Classical Music Time Period Classification Using Machine Learning Algorithms

Arturo P. Caronongan III
College of Computer Studes
De La Salle University - Manila
2401 Taft Avenue
Manila, Philippines
arturocaronongan@yahoo.com

Kenston C. Choi
College of Computer Studies
De La Salle University - Manila
2401 Taft Avenue
Manila, Philippines
kenston11@gmail.com

ABSTRACT
In this paper, we describe about a classification problem which will classify a MIDI file consisting of piano solo pieces as whether they fall under the Baroque, Classical, Romantic, or Impressionism Era. This paper will discuss the algorithms and approaches used in classifying the MIDI files to their respective classifications.

Keywords
Machine Learning, Music, Classification, Artificial Intelligence, Musical Instrument Digital Interface

1. INTRODUCTION
Classification is a field in Machine Learning that has several applications for several purposes. These purposes may be for classifying an animal as a mammal or not, classifying a movie as horror, action, or not, and several other fields which requires identifying the classification of an object given a set of attributes. Music classification is among the most common classification problems being solved [4].

Several studies with regards to music classification have already been undertaken, and are in the process of undergoing further research. Classification with regards to the musical genre of a respective music file has been undertaken which classifies if a piece of music is pop, rock, blues, country, or classical. An example of a study that involved the classification of music genre is the classification of music genre through a multilinear approach where classification is performed by a Support Vector Machine [6].

While previous experiments with regards to music classification have already been undertaken, the difference between this research and previous researches is this research aims to study a deeper genre, classical pieces in particular [2]. It is often satisfactory to classify a piece of music as "Classical" given a specific set of attributes. However, the question may be asked as "what kind of classical piece is it?" The answer may range as one of the following periods of music: Baroque, Classical, Romantic, Impressionism, and eventually the Modern Era [10].

This research aims to provide a more thorough classification of classical music with regards to their time period as successfully identifying if the piece was composed from the Baroque Era, the Classical Era, the Romantic Era, or the Impressionism Era.

2. CLASSICAL MUSIC ERAS
The baroque period saw the development of functional tonality. During the period, composers and performers used more elaborate musical ornamentation, made changes in musical notation, and developed new instrumental playing techniques. Baroque music expanded the size, range, and complexity of instrumental performance. Notable composers during this period were George Frederic Handel, Johan Pachelbel, and Johan Sebastian Bach [8].

The classical era had a noticeable difference. Classical music has a lighter, clearer texture than Baroque music and is less complex. It is mainly homophonic. The variety and contrast within a piece became more pronounced than before, as do the variety of keys, melodies, rhythms and dynamics along with frequent changes of mood and timbre were more common. Notable composers during the period were Franz Joseph Haydn, Wolfgang Amadeus Mozart, and Ludwig Van Beethoven [8].

The composers of the Romantic period sought to fuse the large structural harmonic planning demonstrated by earlier masters such as Haydn and Mozart with further chromatic innovations, in order to achieve greater fluidity and contrast, and to meet the needs of longer works or serve the expression that struggled to emerge. Chromaticism grew more varied, as did dissonances and their resolution. Some notable composers from the Romantic Era were Frederic Chopin, Franz Liszt, and Johannes Brahms [8].

It was during the Impressionism when composers focused on suggestion and atmosphere rather than strong emotion or the depiction of a story. Musical Impressionism occurred as a reaction to the excesses of the Romantic era. While this era was characterized by a dramatic use of the major and minor scale system, Impressionist music tends to make more use of dissonance and more uncommon scales such as the whole tone scale. Notable composers during the Impressionism period were Claude Debussy, Maurice Ravel, and Erik Satie [10].

3. DATA GATHERING
MIDI files were manually chosen and retrieved from the Internet. Each time period has 40 MIDI files giving a total of 160 MIDI files in the entire dataset. In the due process of selecting MIDI files to include in the data set, a careful listening to the MIDI files was also done as there were some MIDI files that were poorly created or did not sound similar to the actual piano piece being played. This was to assure that the research would make use of non-noisy data and include only the valid MIDI files.

