Y. Ioannidis et al. (eds.): EDBT 2006, LNCS 3896, pp. 1164–1167, 2006.

MUSCLE: Music Classification Engine with User Feedback

Stefan Brecheisen[1], Hans-Peter Kriegel[1], Peter Kunath[1], Alexey Pryakhin[1], and Florian Vorberger[2]

[1] Institute for Informatics, University of Munich
{brecheis,kriegel,kunath,pryakhin}@dbs.ifi.lmu.de
[2] Florian.Vorberger@detach.de

Abstract. Nowadays, powerful music compression tools and cheap mass storage devices have become widely available. This allows average consumers to transfer entire music collections from the distribution medium, such as CDs and DVDs, to their computer hard drive. To locate specific pieces of music, they are usually labeled with artist and title. Yet the user would benefit from a more intuitive organization based on music style to get an overview of the music collection. We have developed a novel tool called MUSCLE which fills this gap. While there exist approaches in the field of musical genre classification, none of them features a hierarchical classification in combination with interactive user feedback and a flexible multiple assignment of songs to classes. In this paper, we present MUSCLE, a tool which allows the user to organize large music collections in a genre taxonomy and to modify class assignments on the fly.

1 Introduction

The progress of computer hardware and software technology in recent years made it possible to manage large collections of digital music on an average desktop computer. Thus, modern computer systems are able to compress a piece of music to a few megabytes in very fast time. Easy to use software that automates this process is available. Often, this software stores meta information, such as artist, album or title, along with the audio file. However, the amount and quality of the available meta information in publicly accessible online databases, e.g. freedb.org, is often limited. This meta data is especially useful when searching for a specific piece of music in a large collection. To organize and structure a collection, additional information such as the genre would be very useful. Unfortunately, the genre information stored in online databases is often incorrect or does not meet the user's expectations.

In this demonstration paper, we present MUSCLE, a prototype of a powerful hierarchical genre classification tool for digitized audio. It is often problematic to assign a piece of music to exactly one class in a natural way. Genre assignment is a somewhat fuzzy concept and depends on the taste of the user. Therefore, MUSCLE allows multi-assignments of one song to several classes. The classification is based on feature vectors obtained from three acoustic realms namely *timbre*,

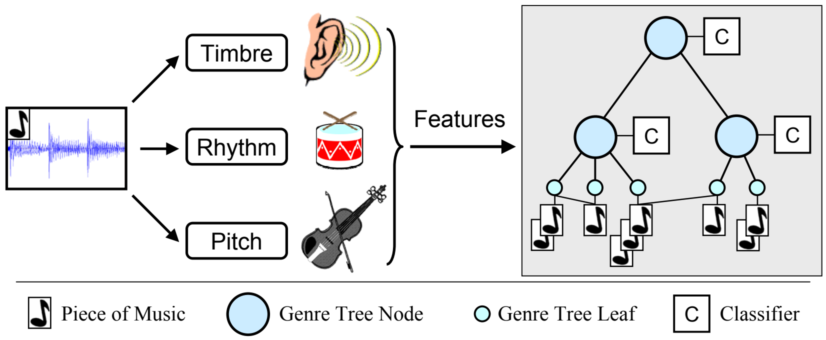


Fig. 1. Architecture of MUSCLE.

rhythm and *pitch*. Timbre features are derived from the frequency domain and were mainly developed for the purpose of speech-recognition. The extraction of the timbral texture is performed by computing the short time fourier transform. We use the Mel-frequency cepstral coefficients (MFCCs), spectral flux and spectral rolloff as trimbral representations [1]. Rythmic content features are useful for describing the beat frequency and beat strength of a piece of music. In our demo, we use features derived from beat histograms [1] as description of the rhythmic content. Pitch extraction tries to model the human perception by simulating the behavior of the cochlea. Similar to the rhythmic content features, we derive pitch features from pitch histograms which were generated by a multipitch analysis model [2]. To sum up, each song is described by multiple representations and multiple instances, i.e. there exists a set of feature vectors per representation.

The general idea of hierarchical classification is that a classifier located on an inner node solves only a small classification problem and therefore achieves more effective results more efficiently than a classifier that works on a large number of flat organized classes. There exist only a few approaches for automatic genre classification of audio data. In [3], music pieces are classified into either rock or classic using k-NN and MLP classifiers. An approach for hierarchical genre classification which does not support user feedback is presented in [1]. Zhang [4] proposes a method for a hierarchical genre classification which follows a fixed schema and where is only limited support for user-created genre folders. Moreover, all above mentioned hierarchical classification methods do not take full advantage of multi-instance and multi-represented music objects. In contrast, MUSCLE handles such rich object representations as well as an arbitrary genre hierarchy, deals with user feedback and supports multi-assignment of songs to classes.

