IEEE 7th Int. Conf. on Multimedia & Expo (ICME 2006), Toronto, Canada, pp. 1385-1388

HIERARCHICAL GENRE CLASSIFICATION FOR LARGE MUSIC COLLECTIONS

Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, Alexey Pryakhin

University of Munich, Institute for Informatics
{brecheis,kriegel,kunath,pryakhin}@dbs.ifi.lmu.de

ABSTRACT

The rapid progress in digital music distribution has lead to the creation of large collections of music. There is a need for content-based music classification methods to organize these collections automatically using a given genre taxonomy. To provide a versatile description of the music content, several kinds of features like rhythm, pitch or timbre characteristics are commonly used. Taking the highly dynamic nature of music into account, each of these features should be calculated up to several hundreds of times per second. Thus, a piece of music is represented by a complex object given by several large sets of feature vectors. In this paper, we propose a novel approach for the hierarchical classification of music pieces into a genre taxonomy. Our approach is able to handle multiple characteristics of music content and achieves a high classification accuracy efficiently, as shown in our experiments performed on a real world data set.

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. Often meta information, such as artist, album or title, is available 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 paper, a content-based hierarchical genre classification framework for digitized audio is presented as sketched in Figure 1. 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, our approach allows multi-assignment of one song to several classes. The classification is based on feature vectors obtained from three acoustic realms namely *timbre*, *rhythm* and *pitch*. Thus, each song is described by multiple representations, each of them containing a set of feature vectors, so called *multiple instances*.

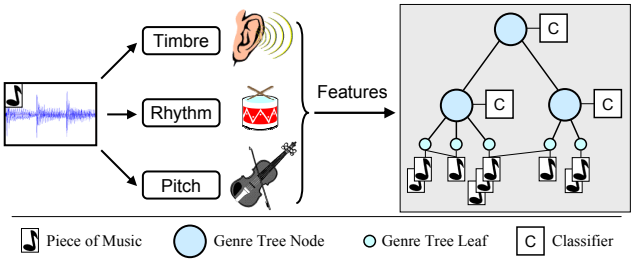

Fig. 1. Architecture of the proposed framework.

Our main contributions are: (1) a novel semi-supervised, hierarchical *instance reduction* (IR) technique which enables us to use only a small number of relevant features for each classifier. (2) An effective and efficient framework for *hierarchical genre classification* (HGC) of music pieces in a *multi-representation* (MR) and *multi-instance* (MI) setting. Let us note that our framework can also be used for *genre classification* (GC) in flat class systems.

2. RELATED WORK

Feature extraction. 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 timbral representations [1]. Rhythmic content features are useful for describing the beat frequency and beat strength of a piece of music. In our framework, we use features derived from beat histograms [1] as the 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].

Genre classification. The general idea of hierarchical classification is that a classifier located on an inner node of the genre tree 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. 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, the above mentioned hierarchical classification methods do not take full advantage of MI and MR music objects. In contrast, our approach handles such rich object representations as well as an arbitrary genre hierarchy, and supports multi-assignment of songs to classes.

Hierarchical Classification. The use of class hierarchies to improve large scale classification problems has predominantly been applied in text classification. Several approaches have been introduced picking up this idea. The authors of [5] investigated multiple representations of objects in the context of hierarchical classification and proposed a so called *object adjusted weighting* for linear combination of MR objects.

Support Vector Machines. In recent years, *support vector machines* (SVMs) [6] have received much attention offering superior performance in various applications. For example, [7] presents a fusion technique for multimodal objects. Basic SVMs distinguish between two classes by calculating the maximum margin hyperplane between the training examples of both given classes. To employ SVMs for distinguishing more than two classes, several approaches were introduced [8]. In order to handle sets of feature vectors in SVMs so called kernel functions were introduced [9]. A weakness of MI kernels is the need to calculate distances between all instances, i.e. $O(n^2)$ single distance calculations are required in order to compare two MI objects with n instances. Thus, MI kernels seem to be unsuitable for solving large scale classification problems in music collections.