The MIDI files served as the training data. They were loaded into the jSymbolic program [3], a MIDI feature extraction tool written

in Java. jSymbolic generated XML files containing the features and values of each MIDI file. It was through the XML files that they were converted into CSV files to be fed in to the classification system. The resulting CSV file contained 103 attributes.

3.1 Manual Feature Selection

As jSymbolic extracted features from the MIDI files, there were several attributes which proved to be unimportant for the research, as the pieces the MIDI files represented were mainly piano solo pieces. Due to this, a manual feature selection was done with the attributes without any feature selection algorithm. The attributes that were eliminated were the features which remained constant or nearly constant throughout all the data sets which proved to be absent in a MIDI file composed of only the acoustic piano (or harpsichord for Baroque era pieces).

Also, some attributes (i.e. range) have been kept due to the familiar knowledge with regards to the difference between the pieces from the different era. The pianos used during the baroque and classical era were different from the pianos that were used during the Romantic and Impressionism era as the former had a more limited range than the latter. Thus, there is a high certainty that certain sets of notes will not appear in Classical and Baroque.

Whilst the features have been manually selected, it should be noted that the original data set was maintained for comparison purposes. Through the manual feature selection set, the following attributes were reached: (1) range, (2) variation of dynamics in each voice, (3) size of melodic arcs, (4) pitch variety, (5) average melodic interval, (6) combined strength of two strongest rhythmic pulses, (7) stepwise motion, (8) variation of dynamics, (9) rhythmic variability, (10) most common melodic interval prevalence, (11) most common pitch class prevalence, (12) dominant spread, (13) melodic thirds, (14) voice equality-number of notes, (15) variability of time between attacks, (16) voice equality-melodic leaps, (17) most common melodic interval, (18) number of common melodic intervals, (19) strongest rhythmic pulse, (20) strong tonal centers, and (21) chromatic motion.

3.2 Training with the Dataset

As the research progressed, data was added to the data set to boost the training process. However, it was noted that there were some pieces that acted as a "bridge" between two musical eras. The most distinct examples are the pieces by Ludwig Van Beethoven as he acted as the bridge from the Classical era to the Romantic era.

It was this reason that Beethoven's pieces were left out as part of the training data in the experiments conducted as MIDI files of his pieces confused the system as to what a Classical and a Romantic piece really are.

4. METHODOLOGY

The research made use of University of Waikato's Weka v. 3.6.3. Weka [1], an open source Java program, is a collection of machine learning algorithms for data mining and classification tasks. The CSV datasets used for training and for testing were manually fed to Weka and were executed with the desired algorithms. The results were then analyzed and interpreted accordingly.

Four feature sets were used. The first set took all features jSymbolic gave, and CFS Subset algorithm was used over the first set to produce the CFS feature set. The manually picked features were also used and ranked. The first fourteen features based on the manually picked and ranked features comprised the fourth feature set.

4.1 Using MIDI Files as a Whole

The first phase of the experiment made use of each MIDI file as a whole without splitting it into several pieces. The features of these MIDI files were extracted using jSymbolic and the results were fed to Weka.

Ten folds cross-validation was used during testing using mostly KNN and J48 to classify the instances. Relevant results are further tested using the same approach but over multiple runs with different incremental random seed values to randomize the dataset. This ruled out the possibility of the order of training data as a factor that increases the accuracy.

4.2 Using Split MIDI Files

In the second phase of the research, the MIDI files were split into several pieces which increased the number of instances in dataset. Two split modes and two separate experiments are discussed in the next subsections.

