2012 IEEE International Conference on Multimedia and Expo Workshops

A Novel Automatic Hierachical Approach to Music Genre Classification

Hasitha B. Ariyaratne, Dengsheng Zhang
Gippsland School of Information Technology
Monash University
Churchill, Australia
e-mail: {hasitha.ariyaratne, dengsheng.zhang}@monash.edu

Abstract— **Automatic music genre classification is an important component in Music Information Retrieval (MIR). It has gained lot of attention lately due to the rapid growth in the use of digital music. Past work in this area has already produced a number of audio features and classification techniques; however, genre classification still remains an unsolved problem. In this paper we explore a hybrid unsupervised/supervised top-down hierarchical classification approach. Most existing work on hierarchical music genre classification relies on human built trees and taxonomies; however these hierarchies may not always translate well into machine classification problems. Therefore, we explore an automatic approach to construct a classification tree through subspace cluster analysis. Experimental results validate the tree building algorithm and provide a new research direction for automatic genre classification. We also addressed the issue of scarcity in publicly available music datasets, by introducing a new dataset containing genre, artist and album labels.**

Keywords- Hierarchical music genre classification; music dataset

I. Introduction

Classification is a key problem in music information retrieval and management, in a broad sense it deals with identifying and assigning labels to audio clips. This task can range from labeling music (according to genre, artist, mood etc.), detecting events in an audio stream to voice recognition. In this paper we primarily focus on music genre classification because genre is the most widely used descriptor in organizing and searching large music collections [1]. But the ideas put forward here have wide applicability in many different areas. Currently there are a number of publications related to MIR and music classification; some of the notable and highly cited works include: Tzanetakis et. al.[2] and McKay et. al.[3]. Most of the previous work in this area focuses on developing new features and classification techniques, a recent survey has listed over 70 highly utilized audio features from over 200 publications [4], while [5] has presented a variety of machine learning algorithms used in music classification. Despite having such a large number of techniques, automatic genre classification still remains an unsolved problem. Some argue that existing audio features and classifiers have reached a "glass ceiling" [6], hence most new features and classifiers only show marginal improvement. Further investigation of existing literature reveals that majority of the methods use a single classifier trained with a collection of different audio features. We also found that relatively little work has been done with regard to feature combination and classifier fusion, especially exploiting hierarchical structures. This led us to investigate alternative ways to improve classification through better structuring and combining of already existing array of features and classifiers. In this paper we put forward a hierarchical genre classification method where a classification tree utilizing supervised learning algorithms is constructed using an unsupervised learning algorithm. We also suggest node specific feature and classifier selection as a further improvement.

Another important but less addressed issue in MIR research is the scarcity of publicly accessible large databases for benchmarking and testing. In most MIR publications, growth of digital music and the need of efficient music management techniques are cited as strong motivations for new research. However, despite rapid growth in the number of digital music files, MIR research community still has very limited access to sufficiently large datasets due to a number of economical, technical and legal issues. This led us to construct our own dataset. And we believe that the research community can benefit from having access to this dataset as it contains a sufficiently large number of audio samples belonging to many different genres, artists and albums.

The rest of this paper is organized as follows: Section 2 explains the intuition behind the hierarchical approach to genre classification, followed by an overview of the proposed system. Section 4 presents the findings of our initial experiments, and section 5 concludes the paper by mentioning some of the future research directions.

II. Hierarchical Classification of Music Genres

The term music genre lacks a proper definition despite its huge popularity, moreover, the boundaries and relationships between them tend to be fuzzy and vague[1]. Use of taxonomies is one way to bring structure and organization to genres. However, there are no standard ways of generating taxonomies; different organizations use their own taxonomies. Pachet et.al[6] further discusses such issues and ambiguities found in large commercially available music libraries. Nevertheless, genre classification is still an important research topic, mainly because of its popularity. Music industry uses genre as the main organizing criteria while psychological experiments have shown that humans are very good at genre identification, furthermore, consumer studies show that listeners are more likely to browse and search for new music by genre than any other criterion (i.e, artist, album, recommendations, popularity) [1].

