Genre Classification Using Graph Representations of Music

Rachel Mellon, Dan Spaeth, Eric Theis

November 14, 2014

Abstract

A song can be represented by a graph, where nodes and edges represent individual pitch-duration tuples and co-occurrence of multiple notes respectively. A set of features can be derived from said graph for use in a variety of classification algorithms. In an attempt to derive meaning and utility from these graph features, we tackled the issue of genre classification–a highly subjective form of categorization in and of itself. We aimed to create a high performing method of genre classification by examining the capabilities of the algorithms SVM, Naive Bayes, multinomial logistic regression, and KNN using the aforementioned features as inputs.

1 Background

The concept of using machine learning to expedite, improve, or enhance the ability to assign genre classifications to music is not a new one. "Classification of Musical Genre: A Machine Learning Approach," a paper authored in 2004 at University of Rome Tor Vergata, provides a breakdown of various machine learning techniques (Naive Bayes, Voting Feature Intervals, J48, PART, NNge, and JRip) and their performance in classifying music based on features extracted from their MIDI files. For this study, researchers highlighted features that functioned like a musical fingerprint for a song, including relative frequency in melodic intervals, which instruments are used, meter and time changes, and extremes such as longest note, highest note, and lowest note. Ultimately, Naive Bayes performed the best, with about 70% accuracy (depending on the testing strategy) [2].

Another paper, "Music Genre Classification," covers a few additional classification methods such as KNN, K-means, SVM, and neural networks, while using Mel Frequency Cepstral Coefficients (MFCC) as features. The highest performing of these algorithms was neural networks with 96% accuracy, although all four algorithms struggled with differentiating two "similar" genres (jazz vs classical, metal vs jazz) [3].

In the past, graph representations of music have been used to derive melodic similarity [12]. In our analysis, we pursued other types of information that could potentially be extracted or derived from graph representations in order to apply those to genre classification.

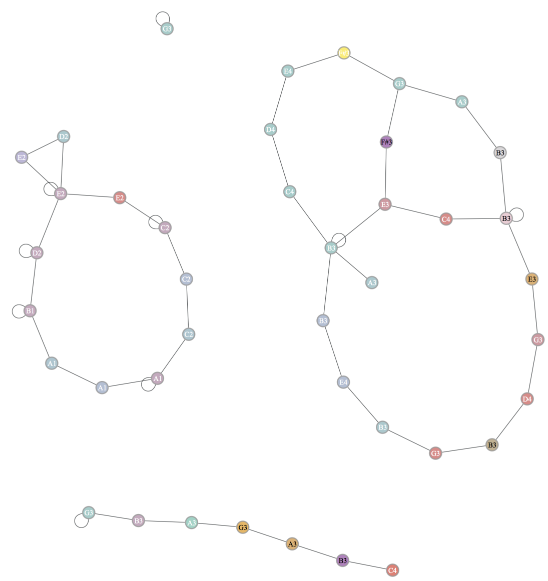
2 Methodology

For our purposes, we explored an unexamined set of features with respect to genre classification. Using musical parsing scripts (which leverage the MIT music21 library and the graph-tools python libraries), we translated MIDI files into graph representations of music, which we then analyzed for patterns to function as features for this classification.

2.1 Music as a Graph

In order to convert our MIDI files into feature vectors, we generated graph representations of each MIDI song.

Conceptually, a song is represented as a graph where each node contains a unique pitch-duration tuple, and each directed edge represents the second node occuring directly after the first node in the song. As such, edge weights represent the number of times the first pitch-duration note is followed immediately by the second pitch-duration note. Self-loops represent repeating pitch-duration notes. Connected components each represent a different part in the song (where all parts are heard simultaneously in the original song). This idea has been explored in depth by Panos Achlioptas [1].



The above figure depicts our graph representation of the song Black and Gold by Sam Sparro.