4.2.1 10-piece and 10-second Split Modes

Two split modes were tried. A MIDI file was split into 10 pieces of equal duration, and for another separate experiment, a MIDI file was split into pieces consisting of 10 seconds each. MidiCut [5], a DOS program, was used to split the MIDI files by giving the start and end time of the split piece. MidiTime [5] was used to determine the total song time in MIDI time units which is more precise than seconds and is related to beats and measures.

The researchers created a program that batch processes the retrieval of the duration, determining the start and end time units of each piece, and splitting the MIDI files based on the parameters. Feature extraction was then done on the split pieces using jSymbolic.

The 10-second split dataset containing 4368 instances was 2.7 times bigger than the one in the 10-piece split containing only 1600 instances. The first and the last piece of every musical piece were removed, based on the observation that some musical pieces may have similar beginning and ending. A quick evaluation of the two split modes was done to choose which is likely better. Each musical piece might not have the equal number of split-pieces in the 10-second split dataset, since some musical pieces were longer. This somehow created an imbalance in the number of pieces per classification group, which did not happen in the 10-piece split mode.

The split pieces and the non-split Whole music dataset were used as training and test set, respectively. The 10-piece split was able to achieve 90.76% accuracy while the 10-second split only reached 86.92%. The 10-second split performed slower since it had more instances. The values in the features of split pieces of a particular musical piece when averaged together may at some point be closed enough to the same musical piece in the Whole dataset version, thus a high accuracy is expected for this initial evaluation of the split modes. It also showed that splitting a musical piece into 10 seconds per split-piece only made it appear to belong to another different group of split pieces. Determining the right number of split-pieces is important. Splitting it based on the number of pieces appeared safer, as the duration of each split-

piece is relative to the duration of the whole musical piece. Thus, the 10-piece split dataset was given more focus.

4.2.2 Majority Voting and Standard-deviation Consolidation Experiments

Two separate experiments were also performed for both the 10-second and 10-piece split datasets. The first involved consolidating the values of the features of each split piece. This was done by computing the standard deviation and the mean of the values for a particular feature of a split piece. Any value that was outside the range of the mean ± the standard deviation was removed. One, two, and three standard deviations were tried, but two standard deviations were used.

The consolidated value for the feature was computed by averaging the remaining values; thus, the split pieces were consolidated to represent one musical piece. This approach was an attempt to rule out those pieces that may contain some features that do not conform to the other split members. The final dataset contained consolidated pieces where the number of instances was similar to the non-split Whole dataset. Cross-validation was then performed over the entire dataset.

The second experiment involved using all split-pieces of a particular musical piece as a single test set against a training set consisting of other split-pieces. The majority of the classification assigned to the test split-pieces was taken as the classification of the music title. This means that if 75% of the split-pieces of a musical piece are classified as Baroque, and 25% are classified as Classical, the final classification of the musical piece is Baroque.

However, ties could also happen, in which the researchers tried two separate approaches in determining the final classification of the musical piece. The first approach averaged the values of the features of the split-pieces of the musical piece to form one piece that will be used as a single test case in determining the classification. The second approach retrieved the feature values of the same song in the non-split Whole dataset, and used the instance instead.

Like the first phase, ten-fold cross-validation was used during testing, using mostly KNN and J48 to classify the instances. The use of different feature sets was also tried. Relevant results were further tested using the same approach but over multiple runs with different incremental random seed values to randomize the dataset.

5. RESULTS AND ANALYSIS

After the experiments were conducted, results were gathered and analyzed respectively. The results using K-Nearest Neighbor in both phases are shown in Table 1.

Table 1: Results of K-Nearest Neighbor (K=1)

Features Set	Whole Dataset	Standard-deviation Consolidated		Majority Voting	
		8-Piece	10-Piece	8-Piece	10-Piece
All	66.88%	72.25%	70.25%	72.25%	72.38%
CFS Subset	68.12%	78.13%	71.88%	76.00%	76.38%
Manual	78.75%	73.75%	73.75%	75.62%	76.00%
Trimmed Manual	72.50%	74.38%	76.70%	77.88%	77.00%

Using the entire data set without filtering any attributes, the classification only yielded an accuracy rating of 66.88%, which brought the conclusion that the classification had a hard time classifying the pieces to their respective eras because of the amount of attributes. While there were definitely some attributes that differed between each pieces, there were several others that did not; thus, the data set requires feature selection.