Music genre classification comprises of two stages: audio feature extraction and classification. Audio features are the fundamental building blocks of any MIR system as they translate raw audio content into statistically meaningful representations which are later used for analyzing the audio content. However, they are designed only to capture specific qualities of sound, therefore are limited in their generalizability. For example, features like Zero Crossing Rate (ZCR), and Linear predictive coefficients (LPC) are good at discriminating instrumental music (i.e Classical) from vocal music (i.e Country) due to their ability in capturing certain characteristics of human voice. However, they are not good at discriminating between pure instrumental genres (i.e classical and jazz). The most common way to solve this problem is to combine

978-0-7695-4729-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICMEW.2012.104

different features. But, fusing features should be done carefully, since all features do not contribute equally to the classification task. For example, a useful feature identifying a certain genre may act as noise for another. Moreover, large feature vectors can lead to high computational complexities and complications (also known as the "curse of dimensionality").

One way to utilize a large number of features but avoid the problems stated above is to adopt a step-by-step (divide and conquer) hierarchical classification approach. In which the problem is divided into several smaller and simpler classification tasks. Some of the early works in hierarchical genre classification [7] relied on pre-defined taxonomies and manually selected audio features based on their application domains (i.e Speech specific and music specific). Later works have proposed automatic feature selection techniques [8], and the use of multiple classifiers at each level of the hierarchy [9]. DeCoro el.al [10] have proposed an aggregation method to combine multiple binary classifiers into a hierarchical classifier with the help of a Bayesian network. A different hierarchical approach was proposed by Zhang et.al[11] where they have manually constructed a unique hierarchy based on heuristics. They start by dividing genres in to pure instrumental and singing, and then further separate each group to sub groups (i.e Instrumental into symphony and solo) based on distinct auditory characteristics till all classes are uniquely identified. This method appears very intuitive; but, constructing this type of a tree requires expert knowledge and a good understanding of fundamental auditory characteristics of each genre. Despite the different approaches, they all use a manually generated hierarchy or genre taxonomy. But, finding a well-defined taxonomy or manually creating one may not always be trivial or feasible.

One important question to ask when using a pre-defined musicological taxonomy is "how well does the taxonomy separates genres"? For example consider the taxonomy in Figure 1, where some of the divisions are based on musicological concepts, i.e : divisions such as "classical" (pre-1900s) vs. "non-classical" (post 1900s); or Chamber music (performed in a small room) vs. Orchestral (performed in front of a larger audience). However, transforming these divisions to a machine understandable form using standard audio features may not always succeed as the differences are more conceptual than auditory.

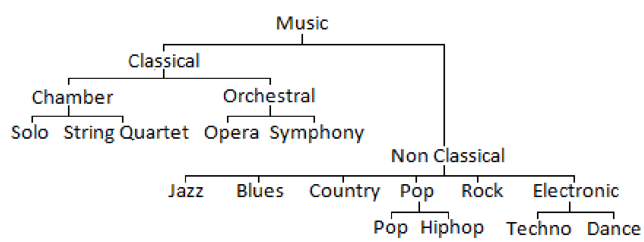

Figure 1. A taxonomy based on musicological concepts

A. *Hierarchical Classification with Decision Trees*

Decision Tree (DT) is a machine learning algorithm which tries to solve a classification task by adopting a divide and conquer approach. A DT iteratively partitions data in a top-down manner where each split tries to separate the data points as "pure" as possible. This is achieved by selecting either a single attribute (univariate) or multiple attributes (multivariate) for each node that best discriminates data. Suitability of attributes are decided by a measurement such as "Information gain" or "Gini Index"[12]. Most popular DTs are univariate because they are simpler to implement and easy to learn. However, as the number of attributes increase, the size of the tree and the complexity of the problem are increased. Moreover, picking a single attribute at a time discards correlations between attributes [13], which otherwise could greatly improve classification accuracy. Due to these limitations, Multivariate DTs with linear discriminative analysis (LDA) were suggested [13], where a linear discriminant function transforms multi-dimensional data into one dimension such that the distance between the two class centroids are maximized[13]. However, linear discriminative analysis may not work well if the classes do not have a linearly separable hyper plane between them.

