Improving Automatic Music Classification Performance by Extracting Features from Different Types of Data

Cory McKay
CIRMMT
McGill University
Montréal, Québec, Canada
1-514-398-4535 x0300
cory.mckay@mail.mcgill.ca

Ichiro Fujinaga
CIRMMT
McGill University
Montréal, Québec, Canada
1-514-398-4535 x00944
ich@music.mcgill.ca

ABSTRACT
This paper discusses two sets of automatic musical genre classification experiments. Promising research directions are then proposed based on the results of these experiments.

The first set of experiments was designed to examine the utility of combining features extracted from separate and independent audio, symbolic and cultural sources of musical information. The results from this set of experiments indicate that combining feature types can indeed substantively improve classification accuracy as well as reduce the seriousness of those misclassifications that do occur.

The second set of experiments examined which high-level features were most important in successfully classifying symbolic data. It was found that features associated with instrumentation were particularly effective.

The paper also presents the jMIR toolset, which was used to carry out these experiments and which is particularly well suited to combining information extracted from different types of data sources. jMIR is a free and open-source software suite designed for applications related to automatic music classification of various kinds.

Categories and Subject Descriptors
H3.3 [**INFORMATION SYSTEMS**]: Information Search and Retrieval

H.5.5 [**INFORMATION SYSTEMS**]: Sound and Music Computing - *systems*

I.5.2 [**PATTERN RECOGNITION**]: Design Methodology – *feature evaluation and selection, pattern analysis*

J.5 [**ARTS AND HUMANITIES**]: Music

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

General Terms
Algorithms, Experimentation

Keywords
Music Information Retrieval, Automatic Music Classification, Genre, Features, Machine Learning, Multi-Modal

1. INTRODUCTION
Music information retrieval (MIR) research on automatic music classification has for the most part tended to focus on extracting information from three primary types of data:

- **Audio recordings:** Digital representations of physical audio signals. These are typically stored in formats such as MP3s, WAVs and FLACs.
- **Symbolic musical representations:** Representations of sound based on abstract symbols that are musically meaningful, such as the music notation used in scores. File formats such as MIDI, OSC and Humdrum are often used to store symbolic data.
- **Cultural data:** Information that is pertinent to the music at hand, but is not a direct representation, abstract or otherwise, of the actual sound associated with the music. The Internet provides the most easily mined source of cultural data, including resources such as edited metadata repositories, unedited listener tags, playlists and web sites in general.

MIR research projects involving each of these types of data has traditionally been relatively segregated from research involving the others, often based on whether a given researcher has a corresponding background in signal processing, music theory or data mining. In recent years, however, MIR researchers have increasingly begun to study these sources of information in combination, with a particular emphasis on research combining audio and cultural sources of data. The value of audio is clear, as it is the essential way in which music is consumed, and cultural information external to musical content is well known to have a large influence on our experience and interpretation of music (e.g., see [10]).

Symbolic data has recently been receiving less attention from MIR researchers, however. This is perhaps unfortunate, as much of the information associated with the types of high-level musical abstractions that can be relatively easily extracted from symbolic data is currently poorly encapsulated by the types of features that

are typically extracted from audio, which tend to focus more on low-level timbral information than on other types of information. Symbolic formats can thus, at the very least, serve as a powerful intermediate representational tool from which features incorporating high-level musical abstractions can be extracted. Research in such high-level features will also become increasingly valuable as polyphonic audio to symbolic transcription algorithms continue to improve.

It is suggested here that features extracted in combination from all three types of data can potentially provide valuable information that could be gainfully used in tasks related to automatic music classification and similarity estimation. If the orthogonal independence of the features associated with each data type is high, then performance gains can likely be attained in a variety of applications by combining features extracted from the different data types. In order to investigate this further, this paper examines experimental data that provides insights on the extent to which combining features extracted from these three types of data can be advantageous, and on which types of features can be most effective in arriving at successful classifications [13][16].

This investigation was performed via two sets of experiments involving automatic genre classification. Genre classification in particular was chosen because it is a complex and difficult task that combines diverse musical variables. Genre classification is also an area of research where classification success rates appear to have hit a "glass ceiling" in recent years. If one examines the yearly results of the MIREX (Music Information Retrieval Evaluation eXchange) audio genre classification contest, shown in Figure 1, one will observe little evidence of improvement from year to year. It is hoped that the combination of features extracted from different types of data will help to break this apparent performance ceiling.

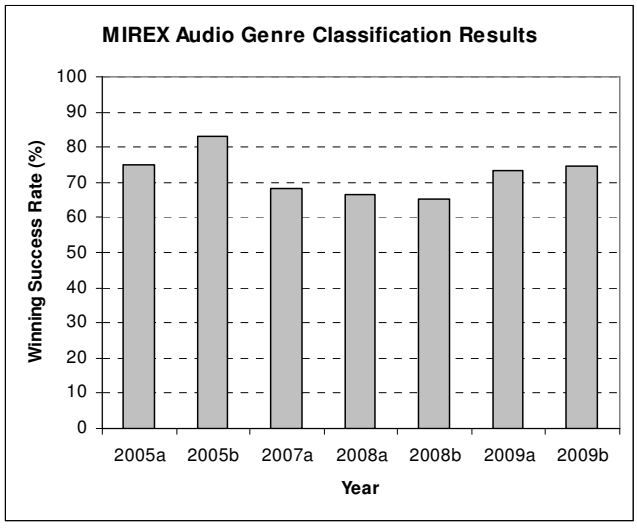

Figure 1: The best classification success rates in each of the MIREX audio genre classification competitions to date [26]. All runs involved 10 genre classes except for the 2005b run, which used a genre ontology of 6 genres. No audio genre classification competition was held in 2006, and the competitions in 2005, 2008 and 2009 each involved separate runs on two different datasets. The datasets were usually varied from year to year.

