A Middleware System for Web-based Digital Music Libraries

Aristomenis S. Lampropoulos, Paraskevi S. Lampropoulou, and George A. Tsihrintzis
University of Piraeus
Department of Informatics
80 Karaoli and Dimitriou St., Piraeus 185 34, Greece.
{arislamp,vlamp,geoatsi}@unipi.gr

Abstract

We present a middleware system that facilitates Internet users' access to web-based digital music libraries and allows them to manipulate audio meta-information taking into consideration content and semantic information of music data. Useful relations in the data are automatically extracted through semantic networks (constructed and maintained in the library). Our system is complemented with a query-by-example retrieval subsystem, user relevance feedback facilities, and a new approach for musical genre classification based on the features extracted from signals that correspond to distinct musical instrument sources, as these sources have been identified by a source separation process. The system operation is illustrated in detail.

1 Introduction

Recent advances in digital storage technology and the huge increase in the availability of digital music have led to the creation of large music collections for use by broad classes of computer users. In turn, this fact gives rise to a need for systems that have the ability to manage and organize efficiently large collections of stored music files. Many currently available music search engines and peer-to-peer systems (e.g., Kazaa [18], eMule [19], etc.) rely on textual meta-information, such as file names and ID3 tags, as the retrieval mechanism. This textual description of audio information is subjective and does not make use of the musical content, while the relevant meta-data have to be entered and updated manually. In turn, this implies significant effort in both creating and maintaining the music database. Therefore, it is expected that extracting the information from the actual music data through an automated process could overcome some of these problems.

There have been many works on audio content analysis using various features and methods [1], [2], [3], [4], [6]. Most of them focused on automatic musical genre classification which provides techniques to organize digital music into categorical labels created by human experts using objective features of audio signals related to instrumentation, timbral texture, and rhythmic and pitch content [1], [2]. In its strictest form, automatic musical genre classification means classification of music pieces into well-defined classes. This is a difficult process due to the facts that on one hand many music pieces may lie on the boundaries between genres and on the other hand music is an evolving art and, thus, performers and composers may be influenced by more than one genres. However, it has been observed that audio signals corresponding to music that belongs to the same genre share certain common characteristics as they are performed by similar types of instruments and have similar pitch distribution and rhythmic patterns [7].

Given these observations, we present an integrated middleware system for automatic organization of a web-based collection of music files, whose architecture is based on three distinct, yet inter-related processes, namely: (a) audio data preprocessing (format normalization and extraction of features), (b) retrieval and (c) visualization/browsing. The organization of music files is achieved by fuzzy c-means clustering according to musical surface features and tempo and visualized via a dendrogram in which every leaf of the tree represents a music file and musical similarity decreases with tree height. Our system is complemented with a query-by-example retrieval subsystem, the results of which are also returned in dendrogram form to the user.

Additionally, our system is complemented with semantic meta-data, which include knowledge about artist, composer, lyricist, genre, album or the collection in which a music file is included, title of a music file, keywords about impact on mood (e.g., relax, happy etc), keywords from lyrics and musical instruments. Our system offers the user the ability to check the clustering result, correct existing textual meta-information (such as genre and album) and add further semantic information about lyrics, theme, other artists who perform similar music, musical instruments, etc. This step is realized via a relevance feedback process and imposes
consistency and reliability between content and semantic information in the music library. As a result, future queries retrieve music files that users perceive as more relevant to their queries and the music retrieval process is significantly improved as it combines objective and perceived (subjective) type similarity.

Finally, we have incorporated into our system a new approach for music genre classification based on the features extracted from signals that correspond to distinct musical instrument sources. Contrary to previous works, our approach uses first a sound source separation method to decompose the audio signal into a number of component signals, each of which corresponds to a different musical instrument source. In this way timbral, rhythmic and pitch features are extracted from separate instrument sources and used to classify a music clip, detect its various musical instruments sources and classify them into a musical dictionary of instrument sources or instrument team. This resembles a human listener who is able to determine the genre of a music signal and, at the same time, distinguish a number of different musical instruments from a complex sound structure.