In our work we propose a "multivariate decision tree like" tree building approach which uses subspace clustering for class separation. Furthermore we employ a set of commonly used feature selection methods for attribute selection. Specific details of the algorithm are discussed in the following section.

III. System Design

This section presents the details of the proposed system with regard to feature extraction, tree construction, attribute selection and classification.

A. *Feature extraction*

Genre classification is aimed at providing efficient and effective means to search, discover and manage of music. Currently, most end user music consumption devices (i.e: MP3 players, stereo systems, mobile phones) do not possess high computing power. Therefore focusing on less complex algorithms for feature extraction is still worthwhile consideration. On the other hand, classifier training and classification can be done at more powerful remote servers, therefore. We decided to use a set of relatively simple, computationally inexpensive widely used low level features for our experiments. We chose a total of 13 audio features comprising of six MPEG7 features and seven other popular features. The statistical properties of each feature's mean, covariance, and their numerical partial derivatives (the differences between successive elements of the feature vector) constituted a 183 dimensional feature vector. The complete list of features used in our experiments are listed in Table I.

TABLE I. AUDIO FEATURES USED FOR THE EXPERIMENT

Feature Name	No.of features
1. Total Energy	4
2. Fundamental Frequency	4
3. Loudness Sensation	32
4. Integral Loudness	4
5. Audio Spectral Centroid	4
6. Spectral Rolloff	4
7. Audio Spectrum Spread	4
8. Audio Spectrum Flatness	16
9. Audio auto correlation	13
10. Log Attack Time	1
11. Temporal Centroid	1
12. Zero Crossing Rate	4
13. Mel Frequency Cepstral Coefficients	96
Total	183

Features 1,10,11 and 12 are time domain features while rest are extracted from the frequency domain. MFCC is a widely used feature in many different areas of MIR. Features 2 and 5 through 8 provide various statistical measurements related to the frequency spectrum of a sound. Features 3 and 4 capture the human perception of loudness while auto correlation features can be used to analyze reoccurring patterns (i.e beats, tempo) in a signal. More details and mathematical representations of these features can be found in [15].

B. Automatic Generation of the Hierarchy

As noted earlier, previous work on hierarchical music genre classification have relied on human generated taxonomies and hierarchies. However, musicological taxonomies and manually generated hierarchies may not adhere to the natural clusters found in the data, especially if the features used for classification are low level. Moreover, manually finding the best features to use at any given node as suggested in [7,11] is not a trivial task. For these reasons we propose a fully automatic approach to generating the classification hierarchy.

We draw our intuition from decision trees; however we use a different approach to tree building where we use an unsupervised clustering technique to aid the node splitting decision instead of information gain.

There are number of unsupervised clustering techniques i.e K-Means, Expectation Maximization (EM), DBSCAN etc. however most of these algorithms rely on distance measurements over the entire feature space. Hence, they do not scale well with large number of features. Subspace methods on the other hand are specifically designed for clustering high dimensional datasets, where only the relevant subspaces of the entire feature space are considered [14]. Thus they scale and perform better than conventional approaches with high dimensional data.

C. Subspace Cluster Analysis

Subspace clustering is an extension of traditional clustering algorithms. They seek to find clusters in different subspaces within the entire feature space [14]. These techniques can handle high dimensional feature spaces effectively since they remove irrelevant, noisy and redundant dimensions by means of feature selection. There are two main approaches to subspace clustering: top-down and bottom up. Bottom-up methods adopt an APRIORI type technique used in data mining. These methods start by building clusters in low dimensional spaces using few features at first, then combine them to form clusters in high dimensional spaces with the assumption of downward closure property of density. The nature of bottom-up techniques generally results in overlapping clusters.