Using CFS as feature selection algorithm, the accuracy improved the classification process. We tried using different values for *k* in KNN, but the value of 1 generally gives the highest accuracy. This was because the items in the dataset and the feature values were distinct. Although musical pieces were composed by the same composer, the source of our MIDI files did not necessarily come from the same person who encoded them as MIDI formats.

The accuracy of the pieces as a whole did not produce fairly accurate results, with the exception of using the manually chosen features, resulting in an average of 78.75% accuracy rating. Through the manual feature selection, the attributes that were selected managed to bring a striking resemblance with the pieces grouped together to enable such classification. However, it should be noted that there were several pieces which had a very long duration and contained several passages which could be classified as a Classical Piece passage or a Romantic Piece passage due to some composers during those era paying homage to their predecessors from previous eras. The reason for the decrease in the trimmed manual (first fourteen of the manually chosen features) feature set was that while there were indeed improvements shown in feature selection, there were indeed some attributes which were very important in differentiating between the different eras. It was possible that one feature could have made an impact when it comes to classifying using Instance Based learning.

Using the standard deviation consolidated dataset, and CFS subset feature set, the accuracy hit 78.13% in the 8-piece dataset. However, the accuracy was only 71.88% in the 10-piece dataset. This showed the possibility that the removal of the first and the last pieces, before computing the standard deviation that led to exclusion of certain feature values could help increase accuracy. Therefore, the first and the last pieces could be seen as probable noise where these pieces shared similarities with other pieces from other classes. The average of the results in 8-piece split was slightly higher compared to the 10-piece split dataset even in most of our experiments and feature sets.

For the majority voting using KNN and the Trimmed Manual feature set, the accuracy reached 77.88%. It showed that features selected play a crucial part in improving the accuracy. Taking the Trimmed Manual feature set, the approach in resolving the ties for majority voting is summarized below in Table 2.

Table 2: Approaching Tie Breaks

Tie-break Approach	8-Split Accuracy (8.38% ties)	10-Split Accuracy (6.5% ties)
Average	77.88%	77.00%
Get from Whole	74.13%	75.50%
Isolate Ties	80.91%	80.79%

The approach that took instances from the Whole dataset to resolve ties only decreased the accuracy. This was perhaps those musical pieces with ties were the transitional period pieces, which when used with the features from the Whole set might still not be classified correctly.

There were fewer ties using the 10-piece split mode since there were more pieces that were used in voting; however, this should

not be translated to giving correct votes, as they might break the ties in favor of the wrong classification as shown in the small difference between the two split modes. Isolating the ties without classifying the items that have tied votes resulted to a higher accuracy, reaching the 80% mark. This was because some musical pieces are composed during the transition period from an earlier to a later one or vice versa.

We were able to list some of the musical pieces which have tied votes. Claude Debussy's pieces and Franz Liszt's pieces in particular had a number of pieces which were tied in between being a Romantic piece and an Impressionism piece. This is evident as some of Debussy's pieces and Liszt's later pieces had a similarity in structure, with Liszt's later pieces (i.e. Nuages Gris) focusing on painting tragic images, which the Impressionism era is mainly accomplished for. Liszt and Debussy served as transitions from one era to another thus it is concluded that the classifier had a hard time distinguishing one era from another given their pieces. An even better approach of resolving tie votes can improve accuracy, and using weighted votes may even be taken into consideration.

