Specifying Melodic Features for Classical and Non-Classical Melody Evaluation

Andrei D. Coronel
Ateneo de Manila University
acoronel@ateneo.edu

Ariel A. Maguyon
Ateneo de Manila University
amaguyon@ateneo.edu

Proceso L. Fernandez
Ateneo de Manila University
pfernandez@ateneo.edu

ABSTRACT
Algorithmic Composition for music is a progressive field of study. The success of an automated music-generating algorithm depends heavily on the fitness function that is used to score the generated music. This fitness function is generally based on music features that a given algorithm is programmed to measure. This study explores features that are important for melody generation by investigating those that can 1) separate classical from non-classical music based on melody, and 2) help distinguish between two specific subgenres of both classical and non-classical music – Chopin vs. Bach, and jazz vs. contemporary compositions. The jSymbolic tool was used to collect 160 standard features from 400 music files. C4.5 was then used to select significant features. A comparative analysis between the feature sets suggested by the C4.5 algorithm and suggested in a previous study of Towsey et al. was performed by running Naïve-Bayes and SVM classifiers on each feature set. The results show that the features that have been identified in this study are better able to classify classical from non-classical music. These features may, therefore, be considered when formulating melody-based fitness functions for automated classical music generation.

1. INTRODUCTION
Computer-generated music is a progressive area of interest. Interdisciplinary works have developed various means to create musical compositions via algorithmic programming and software applications. Computer-generated music can suggest either tool-based computer-aided compositions or algorithmic composition. The former requires users to employ non-algorithmic interactive software to be able to create music from a composer's idea. The Algorithmic Composition (AC) can involve either an application of heuristic principles, automated learning techniques, or evolutionary programming. This study refers to AC methods rather than tool-based methods.

The AC methods generally fall under one of two types: rule-based or evolutionary. The quality of the output of AC, especially in evolutionary methods such as Genetic Algorithms, relies heavily on a suitable fitness function. This function is used to score the computer-generated music by providing a quantitative rather than a qualitative measure of aesthetic value. It is this function, therefore, that determines whether or not the musical output has acceptable aesthetic quality.

To arrive at a suitable fitness function, it is critical to identify the important music features to be measured. One of the challenges, however, is that different music genres intuitively have different fitness functions. This study investigates various music features that can be used in the computer generation of classical music. It is shown that a set of nine (9) features is sufficient in accurately distinguishing between classical and non-classical music. These features, easily captured from MIDI files using jSymbolic, can be explored in fitness functions for AC methods that generate classical music.

2. REVIEW OF RELATED LITERATURE
There are many software applications and generative music systems for Algorithmic Composition (AC) available on the Web. These include IMPROVISOR, Tune Smithy, and Bloom. The main methods used by these applications generally fall under one of two categories: rule-based or evolutionary. Examples of rule-based AC methods include stochastic binary subdivision, auditals, key phrase animation, fractal interpolation, and expert novice pickers. These techniques involve the application of rhythmic generation to melodies, grammar-based melodic generation, computer graphics algorithms applied to music, and the use of knowledge-bases, respectively [6].

Evolutionary AC methods are those that basically run the *generate-evaluate-repeat* loop for music generation. Two of the most popular heuristics for evolutionary AC are Genetic Algorithm and Genetic Programming [10]. These methods rely on a fitness function for evaluating the generated music.

Various fitness functions have been previously explored and applied to different AC methods. There is no recognized gold standard yet, as even modern studies in melodic extension may still partially employ human evaluators [4], or fully utilize them to score improvisations [5]. Included in the list of these scoring mechanisms are human critics, rule-based critics, learning-based critics, and global statistics. Since there are constraints when human critics are used, e.g., fatigue based on the repetitiveness of assessing the fitness of pieces of music, the search for more automatic evaluation techniques is a continuing study [11].