In contrast, top-down approaches start by finding approximate clusters in the entire feature space with equally weighted dimensions. Then each dimension is iteratively assigned an updated weight with regard to each cluster. These weights are then used to compute new clusters in the next iteration. At the end of the algorithm, highest weighted dimensions are selected to represent each cluster. Top down approaches generally do not create overlapping clusters. Another difference is that they require the number of potential clusters as an input parameter. Interested readers are directed to [14] for more details about both techniques.

Among different subspace clustering methods, we chose PROjected CLUstering (PROCLUS) algorithm [16] which uses K-Mediods method for distance measurement. We chose the PROCLUS algorithm because we could control the number of clusters hence the breadth of the tree. For example, setting the number of clusters K=2, will result in a binary tree. The algorithm starts by initializing a random set of mediods then iteratively improves the quality of the mediods through hill climbing, by removing poor mediods. Complete PROCLUS algorithm can be found in [16]. It takes the number of clusters and average dimensions as input parameters, then computes the clusters and most relevant dimensions for each cluster.

In our tree building process the PROCLUS algorithm is executed in a top down manner. We start with all the classes at the root node (the initial configuration). Output clusters are then considered as child nodes on which the algorithm is executed repeatedly till we obtain clusters containing only single class clusters. For performance reasons only a subset of data instances were used for the tree building process. The procedure is illustrated as follows:

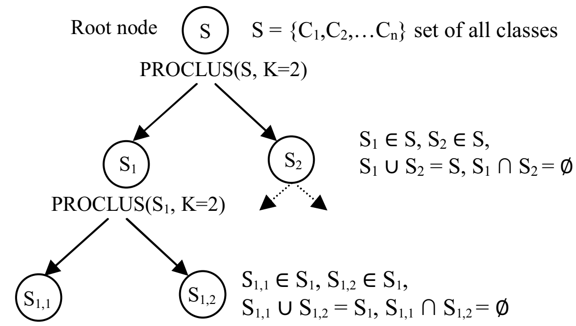

Figure 2. Tree building process.

One of the limitations in PROCLUS algorithm is the need of average cluster dimensionality as an input parameter. The average cluster dimensionality is the number of optimal dimensions in which data points are most correlated. However there are no intuitive guidelines to derive this value. In order to overcome this limitation we first obtained a set of possible cluster configurations by varying the average cluster dimensions. These configurations were then ranked and filtered according to the following criteria to find the best naturally occurring cluster configuration:

- Number of total clustered instances (TCI)
- Average cluster membership of each class (i.e probability of a particular class C_i belonging to a given cluster L_j) :

$$P(C_i \mid L_j) = \frac{\text{Instances of } C_i \text{ belonging to } L_j}{\text{Total instances of } C_i}$$

- Number of classes with a high cluster membership: In certain cluster configurations certain classes are not strongly identified with any of the clusters, yielding poor cluster membership values. We consider only those which havr a strong cluster membership value above a given threshold τ

$$\text{Consider } C_i \text{ iff } \forall L_j \text{ MAX}(P(C_i \mid L_j)) \geq \tau$$

- Minimum number of classes belonging to each cluster (changing this parameter helps control the depth of the tree.)

$$\text{ClassCountOf}(L_i) \geq \lambda \ \forall L_j$$

Finally the cluster configuration with the highest probability among the top ranked results is chosen to split the tree node. To illustrate the algorithm further, consider a binary split (k=2) at a node with 6 genres containing 100 clips each. Nine possible cluster configurations generated by the PROCLUS algorithm are listed as follows:

1. {(70/20)(65/25)(30/60)(25/70)(40/55)(60/10)} TCI=530
2. {(80/10)(75/20)(12/78)(21/64)(26/69)(70/10)} TCI=535
3. {(74/26)(80/15)(11/84)(30/65)(25/70)(85/05)} TCI=570
4. {(72/18)(67/24)(26/55)(55/40)(35/58)(65/30)} TCI=545
5. {(66/20)(73/14)(35/45)(61/28)(21/62)(70/20)} TCI=515