Instance Reduction Techniques. As mentioned above, a piece of music is usually described by a set of feature vectors and is an MI object. The number of instances can vary from tens to hundreds per second, i.e. a song is represented by 10,000 to 50,000 feature vectors. In order to handle such MI objects two classes of IR techniques can be distinguished, namely higher-order and first order. Higher-order IR techniques use optimization algorithms on feature vectors. They describe an MI object as a mix of statistical distributions or cluster representatives. In [10], a higher-order instance technique is presented which is based on Gaussian distributions. The authors use methods such as Expectation Maximization for parameter estimation. The authors of [11] propose an IR approach that computes the optimal representatives by minimizing the Hausdorff distance between the original object and its representation. If the Euclidian metric is used as a distance function on the feature vectors, the k-means method can be applied for summarization of multimedia content [12]. In case of general metric spaces, the k-medoid method can be applied for summarization. A randomized first order IR technique, called signature, is proposed in [13]. A multimedia sequence in the database is described by selecting a number of its instances closest to a set of random vectors. The authors in [13] also propose a specialized distance function on the derived first order summarization vectors. Both first and higher-order techniques reduce the MI object to a small set of feature vectors. Thus, using the reduced representations of the MI object requires the application of kernel functions for SVMs. In context of large databases, the use of kernel functions seems impracticable for efficient classification.

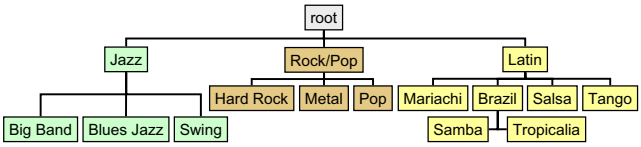

Fig. 2. An example genre hierarchy.

3. EFFICIENT HIERARCHICAL GENRE CLASSIFICATION

In this section, we describe our approach for classifying large collections of music pieces in a genre taxonomy (cf. Figure 2). Since a music piece is described by a set of feature vectors, we first describe a novel hierarchical semi-supervised technique for instance reduction. The reduced descriptions are used afterwards for hierarchical classification of music pieces with SVMs. Furthermore, we use object adjusted weighting in order to take advantage from multiple representations.

Hierarchical Instance Reduction. Let DB be a set of music objects. We argue that an MI object $X = \{x_1, \ldots, x_n\} \in DB$ can be described by a vector $X_{reduced}$ containing minimal distances to a given set of so called *support objects* $S = \{s_1, \ldots, s_m\}$ where $m \ll n$. Formally,

$$X_{reduced} = (\min_{x_i \in X} dist(x_i, s_1), \ldots, \min_{x_i \in X} dist(x_i, s_m)).$$

The set S can either be calculated by a random selection of m instances from DB, or it is possible to choose each $s_i \in S$ as a centroid of a clustering that can be calculated on a small sample of instances from DB. An example for the instance reduction is illustrated in Figure 3.

The number of elements in $X_{reduced}$ may still be too large for solving the classification problem efficiently. Thus, we propose to exploit the hierarchical organization of classes and to select only a small subset $S_N \subseteq S$ for each inner node N of the genre taxonomy. The elements of S_N should be selected so that the subclasses C_N of N can be distinguished in the best possible way. Therefore, the subset of support objects is individual for each inner node N.

To calculate S_N we suggest to apply a semi-supervised method based on the *information gain criterion*. Let $T(C_N)$ be a set of all training objects belonging to C_N. The domains $D(s_i)$ are discretized by using the method described in [14]. After discretization the information gain criterion for each attribute can by calculated by

$$InfoGain(s_i, T(C_N)) = H(T(C_N)) - \sum_{t \in T(C_N)} \frac{|t|}{|T(C_N)|} \cdot H(t),$$

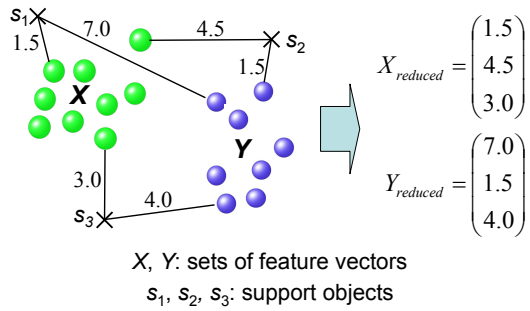

Fig. 3. Instance reduction with help of support objects.

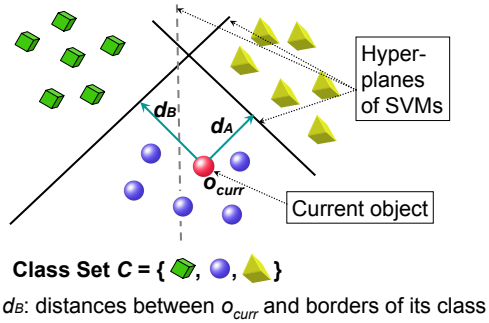

Fig. 4. Border distance based derivation of weights for a multi-represented object.

where $H(t)$ denotes the entropy. Finally, S_N is calculated as follows: $S_N = \{s_j \in S \,|\, |S_N| = k \wedge \forall s_j \in S_N \forall a \in S : InfoGain(a, T(C_N)) \leq InfoGain(s_j, T(C_N))\}$. After that, S_N is used for training and classification on the node N.