2.2 Features

The features that we used come from three main categories: music properties, graph properties, and motif properties. Music properties were extracted from each MIDI file and included (after feature selection) only the numerator of the time signature.

Graph properties were derived from the graph representation of each song. They were clustering co-efficients, pseudo diameter, number of nodes, number of edges, number of self loops, and average edge weight.

Finally, we derived motif counts for each graph (where a motif is a non-isomorphic induced subgraph). The algorithm to derive these count-based features required two steps. The first was finding the count of all induced subgraphs or order k, where k is a predefined number of nodes in each subgraph. The second was aggregating these counts, by grouping each isomorphism class (in other words, adding the counts of all isomorphic subgraphs together).

2.3 Models

We investigated several models, which we selected because of their common use to solve supervised classification problems with continuous features:

Classification Model	Details
SVM	linear, radial basis, and signmoid kernels
Naive bayes	gaussian prior
Linear Regression	Huber loss function
Multinomial Logistic Regression	one vs. rest scheme with L2 norm
K Nearest Neighbors classifiers	uniform and inverse distance neighbor weights

While we experimented with using Radial Basis Function kernel SVMs, Sigmoid kernel SVMs, and Huber Linear Regression, we did not have the opportunity to tune the parameters of these models. Thus, they are not included in the remainder of this report.

All the algorithms we used were implemented in python with the Scikit-Learn package.

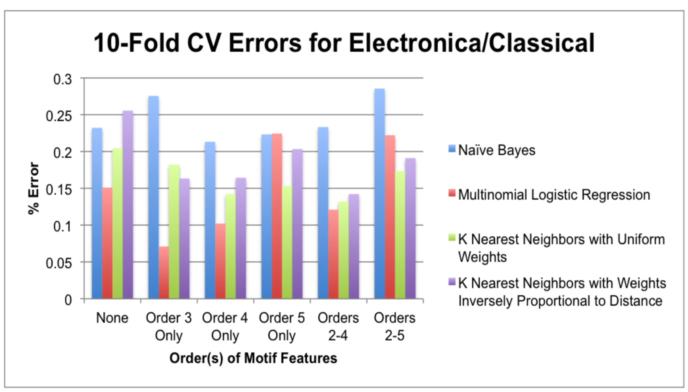
3 Data

Our data-set consisted of two hundred samples. We had 40 samples from each of five genres: classical, electronic, country, jazz, and opera. The genre label assigned to each song in our training sets was determined by the genre assigned by the MIDI source [4] [6] [7] [8] [9] [10] [11]. To ensure each sample contained enough data to generate a structurally unique graph, we only selected files with a minimum size of 3KB. The files can be found in our cloud storage folder [5].

4 Results

4.1 Feature Selection

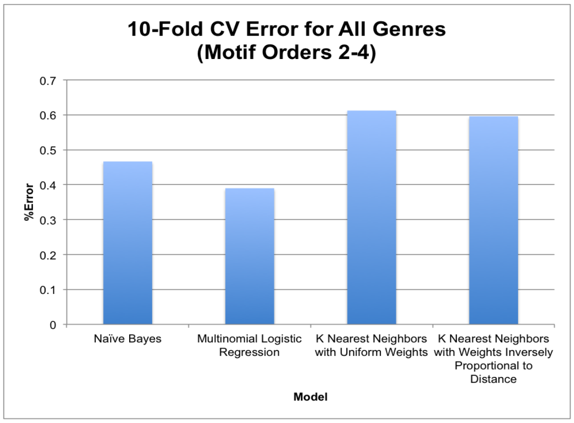
To best understand the uses of graph representations of music in machine learning, we dove deeper on which graph properties produced the most relevant information for genre classification; first, we examined the weights associated with each feature in our feature vector from our linear SVM model. This analysis showed that certain features added no value (such as the denominator of the time signature and the number of parts), while others contained significant importance (including the numerator of the time signature and all of the remaining graph properties and motif properties).