6.{(55/10)(42/05)(23/41)(30/15)(52/20)(60/20)} TCI=373
7.{(70/05)(80/10)(95/02)(88/06)(92/02)(78/10)} TCI=538
8.{(48/40)(45/43)(35/60)(38/48)(30/59)(46/40)} TCI=532
9.{(80/10)(45/41)(40/49)(08/90)(39/48)(79/15)} TCI=544

Note: numbers inside regular brackets indicate how clips of each genre are divided into the two clusters. E.g.: first (70/20) in cfg 1 denotes 70 instances of genre 1 belongs to cluster 1 while 20 to cluster 2, this also means 10 clips from genre 1 are unclustered. Number of Total Clustered Instances for all 6 genres is listed as the TCI value.

The first parameter for the selection criterion is TCI, we can use a threshold to eliminate configurations that do not properly cluster the genres, such as cfg 6 where the total clustered instances are very low. The second criteria helps remove configurations with poor memberships (where majority of the genres don't belong to either of the two clusters with a strong confidence) (i.e cfg.8), these configurations are not very helpful in identifying the naturally occurring cluster formations therefore are removed. The third criterion considers the number of total genres with a good cluster membership. This is different from the previous criterion where only average membership is considered for all genres. For example, cfg. 9 satisfy both previous conditions. However, genres 2,3 and 5 have poor memberships, therefore does not cluster some of the classes properly. Combination of the criteria 2 and 3 helps us to maximize genre specific membership values while maintaining a high average cluster membership. Final criterion is the total number of classes identified for each cluster. For example in cfg 7 all classes are identified with cluster 1, which is again a poor division, this type of a highly biased clustering is usually a result of limited or noisy feature selection, therefore should be discarded. Once poor configurations are eliminated, support for each configuration is calculated with majority voting. E.g cfgs 1,2 and 3 splits the genres as {1,2,6} {3,4,5}, meaning genres 1,2 and 6 belongs to one cluster and rest to the other. Similarly configurations 4, and 5 support a slightly different split : {1,2,4,6} {3,5}. But we choose the first configuration since it has more votes. This way we can identify the most likely natural cluster formations in the data.

D. Attribute Selection

Attribute selection is an important preprocessing step because it helps reduce the number of features used in training a classifier. This can help improve both classification accuracy and performance due to removal of irrelevant and noisy features. There are number of ways to carry out attribute selection, Principle Component Analysis (PCA) is a common choice; techniques like Linear Discriminant analysis and even subspace analysis methods can also be used. A number of commonly used attribute selection methods used in music genre classification are presented in [17]. Even though we used subspace analysis for our tree building process we did not use the same method for attribute selection as a pre-processing step. Subspace clustering produces different attribute sets for each of the generated clusters. This raises few questions when trying to use it as an attribute selection method.

- At a given node, if we are use attribute sets generated for all the sub clusters at that node, then how can they be combined?
- At a given node, if we are to use the attribute set generated at the parent node, then we have to investigate whether or not the attributes chosen as a result of a different classification problem at the parent node are suitable for a new classification problem at the current node.

For these issues we opted to use [17] as a reference for selecting the best feature selection technique for each classifier.

E. Classifier selection

One advantage of having a configurable classification tree is the ability to tune each node with its own classifier and feature set. We chose a set of 6 widely used classifiers for evaluating the performance of our classification tree. Following section presents the experiment setup and our findings.

IV. EXPERIMENTS

Before presenting the details of the experiment, we will briefly look at the newly constructed dataset named HBA Music Collection.