It should be noted that the experiments described in this paper could alternatively have involved other types of classification, such as mood or artist classification. The essential issue being investigated remains the potential performance improvements attained by combining features extracted from the different types of musical data.

This paper ends by summarizing the results of the experiments and by proposing directions for future research that should be prioritized based on the results of the experiments.

2. RELATED RESEARCH

There has been a significant amount of research on combining features extracted from audio and cultural data that can be extracted from the Internet. Whitman and Smaragdis [21] performed particularly important early work of this kind, and achieved substantial performance gains when doing so. Dhanaraj and Logan [4] took a more content-based approach by combining information extracted from lyrics and audio. Others have combined audio and cultural data for the purpose of generating music browsing spaces (e.g., [6]). To give another example, Aucouturier and Pachet [2] used a hybrid training approach based on acoustic information and boolean metadata tags. Research has also been conducted on using audio data to make correlations with cultural labels, which can in turn improve other kinds of classification (e.g., [18]).

There has been much less work on combining symbolic data with audio data. In one of the few such research projects, Lidy and his colleagues [7] found that combining audio and symbolic data can result in improved performance compared to when only audio data is used. To the best knowledge of the authors, the experiments described here represent the first research on combining symbolic and cultural features and on combining all three feature types.

Far too many papers have been published on automatic genre classification in general to cite with any completeness here. One influential publication that bears particular mention, however, is that of Tzanetakis and Cook [20].

3. THE JMIR SOFTWARE SUITE

3.1 Overview

jMIR is an open-source Java software suite consisting of tools for performing the essential tasks associated with automatic music classification research. It can be used to extract information from music in both audio (jAudio) and symbolic (jSymbolic) forms, as well as to mine cultural information from the Internet (jWebMiner). It also includes software for applying machine learning algorithms (ACE) and analyzing and managing metadata associated with musical datasets (jMetaMusicManager). The ACE XML file formats, which are also part of jMIR, provide an expressive and flexible general standard for storing and exchanging information related to automatic music classification. jMIR also includes several labeled datasets for performing musical research and validating new algorithms.

There are three primary priorities underpinning the design of the jMIR components. The first is that they be easy to install and use by individuals with a variety of technical backgrounds. This is essential, as researchers in fields such as musicology, music

theory, psychology and the library sciences have important musical insights that can greatly benefit MIR research, but are often alienated by software requiring a strong technical background to use. Installation difficulties and steep learning curves can also be discouraging for even technically skilled users. The jMIR components are consequently well-documented, relatively simple to install and include easy-to-use GUIs.

The second priority is the provision of an open framework for performing novel research and distributing new approaches to others. This is important in ensuring research transparency and in allowing researchers to evaluate and build upon each other's work. In order to accomplish this, the jMIR components are designed using a modular plug-in approach. It is thus a relatively simple matter for researchers to both develop their own algorithms and add algorithms newly implemented by others to their jMIR distributions. The jMIR feature extractors also automatically make the value of each extracted feature available to each other feature and automatically consider dependencies when dynamically scheduling extraction order, something that greatly facilitates iterative feature design.

The third priority is the provision of functionality allowing features extracted from audio recordings, symbolic representations and cultural data available on the Internet to be combined. As noted above, music researchers traditionally tend to focus on only one of these domains. They consequently risk failing to fully take advantage of valuable complementary sources of information, as demonstrated by the experimental results described in the sections below,

The jMIR software and related documentation may be downloaded from jmir.sourceforge.net.

3.2 jAudio

jAudio [9] is an application for extracting features from audio files in formats such as MP3, AIFF and WAV. It is bundled with 28 implemented features associated with both the frequency and time domains (e.g., Spectral Flux, RMS, etc.). It includes several intermediate-level musical features, mainly related to rhythm, as well as lower-level signal processing-oriented features. A variety of pre-processing options are also available, such as down-sampling, normalization and windowing.

In order to make jAudio as accessible as possible, it includes a GUI interface, a Java API to facilitate integration with other software and a command-line interface for those wishing to perform batch processing. jAudio also includes functionality for synthesizing, recording and displaying audio for the purpose of testing new features.

jAudio places a particular emphasis on facilitating the process of developing and adding new features. As is also the case with jSymbolic, new features can be added to jAudio using a simple inheritance-based plug-in approach that automatically and dynamically solves scheduling dependencies. jAudio also includes implementations of "metafeatures" and "aggregators" that respectively automatically implement features derived from other features (e.g., the standard deviation of a feature across analysis windows) and collapse a set of feature vectors into a single vector or a smaller set of vectors (e.g., area of moments).

3.3 jSymbolic

jSymbolic [14] is a GUI-based application for extracting features from MIDI files. It is bundled with 111 implemented features (e.g., Note Density, Instruments Present, Range, etc.), most of which are based on relatively high-level musical abstractions and many of which are novel. The features fall into the broad categories of instrumentation, texture, rhythm, dynamics, pitch statistics and melody. jSymbolic has far more features than any other existing symbolic feature extractors.

Like jAudio, jSymbolic has a simple inheritance-based modular API for adding new features, and feature dependencies are resolved automatically in order to encourage the iterative development of increasingly high-level features (e.g., using features related to pitch to extract features relating to chords, which can in turn be used to extract features related to harmonic progressions).

An additional 49 features are also proposed for future implementation, including features associated with chords [10].