More specifically, the paper is organized as follows: Section 2 presents a review of existing digital libraries, while Section 3 presents an overview of our system architecture. Section 4 describes the meta-information used in the system and Section 5 describes the source separation technique, the classification part and the results, as well. Section 6 describes the relevance feedback process. Finally conclusions and future research directions are given in Section 7.

2 Review of Existing Digital Music Libraries

There are many implementations of digital libraries based on different techniques over the last years. One of the most important digital music library system is VARIATIONS [12] which provides online access to over 5,000 sound recordings from Indiana University’s library collection by relying on a client-server architecture. In VARIATIONS, information retrieval is facilitated through the employment of meta-data. This means that each sound object is accompanied with an enhanced USMARC bibliographic record, which is stored in an ASCII text file. The platform-specific nature of the corresponding architecture results in a library with limited access capability.

Another application that provides online access to a distributed collection (according to the 3-tier architectural model) of sound recordings is JUKEBOX [13]. Although this system relies on the mp3 format for the encoding of its sound archives, access to the collection is once again limited due to the fact that decoding at the client side requires a specific hardware decoder.

A widely available system, in which the user interface works within a Web browser, is the PATRON [14]. The architecture of the PATRON system follows the client-server model and has been designed at the University of Surrey to provide audio (and video) on demand. The main characteristic of this system is that it is based on a meta-data repository that is structured according to the XML specifications.

Moreover, a digital library of music and audio has been developed in New Zealand according to an innovative approach for sound collections. This approach, instead of using bibliographic meta-data for the retrieval of requested files, uses a ”melody index” system that retrieves music on the basis of a few notes that are sung, hummed or otherwise entered. Related content-based music retrieval techniques are employed for querying and browsing music collections of the SMILE [15] system. Finally recent systems for music delivery on the Web like Napster [16] and Gnutella [17], rely on peer-to-peer architectures for organizing their mp3 file collections and facilitating information retrieval for their users.

3 System Architecture

Our middleware system, illustrated in Figure 1, was embedded in a web multi-tier application which extended our primary desktop application [4], [5]. The middleware system of our web application includes a feature extractor module, whose task is to capture the low-level (content-based) features of music files. This module is complemented with the Lame decoder [23] which normalizes the format of all music files, so that a query-by-example searches for data that are globally similar to the query independently of the query scale. Thus, while each music piece is allowed to be stored in any audio file format, such as .mp3, .au, or .wav, all records in a database are re-sampled into a database-wide uniform sample rate. For feature extraction, first we apply format normalization by decoding each file to raw Pulse Code Modulation (PCM) and converting it to the .wav format with audio CD quality and resolution of 16 bit samples at a sampling rate of 44.1 kHz. From each file, a randomly chosen sample of duration of 30 seconds is used as input to the feature extraction module described in Section 3.

The second module is an ID3 tag extractor and an interface for the user to update the extracted semantic meta-data and add other possible semantic keywords.

The third module realizes the process of fuzzy c-means clustering (FCM) [9], which is used for the initialization of our system and applied on low-level content-based features to reveal the predominant classes in the database. Having the clustering result, the user is allowed to annotate clusters with keywords of semantic knowledge such as albums, artists, etc. and may create semantic information even for those files to which no initial semantic meta-data are anno-
terizes the music file in an objective (low-level) way and allows more efficient computerized processing. Specifically, in our system, we used a 30-dimensional objective feature vector which was originally proposed in [1] and subsequently used in other works [2, 4, 5]. The feature vector consists of three different types of features, namely rhythm-related (Beat), timbral texture (musical surface: STFT, MFCCs) and pitch content-related features. For the extraction of the feature vectors corresponding to the music files, we used MARSYAS 0.1 [2], a public software framework for computer audition applications.