It appeared that the Trimmed Manual feature set gave the slightly higher accuracy with an average of 75.69% across different experiments as compared to including all features and using CFS Subset feature selection which gave an accuracy of 70.80% and 74.10%, respectively. Majority voting gave the highest result across different feature sets with an average accuracy of 75.44%, followed by the STD approach with an average of 74.63%. The Whole (no-split) dataset gave an average accuracy of 71.56%, but it should be noted that it gained the highest accuracy of 78.75% in one experiment using the Manual feature set; however, this was not as consistent when used with other feature sets that resulted to 66.88% accuracy. We saw that Majority Voting approach was consistent, where the accuracy value increased from 72% using all features to 77% using the trimmed manual. This indicated that split pieces that casted their votes could more or less overrule the pieces that tend to be misclassified. Without feature selection, the split approach could already give an accuracy of 72% as compared to the 66% in the Whole dataset.

C4.5 decision tree algorithm was also used. Table 3 shows the results.

Table 3: J48 results

Features Sets	Whole Dataset	Standard-deviation Consolidated		Majority Voting	
		8-Piece	10-Piece	8-Piece	10-Piece
All	62.5%	63.13%	60.55%	67.50%	68.25%
CFS Subset	70.63%	65.38%	61.25%	74.86%	75.63%
Manual	62.50%	64.38%	67.77%	74.50%	74.75%
Trimmed Manual	61.25%	63.13%	65.55%	72.75%	73.00%

The gaps in terms of accuracy between KNN and J48 were getting closer. We see the use of more sophisticated learning approaches that will replace lazy learning. Majority voting approach performed better than the two other approaches in this algorithm. This could be attributed to the many pieces being evaluated by the decision tree, as compared to having only a single musical piece going through the decision tree. The number of ties was fewer using J48 as compared to using KNN by more or less 3%. Features selected by CFS Subset generally gave a higher accuracy compared to those manually chosen features in this algorithm. Feature selection certainly improved the accuracy, considering that the number of features was reduced by more than 50% after feature selections.

In a single KNN (k=1) cross-validation run using the entire 10-piece split dataset, the accuracy reached as high as 92.03%. This was because of the split pieces of specific musical pieces were related in certain ways, and cross-validation in Weka did not assure that all those split pieces are bounded together either in the training or in the test set. But this also showed that the split-pieces are somehow able to join to their proper classes. Running it using KNN with k=12, which was two more than the number of split pieces, resulted to an accuracy of 85.86%. However, we did not highlight this approach, since the above methodologies are much acceptable. This also means that more data is needed to improve the accuracy. Splitting the musical pieces is an attempt to increase the instances in the dataset.

Revisiting the comparison between 10-second and 10-piece split modes, we only used KNN and compared the difference between the two. Both dataset excluded the first and the last pieces. We isolated the results of this experiment for simplicity. For the 10-piece split mode, we copied the previous results under the 8-piece columns in Table 1. The result is shown in Table 4.

Table 4: 10-second vs. 10-piece Split Modes in KNN

Features Set	Standard-deviation Consolidated		Majority Voting	
	10-sec	10-pc	10-sec	10-pc
All	72.50%	72.25%	70.63%	72.25%
CFS Subset	77.50%	78.13%	66.50%	76.00%
Manual	73.13%	73.75%	70.63%	75.62%
Trimmed Manual	77.50%	74.38%	70.62%	77.88%

The 10-second split mode needed more time and space to split and process. We expected that 10-second split mode would not be good when tried with the Majority Voting experiment because there was a higher chance that a 10-second piece might appear too similar to other pieces in another era. This issue did not occur in the standard-deviation consolidation experiment because those 10-second pieces that were affected by that problem would more likely be discarded and the average values would reflect most of the remaining majority pieces. Thus, the 10-second split mode had an advantage of 0.53% on average to as high as 3.12% in certain cases as compared to 10-piece split mode. However, the accuracy reached by using the 10-piece split mode was still higher by 0.38% in the Majority Voting experiment.