3.4 jWebMiner

jWebMiner [15] is a GUI-based application for extracting features from textual information found on the Internet. At its most basic level, the software operates by automatically using Google and Yahoo! web services to acquire statistics on how often particular strings co-occur on the same web pages. This can indicate artist similarity, for example, by measuring how often artists' names co-occur with one another. It can also be used to classify artists by genre by measuring how often their names co-occur with various genre titles.

Search results are processed statistically by jWebMiner in a variety of ways in order to remove noise and improve results. Further processing options include the abilities to filter out sites containing specified strings, to require that sites contain certain strings in order to be counted and to weight results from multiple sites differently.

jWebMiner can parse iTunes XML, ACE XML, Weka ARFF [22] and delimited text files in order to conveniently access the particular strings to use in searches.

3.5 ACE

ACE [12][19] is a meta-learning software package for selecting, optimizing and applying machine learning algorithms. Given a set of extracted features, ACE experiments with a variety of classifier algorithms, parameters, ensemble architectures and dimensionality reduction techniques in order to arrive at a good configuration for the problem at hand. This can be helpful, as a particular algorithm can be more or less appropriate for a given problem in terms of classification accuracy, training speed and classification speed. Even experts in machine learning can have difficulty choosing the best algorithm and parameterization for a given problem, to say nothing of musical experts with limited backgrounds in computer-based research.

ACE is designed to automate the choice of algorithm and to facilitate the use of powerful machine learning technology by users of all technical levels. ACE also provides a framework for experimenting with new algorithms. ACE is built on top of the popular Weka machine learning framework [22], so new

algorithms developed using the Weka API can be easily added to ACE. ACE may also be used directly as a classifier.

ACE improves upon Weka by, in addition to its implementation of meta-learning functionality, adding a custom cross-validation system that is more flexible and open than Weka's approach. ACE also calculates additional statistics that can be helpful in comparing and evaluating algorithms.

3.6 ACE XML

ACE XML [11][12] is a set of standardized file formats for representing feature values extracted from instances; abstract feature descriptions and parameterizations; instance labels and annotations; and class ontologies.

These file formats have been designed to address the significant shortcomings with respect to automatic music classification of the file formats most commonly used in MIR. To provide just a few examples, ACE XML makes it possible to associate multiple weighted class labels with a single instance, to specify ontological relationships between class labels; to group associated feature values in ways that can be meaningful to machine learning algorithms; to express feature arrays of arbitrary dimensionality; to maintain associations between instances and their sub-sections and metadata; to reduce file sizes using compression; and to link to external resources using RDF-like triples. None of the data exchange formats traditionally used in MIR research, such as Weka ARFF, provide all of these options.

3.7 jMusicMetaManager

jMusicMetaManager [17] is a GUI-based software package for profiling and managing large musical datasets and for detecting metadata errors and inconsistencies in them. These tasks are essential, as the success of ground-truth training and evaluation data is contingent upon the quality of the musical datasets from which they are drawn.

jMusicMetaManager uses many metrics to find dataset inconsistencies and redundancies. These can be used to detect mislabeled duplicate recordings that could cross-contaminate training and testing sets, for example, or to detect varying labeling conventions that might cause "Mingus, Charles" and "Charlie Mingus", for example, to be erroneously treated as two different artists during training and evaluation.

In all, jMusicMetaManager provides users with 23 pre-processing operations, and includes several edit-distance and word ordering/subset error detection operations. A total of 39 different HTML reports can also be automatically generated to help profile and publish information about musical datasets.

jMusicMetaManager can parse iTunes XML files and MP3 ID3 tags as well as ACE XML and Weka ARFF files in order to access the metadata that is to be analyzed.

3.8 jMIRUtilities

jMIRUtilities is a set of tools for performing miscellaneous useful tasks associated with jMIR. These tools include a GUI for batch-associating class labels with instances, utilities for merging various kinds of information, and utilities for extracting structured information from iTunes XML files or delimited text files.

3.9 Codaich and Bodhidharma MIDI

Codaich [17] is an audio dataset consisting of 31,300 MP3 recordings by 2811 artists and belonging to 57 genres of music. These recordings are labeled with 19 metadata fields. The Bodhidharma MIDI dataset [10] is a collection of 950 MIDI recordings belonging to 38 genres. These datasets have both previously been used in research projects involving the jMIR software components

These datasets are intended to eventually be made publicly accessible using an OMEN-like [8] system. Such systems enable custom feature extraction requests to be processed at distributed sites with legal access to music so that the features can then themselves be distributed elsewhere without violating copyright legislation.

3.10 The SAC Dataset

The SAC (Symbolic, Audio and Cultural) dataset [16] was assembled in order to provide matching symbolic, audio and cultural data specifically for use in the experiments described in Section 4 below. SAC consists of 250 MIDI files and 250 matching MP3s, as well as accompanying metadata (e.g., title, artist, etc.) for each recording. This metadata is stored in an iTunes XML file, which can be parsed by jWebMiner in order to extract cultural features from the web for each of the associated recordings.

It was decided to acquire the matching MIDI and audio recordings separately, rather than simply synthesizing the audio from the MIDI. Although this made acquiring the dataset significantly more difficult and time consuming, it was considered necessary in order to truly test the value of combining symbolic and audio data. This is because audio generated from MIDI by its nature does not include any additional data other than the very limited information encapsulated by the synthesis algorithms.