A prerequisite for the development of a fitness function for automatic music evaluation is the extraction of features from a music piece. These feature values become the parameters to the fitness functions being developed. Research on feature extraction straight from audio or acoustic signals has already been done [9]. The fitness function developed for this took into consideration music features such as spectral variation, count of sound, frequency strength, amplitude frequency, to name a few. The study focused on pairing each extracted music feature with appropriate weights, where the weight set is genre-specific. This type of music analysis, however, does not isolate melody, since the features were extracted from audio signals from compressed music.

Melodic analysis is especially relevant for evolutionary methods since melody is the factor being evolved in every turn of the iterative program. A study made by Towsey et al. enumerated, categorized, and analyzed features based on melody by applying global statistics to a dataset of MIDI files. The study was able to identify 21 melodic features that are useful for melodic analysis. These features include pitch variety, dissonance of intervals, and contour direction, among others. PCA analysis and subsequent clustering procedures were successful in identifying the strength of influence of the features to the potential fitness rating of melodies [11]. These features are henceforth referred to as Towsey melodic features in this paper.

A more recent study by Freitas et al. enumerated and described important features for melodic evaluation, taking previous studies in consideration [2]. What these studies have not yet addressed, however, is the identification of specific subsets of features from the available feature space that may be used for melodic analysis involving particular genres. It is not hard to imagine that the features relevant to jazz melody evaluation may not be exactly the same for the classical music genre.

In this study, we extend the work of Towsey et al. by verifying if the 21 melodic features that they have identified can optimally differentiate classical from non-classical genres, and between 2 specified subgenres for each of these. Identification of the important features is a step towards the development of a better fitness function for the automated evaluation of music based on melodic features.

The results of this study can contribute to studies concerning the creation of fitness functions for evaluating melodies generated by evolutionary algorithms. Fitness functions are crucial to automated music composition since the latter requires a quantitative scheme for melodic scoring. The construction of such a fitness function begins with determining the key melodic features specific for each music genre.

3. METHODOLOGY

The goal of the study is to identify melodic features that may be used for effectively classifying music according to genre and subgenre in the context of melodic analysis. Several feature sets are identified, and these feature sets are validated by comparing the classification accuracies produced in running SVM and Naïve Bayes on each of these feature sets.

There have been numerous works on music evaluation where the music source is either a dataset of audio signals (compressed music files) or MIDI. This study uses the latter, since melodic analysis is the focus of the study rather than the actual acoustic structure.

It is not the goal of this study to develop and implement a new algorithmic composition technique. However, the identification of important melodic features is an essential step towards developing a fitness function that could potentially evaluate melodies according to their quality in the context of genre matching.