A. New dataset : HBA Music Collection

Despite the availability of large commercial music databases, the MIR research community only has access to a handful of small datasets.. This is largely due to the stringent copyright laws, Storage space limitations, transmission bandwidth constraints and prohibitively high cost in purchasing a large number of audio clips. The following table summarizes some of the currently available popular datasets:

TABLE II. CURRENTLY AVAILABLE MUSIC DATASETS

Name	Number of clips	Audio data Available?
RWC	465	YES
CAL500	502	NO
GTZAN	1000	YES
Audio Benchmarking	1886	YES
USPOP	8753	NO
Magnatagatune	25,863(5405)**	YES
Million Song D.Set	1,000,000	NO

** contains 5405 unique clips, others are different samples of the same clip.

As seen in table II, most large datasets do not provide access to raw audio data; instead they provide a set of pre-computed audio features. This severely limits their usability for developing and testing new features. Given the current state of publicly available datasets, we decided to construct our own dataset as described below.

TABLE III. HBA MUSIC COLLECTION

No. of Genres	15
Genres	Blues, Classical, Country, Disco, Hiphop, Indian, Jazz, Metal, Opera, Pop, Reggae, Rock, Salsa, Techno, Ambient
No. of files	7500 (500 per genre)
Format	22,000KHz, 64kbps MP3
Size	2.6GB

We constructed the dataset by downloading preview clips from a number of online music stores. Tags of each clip (Genre, Artist and Album) were all taken from the respective online sites as provided under album description. We only chose clips that had clear and complete genre labels to make the dataset pure and consistent.

Distribution and use of small clips (typically shorter than 30secons) have special provisions for Educational and noncommercial research. Use of these clips are not considered an infringement of copyright laws under the fair use policies defined in the Australian copyright law contained in the Copyright Act

1968(Cth); Title 17 of the U.S copyright law and the Section 29 of the Copyright Designs and Patents Act 1988 of U.K. However we strongly suggest anyone wanting to gain access to potentially copyrighted material refer the relevant copyright laws of their own countries. [18]. Interested parties can contact the authors for further information on how to obtain a copy of the dataset.

B. Experimental Setup

We used the WEKA machine learning platform as the test bed for conducting all our experiments. We selected the following 6 classifiers and attribute selection methods to evaluate the new hierarchical classification approach:

- K-Nearest Neighbor (IBk)
    - Chi-square Feature Evaluation
- Decision Tree (J48)
    - Gain Ratio Feature Evaluation
- Logistic Regression (LOG)
    - Gain Ratio Feature Evaluation
- Lib SVM (LSVM: Support Vector Machine) with a Radial Basis kernel
    - Chi-square Feature Evaluation
- Multi-Layer Perceptron (MLP : Neural Network)
    - Correlation-based Feature Selection
- Sequential Minimal Optimization (SMO: Support Vector Machine) with a polynomial kernel
    - Chi-square Feature Evaluation

Further details on the attribute selection methods can be found in [17]. We conducted the experiments on the newly constructed HBA Music collection and on the popular GTZAN genre dataset.

C. Experimental results

Figure 3 and 4 show the generated class hierarchies and their respective class assignments. While Tables IV and V list the classification accuracies obtained with different classifier configurations under both Flat (single classifier) and hierarchical configurations. The column labeled "Hierarchical" lists the overall classification accuracy obtained when particular classifier was used at every node of the tree. The last row "Combined (Best)" gives the overall accuracy when best classifier is chosen at every node.

TABLE IV. CLASSIFICATION RESULTS FOR THE HBA DATASET

Classifier	Flat (%)	Hierarchical(%)
IBK	66.5	**67.2**
J48	60.1	**61.7**
LOG	66.5	**67.2**
LSVM	66.1	**66.1**
MLP	75.1	**77.2**
SMO	77.2	**78.2**
Combined (Best)		**78.6**

TABLE V. CLASSIFICATION RESULTS FOR THE GTZAN DATASET

Classifier	Flat(%)	Hierarchical(%)
IBK	57.4	**58.6**
J48	50.5	**53.3**
LOG	57.4	**58.6**
LSVM	60.4	**62.1**
MLP	70.6	**71.2**
SMO	70.9	**71.3**
Combined (Best)		**72.9**

Finally Table VI and VII show the individual genre level accuracy of the best classifier for both datasets under hierarchical and flat classification. From Tables IV and V, we can observe that the hierarchical approach yields about 1% improvement across all the classifiers. This provides a promising perspective for the tree building approach.