SAC is divided amongst 10 different genres, with 25 pieces of music per genre. These 10 genres can be grouped into 5 pairs of similar genres, as shown in Figure 2. This arrangement makes it possible to perform 5-class genre classification experiments as well as 10-class experiments simply by combining each pair of related genres into one class. An additional advantage is that it becomes possible to measure indications of how serious misclassification errors are in 10-class experiments by examining how many misclassifications are in an instance's partner genre rather than one of the other 8 genres.

SAC was designed to be more difficult to classify by genre than the types of datasets normally used in automatic genre classification experiments in order to more realistically simulate real-life genre classification problems. In addition to using pairs of similar genres, which differs from the standard practice of using only genres that are very different from one another, and thus easier to discriminate between, SAC also includes multiple versions of the same pieces in different genres as well as examples of different pieces in different genres by the same artist. This helps to verify that genres themselves are being modeled by pattern recognition algorithms, not just characteristics of individual artists or pieces.

Blues: Modern Blues *and* Traditional Blues
Classical: Baroque *and* Romantic
Jazz: Bop *and* Swing
Rap: Hardcore Rap *and* Pop Rap
Rock: Alternative Rock *and* Metal

Figure 2: The ten genre pairs of the SAC dataset.

4. EXPERIMENT 1: EFFECTS OF COMBINING DATA TYPES

4.1 Experimental Procedure

The first set of experiments was designed to investigate the utility of combining features extracted from different types of musical data. In order to accomplish this, jMIR's three feature extractors were used to extract features from each matched audio recording, MIDI recording and set of metadata in SAC. Details on the particular features extracted are available elsewhere [9][10][14][15].

To provide a clarifying example, features might be extracted from a Duke Ellington MP3 recording of *Perdido,* from an independently acquired MIDI encoding of the same piece and from automated search engine queries using metadata such as artist and title. Three feature sets were therefore extracted for each piece, one corresponding to each of the three data types.

These three types of features were then grouped into all seven possible subset combinations. This was done once for the 5-genre SAC ontology and once for the 10-genre SAC ontology, for a total of 14 sets of features (as shown in Table 1). ACE was then used to perform 14 independent 10-fold cross-validation experiments,[1] one for each of the feature sets. This resulted in two classification accuracy rates for each of the seven feature type combinations, one for each of the two SAC genre ontologies.

It was desirable not only to determine how effective each of the feature type combinations were at performing accurate classifications, but also in gaining insight on the seriousness of those misclassifications that did arise. Two classifiers with similar raw classification accuracy rates can in fact be of very different value if the misclassifications made by one classifier consistently result in classes that are less similar to the "correct" class. For example, misclassifying John Lennon as The Beatles in an artist identification task would be less serious than misclassifying him as Kelly Clarkson.

A weighted classification accuracy rate was calculated for each of the 10-genre experiments in order to examine this issue. This weighted rate was calculated by weighting a misclassifications within a genre pair (e.g., Alternative Rock instead of Metal) as 0.5 of an error, and by weighting a misclassification outside of a pair (e.g., Swing instead of Metal) as 1.5 of an error.

[1] ACE includes dimensionality reduction functionality, so models were actually trained with automatically chosen subsets of the available features.

4.2 Results and Discussion

4.2.1 Data

The average classification accuracy rates across cross-validation folds for each of the 14 experiments are shown in Table 2, including both weighted and unweighted results. Figures 3 and 4 illustrate the unweighted results for the 5-genre and 10-genre experiments respectively. These results are summarized in Table 3 and Figure 5, which indicate the average results for all experiments using one feature type, all experiments using two feature types and all experiments using three feature types.

4.2.2 Effects of Combining Data Types on Accuracy

As can be seen in Figures 3 and 4, all combinations of two or three feature types performed substantially better than all single feature types classified independently. Furthermore, combining all three feature types resulted in better performance than most pairs of feature types.

This result is highlighted in Figure 5, which shows important average increases in classification performance when feature types are combined. Combining all three feature types resulted in increases in performance of 11.3% on the 5-genre ontology and 13.7% in the 10-genre ontology, compared to the average performances of each of the single feature types classified individually. Considered in terms of percentage reduction in error rate, this corresponds to impressive improvements of 78.0% and 39.3% for the 5 and 10-genre genre ontologies, respectively.

A Wilcoxon signed-rank test indicates that, with a significance level of 0.125, the improvements in performance of two or three feature types over one type were statistically significant in all cases. However, the improvements when three feature types were used instead of two were not statistically significant, as the corresponding average increases in performance were only 2.3% and 2.7% for the 5 and 10-genre ontologies, respectively.

Overall, these results provide supportive evidence that the three different types of features contain at least some orthogonally independent information, and can therefore be profitably combined for a variety of purposes.

4.2.3 Types of Misclassification

As described in Section 4.1, weighted classification accuracy rates were calculated for the experiments on the 10-genre ontology in order to evaluate the seriousness of the particular misclassifications that did occur. The results, and how they compare to the unweighted classification accuracies, are shown in Table 2 and Figure 6.

The weighted and unweighted accuracies were not significantly different when the audio and symbolic features were processed individually. However, the weighted performance was 3% higher than the unweighted performance when these two feature types were combined. Although this is not a dramatic increase, it is an indication that combining these feature types may make those misclassifications that do occur be at least somewhat closer to the model classes, in addition to increasing classification accuracy itself, as discussed in Section 4.2.2.

Of greater significance, the differences between the weighted and unweighted classification accuracies were greater in all feature

Table 1: The identifying codes for each of the 14 parts of Experiment 1.