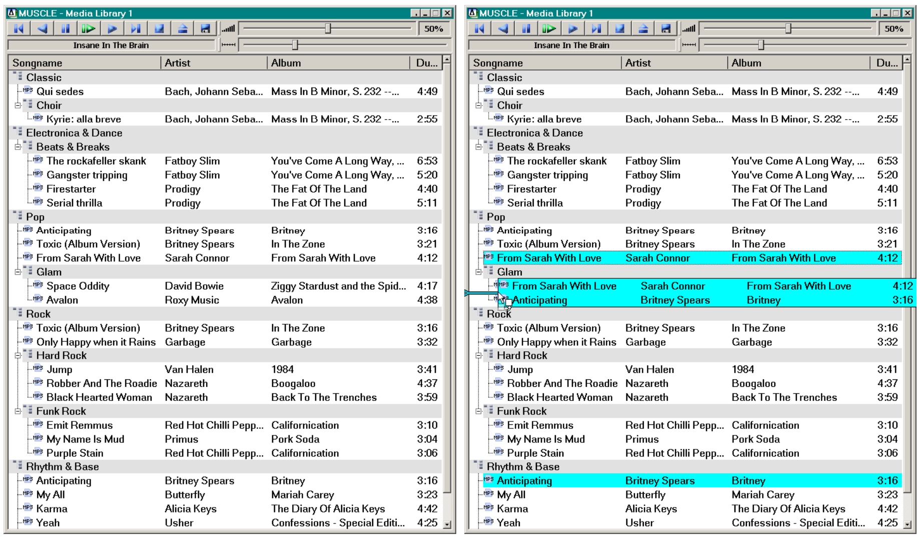
2 Theoretical Foundation

In this section, we briefly describe MUSCLE's approach for classifying music pieces into a genre taxonomy (cf. Fig. 1). Support vector machines (SVM) are used as classifiers which achieve superior classification accuracy in various application areas and have received much attention recently. By using kernel functions in combination with SVMs, any kind of data can be classified. Since a music piece is described by a set of feature vectors, we apply a set kernel function [5] for SVMs.

The hierarchical classification problem is handled by performing a two layer classification process (2LCP) on each inner node N of the genre taxonomy. This process distinguishes only descendent nodes of N as classes C_{single} and acts as a guidepost for the hierarchical classification. We train SVMs in the first layer of the 2LCP that distinguishes only single classes in each representation. Since standard SVMs are able to make only binary decisions we apply the so-called one-versus-one (OvO) approach in order to make a classification decision for more than two classes. We argue that for our application the OvO approach is best suitable because the voting vectors provided by this method are a meaningful intermediate description that is useful for solving the multi-assignment problem in the second layer of our 2LCP. In order to perform the multi-assignment we take advantage of the class properties in our application domain. We limit the possible class combinations to a subset $C_{combi} \subset 2^{C_{single}}$ because there exist several combinations that do not make sense, e.g. a piece of music belonging to the class 'classic' is very implausible to be also in the class 'hip-hop'. Thus, the classifier (SVM) in the second layer of the 2LPC uses an aggregation of the voting vectors from the first layer of the 2LPC as input to assign an object to a class $c \in C_{single} \cup C_{combi}$. The voting vectors provided by the first layer SVMs for each representation are aggregated by using a weighted linear combination. The weights in the combination are calculated by using a so called object adjusted weighting. The intuition behind the object adjusted weighting is that the object to be classified needs to have a sufficient distance from any of the other classes. For more details we refer to [6].

3 Practical Benefits

MUSCLE is implemented in C/C++ and runs on the Windows platform. Its hierarchical playlist acts as a jukebox. The installation archive of MUSCLE contains a default genre taxonomy including the necessary training data in the form of feature vectors for each song. This data is used in the demonstration. Using aggregated information such as feature vectors makes it possible to share the training data without having to distribute the underlying music data. Classes and training data in the genre taxonomy can be deleted, moved or added by the user. When the user commits the changes of the class hierarchy or of the corresponding training data, MUSCLE trains the affected classifiers. Note that usually only a small subset of the entire classifier hierarchy has to be trained because a modification at a node requires a partial adaptation of the node and all parent nodes only. It is also possible to start the training automatically after each modification or to run the training in the background. When the user is



(a) Multi-Assignment of Songs

(b) User Feedback

Fig. 2. MUSCLE User Interface

satisfied with the training setup, a folder to automatically classify all contained songs can be selected.

Fig. 2 illustrates MUSCLE's user interface. In the main window the playlist containing the classification result in form of a genre tree is displayed. An example for a multiple assignment of the song 'Anticipating' to the classes 'pop' and 'rhythm & base' can be seen in Fig. 2(a). In case the user wants to manually adjust the genre assignment of a song, entries can be re-arranged using drag & drop as shown in Fig. 2(b).

References

1. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Transactions on Speech and Audio Processing **10** (2002) 293–302
2. Tolonen, T., Karjalainen, M.: A computationally efficient multipitch analysis model. IEEE Transactions on Speech and Audio Processing **8** (2000) 708–716
3. Costa, C.H.L., Valle, J.D.J., Koerich, A.L.: Automatic classification of audio data. IEEE Transactions on Systems, Man, and Cybernetics **3** (2004) 562–567
4. Zhang, T.: Semi-automatic approach for music classification. In: Proc. of the SPIE: Conf. on Internet Multimedia Management Systems. Volume 5242. (2003) 81–91
5. Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.J.: Multi-instance kernels. In: Proc. of the 19th Int. Conf. on Machine Learning (ICML). (2002) 179–186
6. Kriegel, H.P., Kröger, P., Pryakhin, A., Schubert, M.: Using support vector machines for classifying large sets of multi-represented objects. In: Proc. SIAM Int. Conf. on Data Mining (SDM). (2004)