The tree structures depicted in Figure 3 and 4 stops splitting the tree at levels 2 and 3. Because, while running these experiments we found out that after a certain depth, the classification accuracy drops below that of flat classifier.

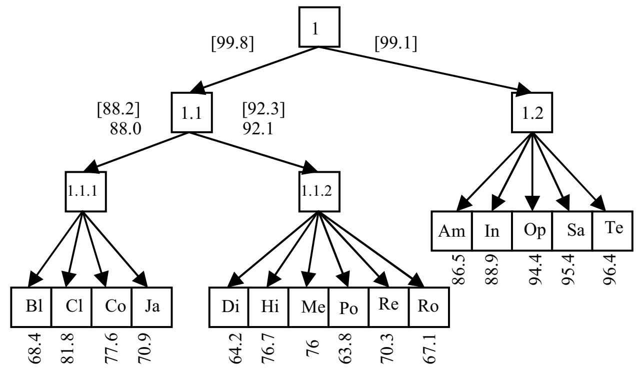

Figure 3. Class Hierarchy for HBA Music Collection [classification accuracies are in % values]

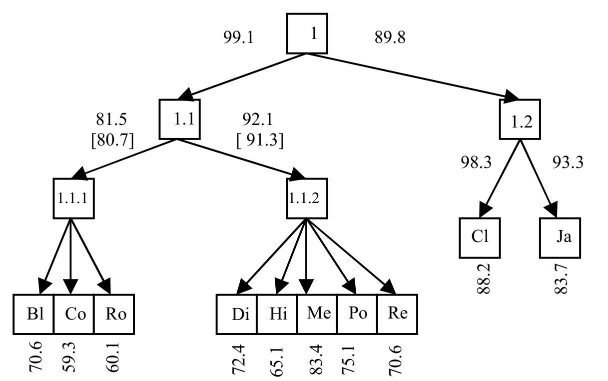

Figure 4. Class Hierarchy for GTZAN Dataset [classification accuracies are in % values]

The reason for this is downward error propagation; an inherent limitation of top down hierarchical approaches. Since each test instance has to go through multiple classifiers starting from the top, the chance of misclassification increases as the depth of the tree increases.

This is also evident in the results of the leaf level classification accuracy as noted in Figures 3 and 4. Despite having an almost 100% accuracy at the root level, node level accuracies (denoted with in square brackets) decrease as the depth increase. Due to space limitations we have only listed the node level accuracies of the best classifier. But all classifiers show a similar trend of decreasing performance down the tree.

TABLE VI. CLASSIFICATION RESULTS FOR THE GTZAN DATASET IN % VALUES

Clf.	Blu	Cla	Con	Jaz	Dis	Hip	Met	Pop	Reg	Roc	Amb	Ind	Ope	Sal	Tech	Tot
SMO	64.2	88.6	78.6	66.6	65.6	76.6	75	59.2	70.4	59	84.6	85.2	93.4	94.8	95.8	77.1
Com	68.4	81.8	77.6	70.9	64.2	76.7	76	63.8	70.3	67.1	86.5	88.9	94.4	95.4	96.4	78.6

TABLE VII CLASSIFICATION RESULTS FOR THE GTZAN DATASET IN % VALUES

Clf.	Blues	Classical	Country	Jazz	Disco	Hiphop	Metal	Pop	Reggae	Rock	Total
SMO	75	86	58	76	66	71	82	70	67	58	70.9
Com	70.6	88.2	59.3	83.7	72.4	65.1	83.4	75.1	70.6	60.1	72.9

However, the fact that we obtain very high classification accuracies at each node (not considering the error propagation) attests to the effectiveness of the subspace clustering based node splitting algorithm, and we believe it warrants further investigation.