Another key part of our feature selection process had to do with analyzing the relevance and effectiveness of different orders of motifs. We ran our algorithms on 6 variations of our feature vector (see graph above), in order to determine which motif orders–if any–provided the most useful information for genre classification.

We found that our best results across various algorithms came from the feature vector with motifs of orders 2-4. The omission of motifs of order 5 improved performance most probably because there are such a large number of motifs of order 5, and they are so specific, they provide more information on the song than the genre level.

4.2 Machine Learning



Binary Problem	% Error
Electronica vs Classical	0.13
Electronica vs Opera	0.10
Electronica vs Country	0.098
Electronica vs Jazz	0.066
Classical vs Opera	0.31
Classical vs Country	0.18
Classical vs Jazz	0.31
Opera vs Country	0.25
Opera vs Jazz	0.34
Country vs Jazz	0.39

% Error for Multinomial Logistic Regression using motif orders 2-4

Our highest performing models were Naive Bayes, Multinomial Logistic Regression, and K Nearest Neighbors (both with uniform weights, in which all points in a neighborhood are weighted equally, and with weights inversely proportional to distance). Across all classification problems, Multinomial Logistic Regression consistently had the lowest error.

Our SVM with a linear kernel also performed relatively well, but unfortunately, due to a lack of computational power, we were only able to run the SVM on the 5-way classification with features including motifs 2-3. Even so, the linear kernel SVM performed extremely well on this classification, outperforming Multinomial Logistic Regression by roughly 5%.

Interestingly, our results also give an informal graph-similarity measure between genres, as binary classification problems between similar genres yeilded higher errors. We see that Classical, Opera, and Jazz seem to be relatively similar in graph structure, and that Country is also surprisingly similar to Jazz.

5 Discussion and Future Work

Since the focus of our project was to determine whether graph representations of music could be used to effectively classify songs by genre, our data serves to prove that they can. In our binary classification tests, algorithms performed comparably to or better than the algorithms implemented on feature vectors comprised of other types of "musical fingerprints" outlined in the Background section of this paper, with binary classification errors in the 10-30% range.

The next steps to improve our results would be to continue to fine tune the parameters for each machine learning algorithm. Additionally, it would be useful to expand the types of graph properties we use as features. For example, we would like to try our models using the counts of connected non-isomorphic sub-graphs (instead of motifs, which lose information since they are induced). Additionally, there are features of the graphs such as edge direction and weight that we neglect entirely, which could be used to derive greater structural meaning from the graphs.

6 Acknowledgements

We would like to thank Panos Achlioptas for introducing us to the idea of graphical representations of music and for his guidance throughout our work on this project.

References

[1] Panos Achlioptas. Musical graphs. arXiv 2014.

[2] Roberto Basili, Alfredo Serafini, and Armando Stellato. Classification of musical genre: A machine learning approach. 2004.

[3] Michael Haggblade, Yang Hong, and Kenny Kao. Music genre classification. 2011.

[4] Country Midi Palace http://countrymidipalace.tripod.com/index.html. Accessed: 20 November, 2014.

[5] Data Repository https://stanford.box.com/s/nylud5x9earmj0ro5mzh.

[6] Aria Database http://www.aria database.com. Accessed: 20 November, 2014.

[7] Classical Archives http://www.classicalarchives.com/. Accessed: 11 November, 2014.

[8] Cool Midi http://www.cool-midi.com/electronica midi.php. Accessed: 11 November, 2014.

[9] Download-Midi http://www.download midi.com/. Accessed: 11 November, 2014.

[10] Glass Pages http://www.glasspages.org/audio.html. Accessed: 11 November, 2014.

[11] The Jazz Page http://www.thejazzpage.de/index1.html. Accessed: 20 November, 2014.

[12] Nicola Orio and Antonio Roda. A measure of melodic similarity based on a graph representation of the music structure. 2009.