kris West** is a Ph.D. researcher at The School of Computing Sciences, Univesity of East Anglia, where he is researching automated music classification, similarity estimation, and indexing. He interned with Sun labs in 2005 on the Search Inside the Music project, where he developed features, algorithms, and frameworks for music similarity estimation and classification. He is a principal developer of the Music-2-Knowledge (M2K) project at the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL), which provides tools, frameworks, and a common evaluation structure for music information retrieval (MIR) researchers. He has also served on the Music Information Retrieval Evaluation Exchange (MIREX) steering commitee and helped to organize international audio artist identification, genre classification, and music search competitions.

**Paul Lamere** is a Principal Investigator for a project called Search Inside the Music, at Sun Labs, where he explores new ways to help people find highly relevant music, even as music collections get very large. He joined Sun Labs, in 2000, where he worked in the Lab's Speech Application Group, contributing to FreeTTS, a speech synthesizer written in the Java programming language, as well as serving as the Software Architect for Sphinx-4, a speech-recognition system written in the Java programming language. Prior to joining Sun, he developed real-time embedded software for a wide range of companies and industries. He has served on a number of standards committees including the W3C Voice Browser working group, the Java Community Process JSR-113 working on the next version of the Java Speech API, the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL), and the Music Information Retrieval Evalua.