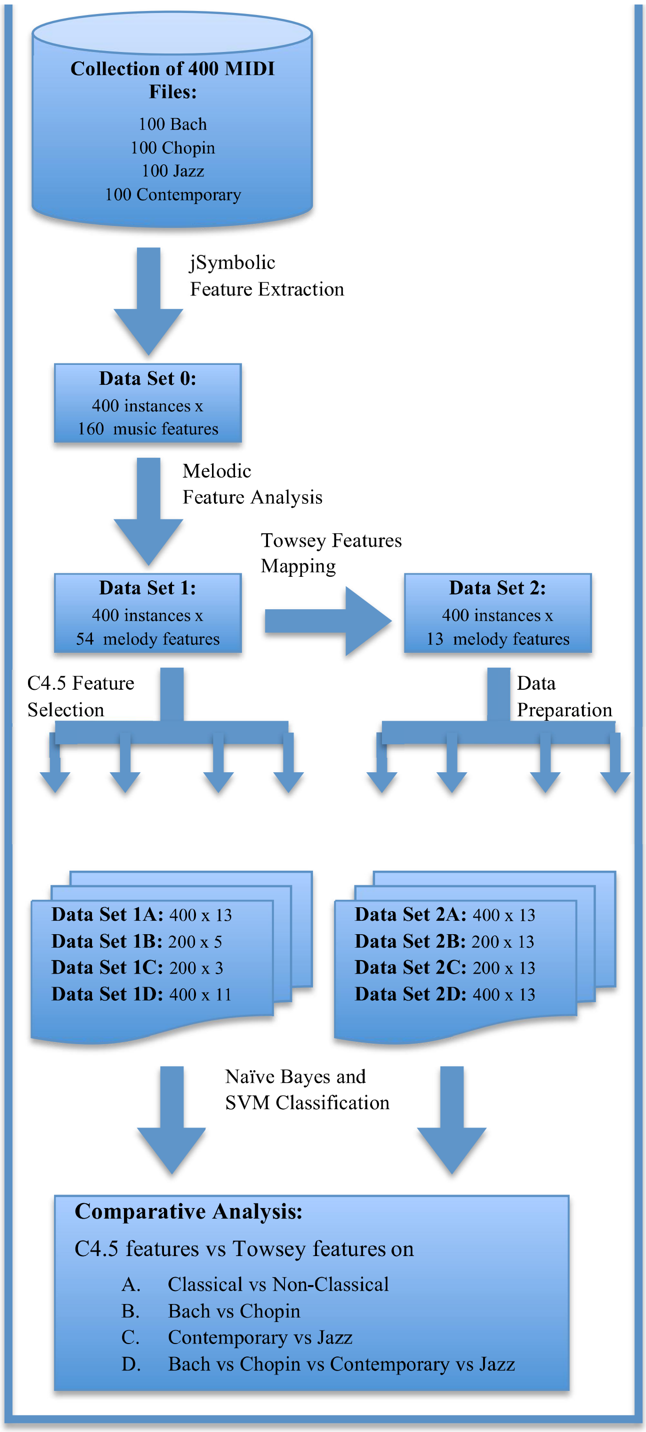

Figure 1: Methodology in this research

The methodology involved several steps (see Fig. 1).

1. Acquire a music dataset (in MIDI format) with a balanced distribution of 4 subgenres: Bach, Chopin, Jazz (various artists), and Contemporary (Beatles). Bach and Chopin represent the classical group, while the Jazz and Contemporary melodies represent the non-classical group. The Beatles were chosen to represent contemporary music as majority of their melodies stay within whole major and minor non-augmented chords – a harmonic trait of common contemporary music still applicable today.
2. Extract features from the music dataset (via jSymbolic). This involves all 160 features that are extractable from each of the MIDI files.
3. Use the jSymbolic documentation in order to identify the features related to melody. Based on feature definitions, only 54 out of the 160 features may actually be applied to melodic analysis. Hence, the extraneous 106 features were removed from the dataset.
4. Apply C4.5 decision tree algorithm to the 4 datasets to determine which of the 54 melodic features are significant for specific classification. The following describes the 4 datasets:
    a. Dataset of 400 MIDI files, each labeled as either classical or non-classical.
    b. Dataset of 200 MIDI files, each labeled as either Bach or Chopin
    c. Dataset of 200 MIDI files, each labeled as either Beatles (contemporary) or Jazz.
    d. Dataset of 400 MIDI files each labeled as either Bach, Chopin, Beatles or Jazz.

    Note that each of the four datasets described above is a subset of the earlier dataset consisting of 400 MIDI files, each with 160 feature values (refer to Steps 1-2). The C4.5 algorithm generated a decision tree whose set of nodes correspond to a subset of the 54 feature values enumerated per song in the datasets. This subset explicitly excludes the features that are not relevant to the specific classification challenge, hence reducing the number of melodic features important to the specific classification task.
5. Create 4 new music datasets with reduced number of features, as recommended by the C4.5 results of the previous step.
6. Map the Towsey melodic features to the jSymbolic-extractable features. Mapping between Towsey features and jSymbolic features is based explicitly on feature definitions.
7. Create additional 4 music datasets similar to Step 5, but this time using the Towsey-mapped features of Step 6.
8. Run Naïve Bayes and SVM on the 8 datasets from Steps 5 & 7, and estimate the accuracy using 10-fold cross-validation method. Accuracy here is measured by computing the number of correct classification divided by the total number of instances classified.
9. Determine the features involved in the best results for each specific classification challenge. This step compares the classification accuracy of feature-sets recommended by the C4.5 algorithm vis-a-vis the Towsey-recommended features. Specifying which melodic features are relevant per classification challenge is done by choosing the feature set with the higher classification accuracy.