Feature Type(s)	5-Genre Code	10-Genre Code
Symbolic	S-5	S-10
Audio	A-5	A-10
Cultural	C-5	C-10
Symbolic + Audio	SA-5	SA-10
Audio + Cultural	AC-5	AC-10
Symbolic + Cultural	SC-5	SC-10
Symbolic + Audio + Cultural	SAC-5	SAC-10

Table 2: The unweighted classification accuracy rates for the 5-genre (5-UW) experiments and both the unweighted (10-UW) and weighted (10-W) classification rates for the 10-genre experiments. Results are reported for each feature type combination, as described in Table 1. All values are average percentages calculated over cross-validation folds.

	S	A	C	SA	AC	SC	SAC
5-UW	86.4	82.8	87.2	92.4	95.2	94	96.8
10-UW	66.4	67.6	61.2	75.6	78.8	75.2	78.8
10-W	66.4	67.4	66.6	78.6	84.6	81.2	84.2

Table 3: The average classification accuracy rates for all experiments employing just one type of feature (S, A and C), two types of features (SA, AC and SC) or all three types of features (SAC). Results are specified for the 5-genre ontology (5-UW), the unweighted 10-genre ontology (10-UW) and the weighted 10-genre ontology (10-W). All values are percentages, and are calculated as simple mean averages from Table 2.

	1 Type	2 Types	3 Types
5-UW	85.5	93.9	96.8
10-UW	65.1	76.5	78.8
10-W	66.8	81.5	84.2

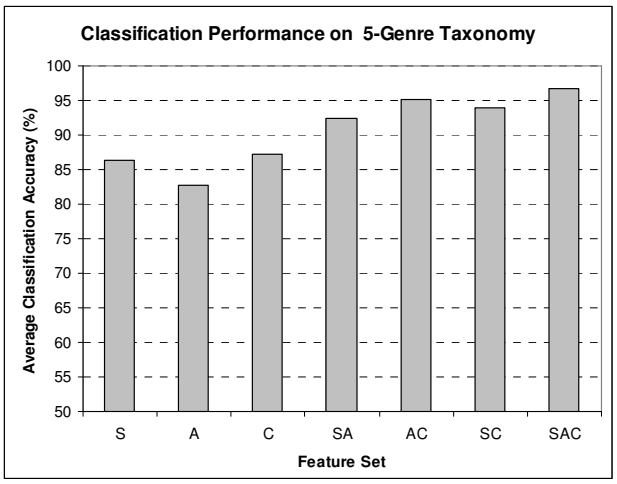

Figure 3: The classification accuracy rates for the 5-genre ontology, as described in Table 2.

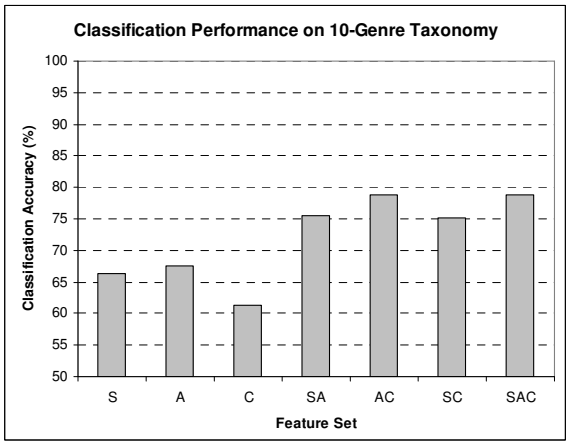

Figure 4: The unweighted classification accuracy rates for the 10-genre ontology, as described in Table 2.

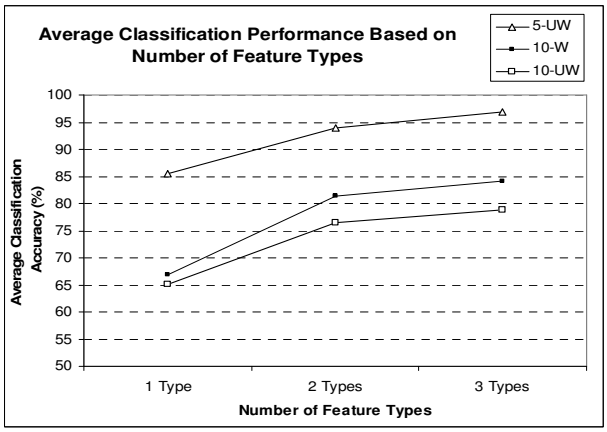

Figure 5: The average classification accuracy rates for all experiments employing just one type of feature, two types of features or all three types of features, as described in Table 3.

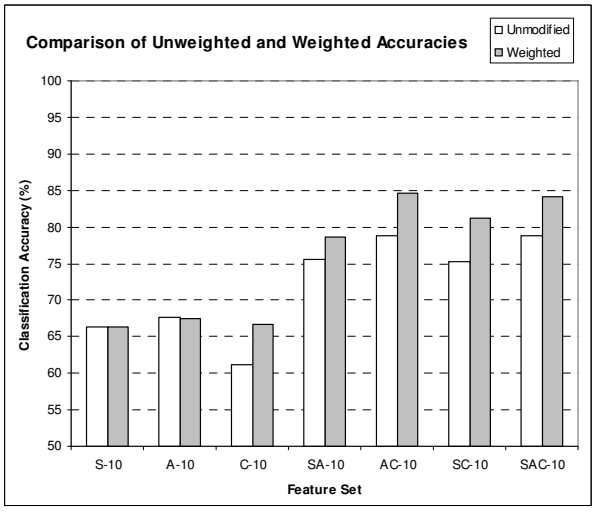

Figure 6: The differences between the unweighted and weighted classification accuracies on the 10-genre ontology for each of the feature type combinations, as described in Table 2.

sets that included cultural features. These weighted rates were higher than the unweighted rates by an average of 5.7%, a difference that, based on Student's paired t-test, is statistically significant with a significance level of 0.005.