V. CONCLUSION

Even though there are number of hierarchical genre classification approaches in the literature, they are almost always dependent on human generated taxonomies or hierarchies. However, genre taxonomies are not always consistent; and manually generating classification hierarchies is not a trivial task either. In this paper we proposed a hybrid unsupervised/supervised approach to hierarchical music genre classification where the tree building and feature selection are fully automated. Implementation of the tree building algorithm utilizing a subspace clustering method to find the natural clusters in the data space shows potential as validated by the experiment results. So far we have only considered a binary split in the tree, however multiple splits are possible, and yet to be fully explored. Another area worth investigating is classifier feature selection. Potential research directions include investigating the possibility of incorporating subspace feature reduction methods.

We also introduced a sufficiently large music dataset with Genre, Artist and Album labels which we believe could be of benefit to the MIR research community.

REFERENCES

[1] C. McKay, I. Fujinaga, “Musical Genre Classification: Is it worth pursuing and how can it be improved?”, Proc. 7th International Conference on Music Information Retrieval (ISMIR 2006)

[2] G. Tzanetakis, P. Cook, “Musical Genre Classification of Audio Signals,” IEEE Transactions on Speech and Audio Processing, vol. 10.

[3] C. McKay, I. Fujinaga, “Automatic Genre Classification Using Large High-Level Musical feature Sets”, Proc. International Conference on Music Information Retrieval, 2004.

[4] D. Mitrovic, M. Zeppelzauer and C. Breiteneder, “Features for Content-Based Audio Retrieval”, Advances in computers, Vol 78, 2010.

[5] N. Scaringella, G. Zoia and D. Mlynek, "Automatic genre classification of music content: a survey", IEEE Signal Processing Magazine, Vol 23, 2006.

[6] F. Pachet, D. Cazaly, “A taxonomy of musical genres”, Proc. Content-Based Multimedia Information Access, Paris, France, 2000.

[7] G. Tzanetakis, G. Essl and P. Cook,"Automatic Musical Genre Classification of Audio Signals", Proc. International. Symposium on Music Information Retrieval (ISMIR), 2001.

[8] J. Burred,A. Lerch, “A hierarchical approach to automatic musical genre classification”, Proc. International conference in Digital Audio Effects, 2003

[9] C. Silla,A.Freitas,”Novel top-down approaches for hierarchical classification and their application to automatic music genre classification”, Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC), 2009.

[10] C. DeCoro, Z. Barutcuoglu and R. Fiebrink,"Bayesian Aggregation for Hierarchical Genre Classification". Proc. International Symposium on Music Information Retrieval (ISMIR), 2007.

[11] T. Zhang,"Semi-automatic approach for music classification",Proc. Internet Multimedia Management Systems, 2003.

[12] L.E.Raileanu, K.Stoffel, "Theoretical Comparison between the Gini Index and Information Gain Criteria", Annals of Mathematics and Artificial Intelligence, vol 41, 2004.

[13] L. Chen, D. Wang,"Multivariate Decision Trees Based on Regression and Discriminant Analysis", Proc. International Conference on Convergence Information Technology (ICCIT), 2007.

[14] L. Parsons, E. Haque, ”Subspace Clustering for High Dimensional Data: A Review”, Special issue on learning from imbalanced datasets, Vol6, 2004.

[15] G. Peeters, “A large set of audio features for sound description (similarity and classification) in the cuidado project,” Technical report, Institut de Recherche et de Coordination Acoustique Musique, 2004.

[16] C. Aggarwal, C. Procopiuc, J. Yu.J and J. Park, "Fast algorithms for projected clustering", Proc. ACM International Conference on Management of Data (SIGMOD) 1999.

[17] S.Doraisamy, s.Golzari, N.Norowi, N.Sulaiman, and N.Udzir ,”A Study on Feature Selection and Classification Techniques for Automatic Genre Classification of Traditional Malay Music”, Proc. International Symposium on Music Information Retrieval (ISMIR), 2008.

[18] P. Ruddock, "Fair use and other copyright exceptions", Australia. Attorney-General's Dept, 2005.