4. RESULTS AND ANALYSIS

After subjecting a set of 400 MIDI files to the jSymbolic processing, it was noted that key features used for genre classification included ones not exclusive to melodic analysis. Some examples of these are percussion-based features, analysis of MIDI layers/voices, and specific instrument fractions. These features are not useful since subsequent work after this study will involve evaluation of evolved melodies. Selecting features exclusive to melodic analysis was performed to address this. Based on jSymbolic documentation regarding feature definitions, 54 of the available 160 music features were identified to be related to melody.

After identifying the 54 melody-related features, C4.5 was used to help identify which of these features are important in relation to a specific melody classification challenge. C4.5 is an algorithm that generates a decision tree representing rules for classifying instances of a given data set into 1 of several classes. A sample decision tree resulting from executing C4.5 in one of the classification challenges in this study is shown in Fig. 2.

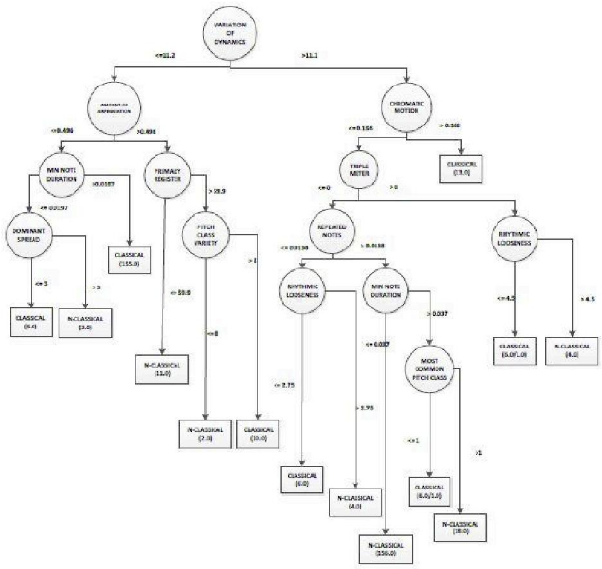

Figure 2: C4.5 Decision Tree result for classifying the 400 MIDI melodies into classical and non-classical music.

In a C4.5 classification decision tree, each node corresponds to a feature and the links from a node to its subtrees correspond to the mutually exclusive rules (related to that feature) that is used to partition a data set. The c4.5 algorithm reads in vectors from the data set and, using some entropy-based metric, determines which features are relevant. As can be observed from Fig. 2, many of the 54 melody-related features are actually not included in the resulting decision tree. The algorithm has subsequently reduced the number of features by virtue of indicating which melodic features were relevant and which were not, for this classification challenge . Running the C4.5 algorithm on the data sets for the other classification challenges similarly produce a reduced feature set. The results are shown in Table 1. These reduced feature sets may be considered for use in specific classification challenges.

Classification Challenge	***Relevant Features (Listed according to importance in C4.5 Decision Tree)***
Classical vs. Non-Classical	*Variation of Dynamics* *Amount of Arpeggiation* *Minimum Note Duration* *Dominant Spread* *Primary Register* *Pitch Class variety* *Triple Meter* *Repeated Notes* *Rhythmic Looseness* *Most Common Pitch Class* *Chromatic Motion*
Bach vs. Chopin	*Size of Melodic Arcs* *Melodic Tritones* *Note Density* *Interval Between Strongest Pitches* *Staccato Incidence*
Contemporary vs. Jazz	*Rhythmic Variability* *Number of Common Pitches* *Average Note Duration*
Bach vs. Chopin vs. Contemporary vs. Jazz	*Rhythmic Variability* *Range* *Stepwise Motion* *Triple Meter* *Pitch Variety* *Repeated Notes* *Rhythmic Looseness* *Melodic Octaves* *Staccato Incidence* *Size of Melodic Arcs* *Interval Between Strongest Pitch Classes*

Table 1: Features found to be significant based on C4.5 Decision Tree Algorithm for each classification challenge

The reduced features sets specific to each classification challenge form the potentially-recommended features to be used whenever the goal is to measure the fitness of an evolved melody into either classical, non-classical, Contemporary or Jazz. We now compare these to the features recommended in a previous study.