Hierarchical Genre Classification by Using Multiple Representations. A *two layer classification process* (2LCP) handles the hierarchical classification problem on each inner node N of the genre taxonomy. This process acts as a guidepost for the hierarchical classification. We train SVMs in the first layer of the 2LCP that distinguishes only single classes C_{single} in each representation. Since standard SVMs are able to make only binary decisions we apply the so-called one-versus-one (OvO) approach (cf. Figure 4) 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 Φ_i 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 'salsa' is very implausible to be also in the class 'metal'. For this purpose, we only take those $c \in 2^{C_{single}}$ into account, which occur in the training set.

The SVM classifier in the second layer of the 2LPC uses an aggregation of the voting vectors Φ_i from the first layer of the 2LPC as input to assign an object to a class $c \in C_N = C_{single} \cup C_{combi}$. The second task that is handled by the classifier in the second layer is the aggregation of multiple representations. The voting vectors $\Phi_1, \ldots, \Phi_k$ provided by the first layer SVMs for each representation $R_1, \ldots, R_k \in R$ are aggregated by using a weighted linear combination $V = \sum_{i=1}^{k} \omega_i \Phi_i$. Then V is used as the input for the classifier in the second layer. The weights ω_i in the combination are calculated by using object adjusted weighting. The intuition behind the object adjusted weighting is that the current object o_{curr} used in training or to be classified needs to have a sufficient distance from any of the other classes. More formally, let c_j be the class of o_{curr} determined by majority vote in Φ_i, then $\omega_i = \min_{c_i \in C_{single} \wedge c_i \neq c_j} dist(o_{curr}, HyperPlane(c_j, c_i))$, where $HyperPlane(c_j, c_i)$ denotes the maximum margin hyperplane separating the classes c_j and c_i. Figure 4 depicts an example of weight calculation where the weight ω should be set to d_A.

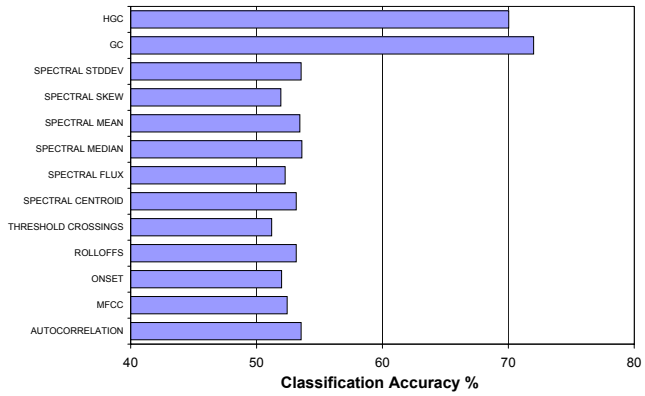

Fig. 5. Accuracy for classification on single- and multi-representations.

4. EXPERIMENTAL EVALUATION

We implemented our approach in Java 1.5 and performed all experiments on a Pentium IV workstation equipped with 2 GByte main memory. The genre hierarchy depicted in Figure 2 was used in all following experiments. A music collection consisting of almost 500 songs was the basis for the classification experiments, which results in approximately 30 songs per class. Depending on the representation, we extracted 30 to 200 features per second. We performed 10-fold cross-validation for evaluating the classification accuracy. In the following, we present the results of our experiments with particular emphasis to efficiency and effectiveness.

Effectiveness. In the first experiment, we compared the quality of GC on multiple, and HGC on single and multiple representations. Figure 5 depicts the experimental results. When working with multiple representations, our HGC approach (70.03%) achieves higher classification accuracy than using a single representation only. Furthermore, the classifi-

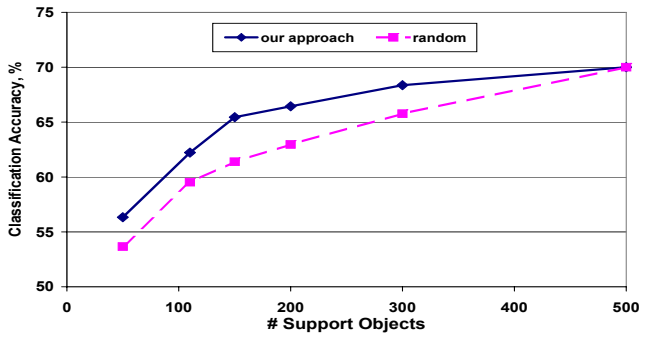

Fig. 6. Accuracy for classification on single- and multi-representations.