Content independent meta-data are not directly related to the audio signal content and could not necessarily be extracted from it. This type of meta-information is subjective and textual and offers a semantic (high level) description of music files. The best known example of this meta-information is probably the ID3 tags, an extension to the popular mp3 format, which allows the user to add tags to the beginning or end of the file, which contain such information as song title, artist, album name, genre, etc.

The combination of content-independent and content-dependent meta-data in our system is crucial. Indeed, the content-independent meta-data are useful to the music tutor in describing the domain knowledge and creating hierarchies, while, on the other hand, the content dependent meta-data are useful to him/her for exploring music data.

5 Source Separation Technique

The problem of separating the component signals that correspond to the distinct musical instruments that generated an audio signal is ill-defined as there is no prior knowledge about the instrumental sources. Many techniques have been successfully used to solve the general blind source separation problem in several application areas; among these, the Independent Component Analysis (ICA) method [20], [23] appears to be one of the most promising. ICA assumes that the individual source components in an unknown mixture have the property of mutual statistical independence [21]. This property is exploited in order to algorithmically identify the latent sources. Moreover, ICA-based methods require certain limiting assumptions, such as the assumption that the number of observed mixture signals be at least as high as the number of source signals and that the mixing matrix be full rank. However, a method has been proposed which is based on ICA but relaxes the constraint on the number of observed mixture signals. This is called the Independent Subspace Analysis (ISA) method and can separate individual sources from a single-channel mixture by using sound spectra [20].

Signal independence is the main assumption of both the ICA and ISA methods. In musical signals, however, there exist dependencies in both the time and frequency domains.

4 Meta-Information used in the System

Two types of meta-information are used for the purposes of the description and quantification of the relations between data and hierarchies in a music collection. Specifically, these types are (1) content-dependent meta-data and (2) content-independent meta-data.

Content-dependent meta-data are perceptual facts which can be automatically extracted from audio signals. There exists an extensive variety of features that can be extracted from raw audio data using either time-domain or frequency-domain (spectral) representations. These features are grouped into a feature vector for each music file, which constitutes a numerical representation that characterizes the music file in an objective (low-level) way and allows more efficient computerized processing. Specifically, in our system, we used a 30-dimensional objective feature vector which was originally proposed in [1] and subsequently used in other works [2, 4, 5]. The feature vector consists of three different types of features, namely rhythm-related (Beat), timbral texture (musical surface: STFT, MFCCs) and pitch content-related features. For the extraction of the feature vectors corresponding to the music files, we used MARSYAS 0.1 [2], a public software framework for computer audition applications.

Content independent meta-data are not directly related to the audio signal content and could not necessarily be extracted from it. This type of meta-information is subjective and textual and offers a semantic (high level) description of music files. The best known example of this meta-information is probably the ID3 tags, an extension to the popular mp3 format, which allows the user to add tags to the beginning or end of the file, which contain such information as song title, artist, album name, genre, etc.

The combination of content-independent and content-dependent meta-data in our system is crucial. Indeed, the content-independent meta-data are useful to the music tutor in describing the domain knowledge and creating hierarchies, while, on the other hand, the content dependent meta-data are useful to him/her for exploring music data.
To overcome these limitations, we use in our system a recently proposed data-adaptive algorithm that is similar to ICA and called Convolutive Sparse Coding (CSC) [20]. This method is presented in detail in the following Section 2.1.

5.1 Source Separation using improved Convolutive Sparse Coding

For source separation, we make the assumption that only parts of the signal are adequate to apply the CSC algorithm. With this assumption we overcome the huge computational load of applying the CSC to the entire input signal. The first step for the source separation technique is to find the length of each part. Then, we examine three parts, one at the beginning, one at the middle and one at the end of the signal, as shown in Figure 2. We apply the CSC algorithm to the three signal parts in parallel. We choose the method of convolutive sparse coding because it solves, at least partially, the assumptions of fixed spectra over time and the model fitting criterion of the reconstruction error which are not correct for audio signals. Moreover, this technique uses compression and enables higher perceptual quality of separated sources.