The study by Towsey et al. suggested the use of 21 identified features. Based on comparing and cross-referencing feature definitions of both jSymbolic extractable features and Towsey recommended features, the 13 of the 54 jSymbolic extractable music features encompass the 21 features mentioned by Towsey et al. Table 2 shows this mapping.

jSymbolic Extractable Feature	***Recommended Features for Towsey et al***
Note Density	*Note Density*
Rhythmic Variability	*Rhythmic Range*
Rhythmic Looseness	*Rhythmic Variety, Rest Density, syncopaion, Patterns (Repeated Rhythmic values)*
Pitch Variety	*Pitch Variety*
Range	*Range*
Direction of Motion	*Contour Direction, Contour Stability*
Repeated Notes	*Climax Strength, Patterns (Repeated Pitch), Movement by step, Leap Returns*
Stepwise Motion	*Contour Direction, Contour Stability*
Quality	*Key-centered, non-scal notes, dissonant intervals*
Melodic Octaves	*Patterns*
Melodic Thirds	*Patterns*
Melodic Tritones	*Patterns*
Melodic Fifths	*Patterns*

Table 2: Mapping between jSymbolic extractable features and the recommendations by Towsey, et al.

After acquiring this information, a comparison of the classification accuracy in datasets involving the C4.5-suggested feature sets and the Toswsey-based features was performed. Instead of checking the classification performance of the resulting subset of features with one classifier, two classifiers were used for thoroughness. The classifiers used in this study were Naïve-Bayes and SVM. The choice of classifiers stems from the fact that their classification methodologies are different, and may potentially yield interesting results, e.g., consistencies or lack thereof.

The classifiers were used to check the performance of the classification with varying features sets. To estimate the actual performance of each of these classifiers, 10-fold cross validation was used. The actual accuracy values are shown in Table 3.

Results presented in Table 3 indicate that the choice of the classification algorithm (Naïve Bayes and SVM) does not affect which feature set (Towsey based or C4.5 recommended feature-set) yields the higher classification accuracy. That is, for each of the four tests performed, the features on which Naïve Bayes performed better were also those features where SVM registered a higher accuracy.

Furthermore, the feature sets based on the C4.5 results were able to classify subgenres and outperform the Towsey-based set when the MIDI files to be classified have a larger spectrum of differentiation (i.e., a dataset with 4 different subgenres, belong to both classical and non-classical groups).

Comparison Test	Naïve-Bayes (Recommended Features by C4.5)	Naïve-Bayes (Recommended features mapped from Towsey et al)	SVM (Recommended Features by C4.5	SVM (Recommended Features mapped from Towsey et al)
1. Classical vs. Non-classical	***96.5%***	91.5%	***97.5%***	95.25%
2. Bach vs. Chopin	75.5%	***91.5%***	77%	***92.5%***
3. Contemporary vs. Jazz	97%	***99***%	98%	***99***%
4. Bach vs. Chopin vs. Contemporary vs. Jazz	***94.5%***	90.25%	***95.5%***	92%

Table 3: Comparison among the classification accuracy results between the C4.5 recommended features and the features suggested by Towsey et al.

For the second classification challenge (Bach vs. Chopin), the Naïve-Bayes tenfold cross-validation results are far lower than those of the SVM-based results. This may be attributed to the similarity between Bach and Chopin compositions, as Naïve-Bayes classification makes use of the vectors of feature values as a basis for building a probability based model for classification. Further studies on the intricacies of these similarities on the works of the two composers are suggested. These results, however, do not have a negative effect in the subsequent identification of a good feature set for this specific classification challenge since what is important to note here is that in both Naïve-Bayes and SVM classification methods, Towsey et al recommended features have consistently scored higher than the C4.5 recommended features in the context of ten-fold cross-validation classification accuracy.