6. CONCLUSIONS

With the experiment's scope, the KNN performs better than the J48 by an average of 6% in most experiments and as high as 10% in some cases. On average, splitting the MIDI files works more effectively across different feature sets than not splitting them by at most 4% using KNN and by at most 8% using J48. Removing the first and last split-piece (8 pieces in 10-piece split mode) has a very slight advantage by a small margin of 0.30% on average to as high as 1.48%. We recommend manual selection and testing of features that suits the given classification problem that considers the strong features of each era.

Under the 10-piece split mode, majority voting approach is preferred over the standard-deviation consolidation approach as it performs better by an average of 5.15%. The 10-second split

appears to be more accurate when done using the Standard-deviation consolidation approach. 10-piece split is preferred over 10-second split because of the time and space requirements in processing the files.

The experiment was able to achieve the highest accuracy of 78.75% using the Whole dataset, KNN, and Manual feature sets, but more data and further testing can show if this particular experiment configuration will maintain its position. This is followed by the 78.13% mark using 8 pieces of the 10-piece Split Standard-deviation Consolidated dataset, KNN, and CFS Subset feature set, and the 77.88% mark using 8 pieces of the 10-piece Split Majority Voting dataset, KNN, and Trimmed Manual feature set. The inconsistency of the Whole dataset using other learning algorithm and feature set gave us an impression that it may be easily affected by the addition of more data. We think that the Majority approach using KNN and 8 pieces of the 10-piece split data will produce a better result due to the consistency of the results across feature sets as well as registering the highest average accuracy as compared to the other approaches or split modes. We also see the 10-second split dataset used with the Standard-Deviation Consolidation approach as an alternative when time and space requirements are not an issue.

We realized that there is a need for more data for us to see a clearer trend and establish a stronger conclusion. This is evident as we are dealing with many factors that affect the results. Under the scope of our study, we have reduced the issue to the pairing of the feature set and the dataset to be used as other dependent factors like the choice of learning algorithms, strengths between standard-deviation or majority voting when applied with different datasets, and the advantage of removing the first and last pieces have been more or less established.

Using MP3 files or wave files for the classification of classical pieces instead of the symbolical nature in MIDI can also be tried. Making use of hybrid approaches by including standard deviation with majority voting can also be used. There is also a need to pay close dividends to transitional pieces that should not be used in the training set to confuse the classification but this should not limit the need for more data sets to provide a more accurate classification. The experiment is headed to the right track and more feasible experiments in the future related to classification should eventually be able to reach the proper classification model needed.

7. ACKNOWLEDGMENTS

Our thanks to Cory McKay of McGill University for developing jSymbolic and for answering our questions related to the software.

8. REFERENCES

[1] Hall, M., Frank, E., Holmes G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The Weka Data Mining Software. Retrieved from http://www.cs.waikato.ac.nz/ml/weka/

[2] McKay, C. (2004). Automatic Genre Classification of MIDI Recordings. McGill University, Montreal.

[3] McKay C. & Fujinaga I. (2006). jSymbolic: A Feature Extractor for MIDI Files. *Proceedings of the International Computer Music Conference*. 302-5.

[4] McKay, C. (2010). Automatic Music Classification with jMIR. *Ph.D. Thesis*. McGill University, Canada.

[5] Nagler, G. (1997). Freeware MIDI Software. Retrieved from http://www.gnmidi.com/gnfreeen.htm

[6] Panagakis I, Benetos E, and Kotropoulos C. (2008). Music Genre Classification: A Multilinear Approach.

[7] Peyser, J. (1971). The New Music, the Sense behind the Sound. Dell Publishing Co., Inc., New York, pp. 10-11.

[8] Swann, J. (2004). Classical and Romantic Music. Retrieved from: http://trumpet.sdsu.edu/M151/Romantic_Music1.html

[9] Swann, J. (2004). 20th Century Music. Retrieved from http://trumpet.sdsu.edu/M151/20th_Cen_Mus1.html