The basic signal model in general sparse coding is that each observation vector \( x_i \) is a linear mixture of source vectors \( s_j \):

\[
x_i = \sum_{j=1}^{J} a_{i,j} s_j, \quad i = 1, ..., I,
\]

where \( a_{i,j} \) is the weight of \( j^{th} \) source in the \( i^{th} \) observation signal.

Both the source vectors and the weights are assumed unknown. The sources are obtained by multiplying the observation matrix by the estimate of an unmixing matrix. The main assumption in sparse coding techniques is that the sources are non-active most of the time, which means that the mixing matrix has to be sparse. The estimation can be done using a cost function that minimizes the reconstruction error and maximizes the sparseness of the mixing matrix. More specifically, this method is called convolutive sparse coding because the source model is formulated as the convolution of a source spectrogram and an onset vector. The suitability of this model over-covers the case of respective transient sources.

The input part of signal is represented using the magnitude spectrogram, which is calculated as follows: first, the time domain input signal is divided into frames and windowed with a fixed 40 ms Hamming window with 50% overlap between frames. Next, each frame is transformed into the frequency domain by computing its discrete Fourier transform (DFT) of length equal to the window size. Only positive frequencies are retained and phases are discarded by keeping only the magnitude of the DFT spectra. This results in a spectrogram \( x_{f,t} \), where \( f \) is a discrete frequency index and \( t \) is a frame index. A two-dimensional magnitude spectrogram is used to characterize one event of a source at discrete frequency \( f \), \( t \) frames as the onset varies between 0 and \( D \).

5.1.1 The iterative algorithm:

The magnitudes \( x_{f,t} \) and weights \( w_{f,t} \) are calculated. The number of sources \( N \) is set by hand. \( N \) should be equal to the number of clearly distinguishable instruments. If the spectrum of one source varies significantly, for example because of accentuation, one may have to use more than one component per source. The model considers the different fundamental frequencies of each instrument as separate sources. Initialize \( a_1...a_n \) with the absolute values of Gaussian noise.

\textbf{Iteration:}

1. Update \( s_{f,t} \) using the multiplicative step

\[
\hat{s}_{f,t}^{(p+1)} = \frac{1}{A^T W_f T W_f x_f} \cdot \frac{1}{A^T W_f T W_f} A s_{f,t}^{(p)},
\]

where the \( s_{f,t}^{(p+1)} \) is the updated \( s_{f,t}^{(p)} \) for \( p^{th} \) iteration given the \( A, W_f \).

2. Calculate \( \nabla a_n = \frac{\partial c_{tot}(\lambda)}{\partial a_n} \).

3. Update \( a_n \leftarrow a_n - \mu_n \nabla a_n \). Set the negative \( a_n \) elements to zero. \( \mu_n \) is the step size, which is adaptively set.

4. Evaluate the cost function.

5. Repeat Steps 1-4 until the value of the cost function remains unchanged.

In the synthesis mode, the convolutions are evaluated to get frame-wide magnitudes of each source. To get the com-


5.2 Classification Results

We have tried and evaluated different classifiers contained in the machine learning tool called WEKA [24], which we have connected to our system. Specifically, the classifiers tested were: (1) a nearest neighbor classifier (NN), (2) a multilayer perceptron with four hidden layers (MLP4), (3) a multilayer perceptron with ten hidden layers (MLP10), and (4) the classifier AdaBoostM1 (ABM1). The classifier input was the feature vector corresponding to the component signals produced by source separation. The number of audio classes we wanted to classify into was six in this work. Classification results were calculated using 10-fold cross-validation evaluation, where the dataset to be evaluated was randomly partitioned so that 10% be used for testing and 90% be used for training. This process was iterated with different random partitions and the results were averaged. This ensured that the calculated accuracy was not biased because of the particular partitioning of training and testing. Our data set consists of 1100 music pieces from Greek songs of 4 genres (Rebetico, Dimotiko, Laiko, and Edehno). Each music piece is represented by a 30-dimensional objective feature vector which consists of three different types of features: a) rhythm related (Beat) features, b) timbral texture (musical surface: STFT, MFCCs) features, and c) pitch content-related features.