Overall, these results indicate that the types of misclassifications that occur when cultural features are used are less serious than when audio or symbolic features are used alone. Quite encouragingly, it also appears that this improvement in error quality carries through when cultural features are combined with audio and symbolic features.

4.2.4 Overall Performance

In order to put the experimental results described here in context, it is appropriate to compare them with classification accuracies achieved by other genre classification systems. It is important, however, to keep in mind the essential caveat that different classification systems can perform dramatically differently on different datasets, so direct comparisons of experiments performed on different datasets can give only a very rough indication of comparative performance.

The MIREX evaluations offer the best benchmarking reference points available. Although no evaluations of genre classification based on cultural data have been carried out yet at MIREX, both symbolic and audio genre classification evaluations have been held, most recently in 2005 and 2009, respectively. The highest accuracy for symbolic classification was 84.4%, attained on a 9-genre ontology by McKay and Fujinaga's Bodhidharma system [23]. The highest classification accuracy attained in general audio genre classification in 2009 was 73.3%, achieved by Cao and Li on a 10-genre ontology [24].

The experiments described in this paper achieved classification accuracies of 67.6% using only features extracted from audio and 66.4% using only features extracted from symbolic data. This is lower but possibly comparable to the best MIREX audio result of 73.3%, but significantly lower than the best MIREX symbolic result of 84.4%, which was achieved on an ontology only smaller by one class (9 vs. 10).

This latter result is intriguing, as jSymbolic uses the same features and feature implementations as Bodhidharma. The difference may be due at least in part to the specialized and sophisticated hierarchical, round-robin and flat learning ensemble algorithms used by Bodhidharma [10], whereas ACE only experiments with general-purpose machine learning algorithms.

When all three feature types were combined, the jMIR experiments described in this paper achieved a success rate of 78.8% which was still lower than Bodhidharma's performance, but higher than the best audio MIREX results to date.

Taken in the context of the particular difficulty of the SAC dataset (see Section 3.10), and when it is considered that the accuracy on the 10-genre ontology improves to 84.2% when weighted, the results attained here are encouraging, and may be an indication that the ultimate ceiling on performance might not be as low as some have worried [1]. It may well be that the use of more sophisticated machine learning approaches, such as those used by Bodhidharma or by DeCoro et al. [3], combined with the development of new features that highlight particularly pertinent information, could significantly improve performance still further.

5. EXPERIMENT 2: FOCUSING ON HIGH-LEVEL FEATURES

5.1 Experimental Procedure

The goal of the second experiment was to gain empirical insight into which high-level features were the most effective in classifying symbolic recordings by genre. The jSymbolic features were extracted from the Bodhidharma MIDI dataset and were classified by a classifier ensemble consisting of ensembles of multilayer perceptrons and k-NN classifiers [10]. Genetic algorithms were used to evolve feature selections and weightings to use when training the classifiers in order to avoid the curse of dimensionality and, more importantly from the perspective of this particular experiment, to provide an indication as to which features were the most useful in performing classifications.

Two separate experiments were conducted, one involving the full 38-class hierarchical Bodhidharma MIDI dataset and another involving only a subset of this dataset consisting of 9 leaf genres, each belonging to one of the 3 parent genres of Classical, Jazz and Popular. The second ontology was designed to provide a set of classes comparable in size to ontologies used in other automatic genre classification experiments, and the first was designed in order to permit tests under more realistic conditions.

5.2 Results and Discussion

The MIDI recordings were correctly classified by leaf genre 90% of the time for the 9-class ontology and 57% of the time for the 38-class ontology. The root genre was correctly classified 98% of the time for the 9-class ontology and 81% of the time for the 38-class ontology. The results for the 9-class ontology are shown in more detail in Figure 7.

These results demonstrate the effectiveness of the kind of high-level features making up the jSymbolic feature set with respect to the small to medium size class ontologies typically dealt with in the MIR literature. Less encouragingly, these results also demonstrate the work that remains to be done with respect to realistically sized large ontologies.

Table 4 indicates the average weightings evolved by the GAs for the features comprising each of the 6 jSymbolic feature groups. It can be seen that the instrumentation-based features were collectively given a weighting of 41.8%, a value over twice as large as their numerical representation in the complete feature set (18%). Features based on instrumentation were also weighted much higher than any of the other feature groups, with the next highest contender (pitch-based features) only receiving a collective weighting of 27.8%, an amount much closer to its numerical portion of the total feature set (22%). Furthermore, two of the top three individual features were based on instrumentation.

These results indicate that information related to instrumentation can be particularly effective in classifying music by genre. This is an interesting result, as instrumentation can be seen as a high-level representation of timbre, something that is more typically associated with audio features. Indeed, the majority of features typically extracted from audio typically are timbre-related, something that has led some to question whether this emphasis on timbre may be at least partly responsible for the failure of audio genre classification systems to improve significantly in recent years. The results from this experiment indicate that timbre-

related information can in fact be very effective, at least at a relatively high level of musical abstraction.