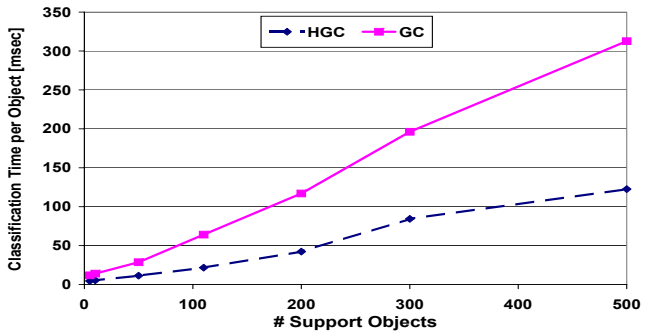

Fig. 7. Classification time per object.

cation accuracy of HGC is comparable to that of the flat GC approach (72.01%).

In the next experiment, we investigated how the classification accuracy of our approach is influenced by the number and the choice of the support objects. For choosing S_N, we either randomly picked the support objects or applied our strategy described in Section 3. The experimental results are depicted in Figure 6 and show that our approach always outperforms the random selection. For both approaches, the accuracy increases with an increasing number of support objects. However, especially for a low number of support objects, the random approach achieves a lower accuracy compared to our method. For a high number of support objects, both approaches yield a similar classification accuracy.

Efficiency. In a last experiment, we examined the runtime performance of GC and HCG for a varying number of support objects. As depicted in Figure 7, the runtime increases with an increasing number of support objects. The higher the number of support objects, the larger the runtime difference. Altogether, our approach achieves a good trade-off between the quality of the result and the required runtime when using 300 support objects.

5. CONCLUSIONS

In this paper, we introduced a framework for hierarchical music classification using multiple representations consisting of multiple instances. We showed that our hierarchical classification can compete with a flat class system in terms of effectiveness and greatly surpasses it in terms of efficiency. An implementation of our framework has been demonstrated recently [15]. In the future, we plan to extend the framework to handle video data.

6. REFERENCES

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

[2] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, 2000.

[3] C. H. L. Costa, J. D. Jr. Valle, and A. L. Koerich, "Automatic classification of audio data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 6, 2004.

[4] T. Zhang, "Semi-automatic approach for music classification," in *Proc. SPIE Conf. on Internet Multimedia Management Systems*, 2003.

[5] H.-P. Kriegel, P. Kröger, A. Pryakhin, and M. Schubert, "Using support vector machines for classifying large sets of multi-represented objects," in *Proc. SIAM Int. Conf. on Data Mining*, 2004.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, 1995.

[7] Y. Wu, C.-Y. Lin, E. Chang, and J. R. Smith, "Multimodal information fusion for video concept detection," in *Proc. ICIP*, 2004.

[8] J. Platt, N. Cristianini, and J. Shawe-Taylor, ""Large Margin DAGs for Multiclass Classification"," in *Proc. NIPS*, 1999.

[9] T. Gärtner, P. A. Flach, A. Kowalczyk, and A. J. Smola, "Multi-instance kernels," in *Proc. ICML*, 2002.

[10] H. Greenspan, J. Goldberger, and A. Mayer, "A probabilistic framework for spatio-temporal video representation & indexing," in *Proc. ECCV*, 2002.

[11] H. S. Chang, S. Sull, and S. U. Lee, "Efficient video indexing scheme for content-based retrieval," in *IEEE Transactions on Circuits and Systems for Video Technology*, 1999, vol. 9.

[12] Y. Zhuang, Y. Rui, T. S. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering.," in *Proc. ICIP*, 1998.

[13] S.S. Cheung and A. Zakhor, "Efficient video similarity measurement with video signature," in *Proc. ICIP*, 2002.

[14] U. M. Fayyad and K. B. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Machine Learning*, vol. 8, 1992.

[15] S. Brecheisen, H.-P. Kriegel, P. Kunath, A. Pryakhin, and F. Vorberger, "Muscle: Music classification engine with user feedback," in *Proc. EDBT*, 2006.