Overall, results indicate that the C4.5-based features are more appropriate than the Towsey features for classification involving a wide variety of elements (belonging to different music genres), while the Towsey features are still preferred for classifying within a genre.

5. CONCLUSION

The classification accuracy of Towsey-based feature set was compared with those based on the results of the C4.5 decision tree algorithm applied to a variety of classification challenges. All features involved in these tests were exclusive to melodic analysis. Features that analyze instrument fraction were removed.

Results have shown that C4.5-based feature sets vary for different classification challenges. This indicates that the features for fitness evaluation in future studies would have to consider various feature sets for computer generation of music from multiple genres.

The features that the C4.5 recommends appear to be better for classifying a wider variety of music samples. The Towsey-based feature set, however, still appears to be best used in a more specific "subgenre" classification of music files for datasets that contain elements from under the same major genre.

These results are useful as these feature sets will be used alongside distance measures to evaluate the quality of an evolved melody (i.e., output of an evolutionary algorithm) by analyzing it against target melodies, in the continuing effort to develop an automated fitness function for melodic evaluation.

6. FURTHER STUDY

It would be interesting to investigate the classification accuracy if the dataset size was increased significantly (e.g., several thousand records). The use of other classifiers, and the collection of the accuracy results from these will also allow better comparison between the C4.5 and Towsey features. Still, another possible direction can apply a different technique for feature selection in order to possibly identify different feature sets for different classification tasks.

This study has mapped features suggested by Towsey et al. for melodic analysis to the features that can be extracted by jSymbolic to produce a Towsey-based feature set for music genre classification based on feature definitions. Different mapping methodologies may also be explored in the future.

7. REFERENCES

[1] M. Alfonseca, M. Cebrian, and A. Ortega, "A Fitness Function for Computer Generated Music using Genetic Algorithms," *WSEAS Transactions on Information Science and Applications, Vol. 3:3*, p.518-525, Mar. 2006.

[2] A. Freitas, F. Guimaraes and R. Barbosa, "Ideas in Automatic Evaluation Methods for Melodies in Algorithmic Composition," Proceedings of the 9th Sound and Music Computing Conference, Copenhagen, Denmark, p.514-520, 2012

[3] J. Jensen, "Evolutionary Music Composition," *Institutt for datateknikk og informasjonsvitenskap NTNU University Library*, 2011.

[4] B. Johanson and R. Poli, "GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters," *Proceedings of the Third Annual Conference*, *MIT Press,* pp. 181-186, 1998.

[5] A. Jordanous, "A Fitness Function for Creativity in Jazz Improvisation and Beyond," *Proceedings of the First International Conference on Computational Creativity (ICCCX), Lisbon, Portugal*, 2010.

[6] P. Langston, "Six Techniques for Algorithmic Music Composition," *15th International Computer Music Conference (ICMC) in Columbus, Ohio, November 2-5,* 1989.

[7] M. Lo and S.M. Lucas, "Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions," *IEEE International Conference on Evolutionary Computation, pp. 601-608,* 2006.

[8] B. Manaris, P. Machado, C. McCauley, J. Romero and D. Krehbiel, "Developing Fitness Functions for Pleasant Music: Zipf's Law and Interactive Evolution Systems," *Applications of Evolutionary Computing, EvoWorkkShops Proceedings,*pp. 498-507, 2005.

[9] T. Nozaki and K. Kameyama, "Feature Selection for User-adaptive Content-Based Music Retrieval using Particle Swarm Optimization," *Proceedings of 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA 10)*, pp. 941-946, 2010.

[10] G. Papadopoulus and G. Wiggins, "AI Methods for Algorithmic Composition: A Survey, a Critical View and Future," *Proceedings from the AISB'99 Symposium on Musical Creativity*, Edinburg, Scotland, pp. 110-117.

[11] M. Towsey, A. Brown, S. Wright and J. Diederich, "Towards Melodic Extension Using Genetic Algorithms," *Educational Technology and Society 4, vol. 2,* 2001