As seen in Table 1, the results after implementation of the source separation technique had an improvement of 1% - 2%. This was due to the fact that the source separation technique revealed more information about timbral texture, rhythm and pitch (harmony) content, not only for the signal as a whole, but for a number of the separated instrument team sources.

![Diagram of Relevance Feedback](Figure 3. Relevance Feedback)

### Table 1. Correctly Classified Instances without/with Source Separation

<table>
<thead>
<tr>
<th>Classifier</th>
<th>% w/out SS</th>
<th>% with SS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>67.5882</td>
<td>68.1655</td>
</tr>
<tr>
<td>MLP4</td>
<td>73.2126</td>
<td>74.8946</td>
</tr>
<tr>
<td>MLP10</td>
<td>73.9752</td>
<td>75.0164</td>
</tr>
<tr>
<td>ABM1</td>
<td>75.8818</td>
<td>77.4978</td>
</tr>
</tbody>
</table>

6 Relevance Feedback

Relevance feedback is an iterative process at each stage of which retrievals are updated and refined on the basis of user-supplied feedback, expecting that the updated retrieval set will be (perceived by a specific user as) more similar to the query. Relevance feedback adds considerable performance boost over retrieval systems without relevance feedback and typically follows the scenario:

1. Initial retrieval results are provided through a query-by-example; 2. the user judges the currently retrieved files according to their degree of relevance to his/her request; 3. the system analyzes the user feedback and goes to Step 2 to repeat the retrieval process.

In our system, we follow the framework in [10], which unifies semantics and content-based features in image retrieval and uses user feedback to learn semantic relevance between image clusters and improve the retrieval performance. This framework maintains the strength of low-level (content-based) feature retrieval while incorporating learning and annotations through the relevance feedback process. It has been implemented into iFind, a web-based image retrieval system developed by Microsoft Research, China [11], and experiments have shown improvement in retrieval performance in a given query session and utilization of knowledge learnt from previous queries to reduce the number of iterations in subsequent queries. In our system, we allow relevance feedback on both music semantic keywords (herein artist name, genre, etc.) and the music content-based features as shown in Figure 3 along lines similar to those in [10],[11].

7 Conclusions and Future Work

We presented a middleware system for web-based digital music libraries, which provides a flexible semi-automatic interface combining (a) audio data preprocessing (format normalization and feature extraction), (b) retrieval and (c) visualization/browsing facilities. The organization of mu-
usic files is achieved by fuzzy c-means clustering according to musical surface features and tempo and visualized via a dendrogram in which every leaf of the tree represents a music file and musical similarity decreases with tree height. Our system is complemented with a query-by-example retrieval subsystem (the results of which are also returned to the user in dendrogram form) and with semantic metadata which offer the user the ability to check the clustering result, correct existing textual meta-information (such as genre and album) and add additional semantic information (about lyrics, theme, other artists who perform similar music, musical instruments, etc). Furthermore, we presented a new approach for musical genre classification based on the features extracted from signals that correspond to distinct musical instrument sources, as these sources have been identified by a source separation process. The operation of our system was illustrated in detail.

Currently, we are in the process of improving further the classification - retrieval accuracy and perception efficiency of our system by considering additional low-level music features (MPEG 7 low-level audio descriptors), incorporating improved clustering algorithms (immune clustering algorithms) and additional relevance feedback modules [26], and pursuing an extensive system performance evaluation.

References

[16] www.napster.com