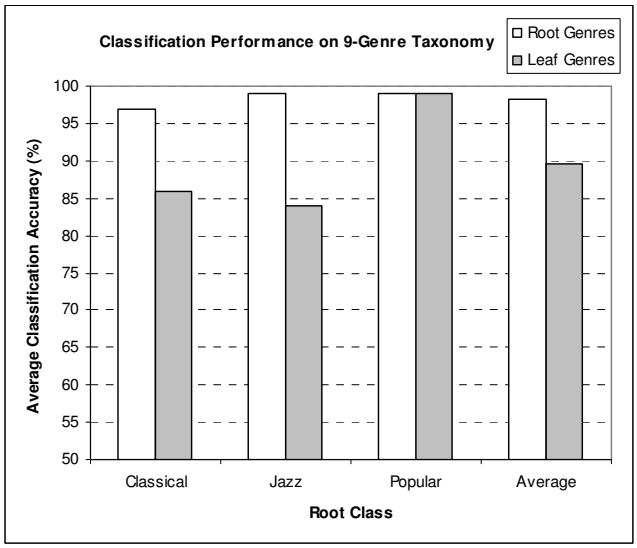

Figure 7: The classification accuracy rates for the 9-genre experiment. Results are reported for each of the leaf genres as well as for the 3 root genres. All values are average percentages calculated over cross-validation folds.

Table 4: Relative importance of feature types. The *Number of Features* column indicates the number of candidate features in the given group, with the percentage of the jSymbolic feature library that this represents in parentheses. The *Weighting* column indicates the cumulative weighting evolved by the feature selection and weighting genetic algorithms for the features in each of the feature groups.

Feature Group	Number of Features	Weighting
Instrumentation	20 (18%)	41.8%
Pitch	25 (22%)	27.8%
Rhythm	30 (27%)	19.5%
Melody	18 (16%)	9.5%
Texture	14 (13%)	1.1%
Dynamics	4 (4%)	0.3%

6. CONCLUSIONS & FUTURE RESEARCH

The results of the first set of experiments confirm that it can indeed be beneficial to combine features extracted from audio, symbolic and cultural data sources when classifying music by genre. All feature groups consisting of two feature types performed significantly better than any single feature type classified alone. Combining all three feature types also resulted in small further improvements over the feature type pairs on average, although these additional improvements were not statistically significant. It was also found that combining feature types tends to cause those misclassifications that do occur to be less serious, as the misclassifications are more likely to be to a class that is more similar to the model class. Such improvements were particularly pronounced when cultural features were involved. Overall, encouragingly high genre classification accuracy rates were attained on a relatively difficult dataset when feature types were combined, something that provides hope that any ultimate ceiling on genre classification performance might not be as low as some have worried.

The results of the second set of experiments emphasized the particular importance of features associated with instrumentation in classifying symbolic music by genre. As noted above, these results were somewhat surprising given that most audio genre classifiers emphasize timbre, and such features have so far been ineffective in achieving significant classification improvements. Experiment 2 therefore suggests that timbral information can in fact be very effective, but at a high level of abstraction.

Taken together, the results of these two experiments suggest several interesting research directions. Audio data provides useful low-level information that is not fully encapsulated in symbolic data, and symbolic data represents high-level musical information that can be difficult to derive from the types of features usually used in audio music classification. The benefits of combining both low-level and high-level features were made apparent by the results of Experiment 1, where the combination of low-level features extracted from audio and high-level features extracted from symbolic data resulted in significant improvements over features extracted from either type of data individually (see Table 2). In practice, of course, most researchers are more interested in audio data than symbolic data, and acquiring both symbolic and audio versions of a given piece can be difficult. So, while it is beneficial to combine features extracted from both audio and symbolic representations of a given piece of music, it is not always practically feasible to acquire independent symbolic and audio versions of a piece.

This suggests that it would be profitable to focus research efforts on extracting both high-level and low-level features directly from audio. It stands to reason that the combination of both feature types might indeed effectively provide access to the same gains achieved by combining symbolic and audio data, but without the need for separately acquired symbolic representations. Although several researchers certainly have developed and used intermediate-level features extracted from audio in the past, such as features derived from beat histograms and pitch histograms (e.g., [20]), these features do not provide access to the same variety and depth of high-level information easily accessible in MIDI files.

A first step suggested by the results of Experiment 2 would be to focus on developing instrument identification pre-processing systems that could be used to generate features similar to the instrumentation-based features found in the jSymbolic feature library, something that has not been done previously to the best of the authours' knowledge. Other high-level timbre-related features could also be developed, such as features based on audio production characteristics or on instrument-specific performance gestures (e.g., bowing speed or pressure).

Many of the pitch-based and rhythm-based features extracted by jSymbolic, which were respectively found to be the second and third most important feature types in Experiment 2, could also be accessed from audio using current technologies. Such jSymbolic features incorporate a much higher level of musical abstraction

than the low-level and mid-level beat and pitch features used in most audio genre classification systems.

Although the current state of the art does not always make it possible to access such high-level information with perfect or even near perfect accuracy, it is important to note that machine learning-based classification algorithms can be relatively robust to such noise, as found by Lidy and his colleagues [7]. An investment now in developing high-level features could pay increasing dividends as automatic polyphonic transcription technologies eventually improve to the point where one can derive full symbolic transcriptions from audio recordings containing at least as much information as MIDI files.

It is also possible to access cultural and other types of data relatively easily from audio files by using fingerprinting technology, such as that made available by MusicBrainz [25], to extract identifying tags that can in turn be used to mine the Internet for cultural data. Such tags could also be used to access other kinds of information that could potentially be combined profitably with audio, symbolic and cultural data, such as features extracted from lyrics or album art. Hu, Downie and Ehman [5], for example, have already achieved compelling results by combining features extracted from lyrics with features extracted from audio and cultural information available on-line.

All of this means that one could simply take a given audio file and use it to extract not only the types of features typically associated with audio files, but also features more often associated with symbolic, cultural, lyrical, visual and other data.

Further research also remains to be performed investigating whether the benefits observed in Experiment 1 generalize to other types of music classification, such as classification by mood or artist. Additional research could also be pursued using more sophisticated machine learning techniques. For example, one might combine feature types in more sophisticated ways, such as by segregating them among different weighted specialist classifiers collected into blackboard ensembles. One might also use classification techniques that take full advantage of ontological class structuring.

It is notable that the jMIR software suite was demonstrated to be an effective and convenient tool for performing feature extraction and classification research on different types of musical data. The jMIR components will continue to be expanded and improved by the authors and, it is hoped, by the music research community as a whole. The jMIR components provide an infrastructure for collaborative feature development, and also provide powerful core libraries of features, machine learning algorithms and musical data that can help avoid duplicated effort.

7. ACKNOWLEDGMENTS

The authors would like to thank the *Andrew W. Mellon Foundation,* the *Centre for Interdisciplinary Research in Music Media and Technology* (CIRMMT), the *Social Sciences and Humanities Research Council* (SSHRC) and the *Canadian Foundation for Innovation* (CFI) for their generous financial support.

8. REFERENCES

[1] Aucouturier, J. J., and F. Pachet. 2004. Improving timbre similarity: How high is the sky?" *Journal of Negative Results in Speech and Audio Sciences* 1 (1).

[2] Aucouturier, J. J., and F. Pachet. 2007. Signal + context = better. Proceedings of the International Conference on Music Information Retrieval. 425–30.

[3] DeCoro, C., Z. Barutcuoglu, and R. Fiebrink. 2007. Bayesian aggregation for hierarchical genre classification. *Proceedings of the International Conference on Music Information Retrieval*. 77–80.

[4] Dhanaraj, R., and B. Logan. 2005. Automatic prediction of hit songs. *Proceedings of the International Conference on Music Information Retrieval*. 488–91.

[5] Hu, X., J. S. Downie, and A. F. Ehman. 2009. Lyric text mining in music mood classification. *Proceedings of the International Society for Music Information Retrieval Conference*. 411–6.

[6] Knees, P., M. Schedl, T. Pohle, and G. Widmer. 2006. An innovative three-dimensional user interface for exploring music collections enriched with meta-information from the web. *Proceedings of the ACM International Conference on Multimedia*. 17–24.

[7] Lidy, T., A. Rauber, A. Pertusa, and J. M. Iñesta. 2007. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. *Proceedings of the International Conference on Music Information Retrieval*. 61–6.

[8] McEnnis, D. 2006. On-demand metadata extraction network (OMEN). *M.A. Thesis*. McGill University, Canada.

[9] McEnnis, D., C. McKay, and I. Fujinaga. 2006. jAudio: Additions and improvements. *Proceedings of the International Conference on Music Information Retrieval*. 385–6.

[10] McKay, C. 2004. Automatic genre classification of MIDI recordings. *M.A. Thesis*. McGill University, Canada.

[11] McKay, C., J. A. Burgoyne, J. Thompson, and I. Fujinaga. 2009. Using ACE XML 2.0 to store and share feature, instance and class data for musical classification. *Proceedings of the International Society for Music Information Retrieval Conference*. 303–8.

[12] McKay, C., R. Fiebrink, D. McEnnis, B. Li, and I. Fujinaga. 2005. ACE: A framework for optimizing music classification. *Proceedings of the International Conference on Music Information Retrieval*. 42–9.

[13] McKay, C., and I. Fujinaga. 2005. Automatic music classification and the importance of instrument identification. *Proceedings of the Conference on Interdisciplinary Musicology*. CD-ROM.

[14] McKay, C., and I. Fujinaga. 2006. jSymbolic: A feature extractor for MIDI files. *Proceedings of the International Computer Music Conference*. 302–5.

[15] McKay, C., and I. Fujinaga. 2007. jWebMiner: A web-based feature extractor. *Proceedings of the International Conference on Music Information Retrieval*. 113–4.

[16] McKay, C., and I. Fujinaga. 2008. Combining features extracted from audio, symbolic and cultural sources. *Proceedings of the International Conference on Music Information Retrieval*. 597–602.

[17] McKay, C., D. McEnnis, and I. Fujinaga. 2006. A large publicly accessible prototype audio database for music research. *Proceedings of the International Conference on Music Information Retrieval*. 160–3.

[18] Reed, J., and C. H. Lee. 2007. A study on attribute-based taxonomy for music information retrieval. *Proceedings of the International Conference on Music Information Retrieval*. 485–90.

[19] Thompson, J., C. McKay, J. A. Burgoyne, and I. Fujinaga. 2009. Additions and improvements to the ACE 2.0 music classifier. *Proceedings of the International Society for Music Information Retrieval Conference*. 435–40.

[20] Tzanetakis, G., and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10 (5): 293–302.

[21] Whitman, B., and P. Smaragdis. 2002. Combining musical and cultural features for intelligent style detection. *Proceedings of the International Symposium on Music Information Retrieval*. 47–52.

[22] Witten, I., and E. Frank. 2005. *Data mining: Practical machine learning tools and techniques with Java implementations*. San Francisco: Morgan Kaufmann.

[23] 2005 MIREX contest results – symbolic genre classification. Retrieved 10 December 2009, from http://www.music-ir.org/evaluation/mirex-results/sym-genre/index.html.

[24] Audio genre classification (mixed set) results. Retrieved 10 December 2009, from http://www.music-ir.org/mirex/2009/index.php/Audio_Genre_Classification_%28Mixed_Set%29_Results.

[25] MusicBrainz. Retrieved 10 December 10 2009, from http://musicbrainz.org.

[26] Virtual home of music information research. Retrieved 10 December 10 2009, from http://www.music-ir.org.