THE DIFFERENCE BETWEEN HAYDN AND MOZART: NATURAL LANGUAGE APPROACHES TO COMPOSER RECOGNITION

Benjamin Selfridge
Dept. of Computer Science
University of Texas at Austin
benself@cs.utexas.edu

Aashish Sheshadri
Dept. of Computer Science
University of Texas at Austin
aashishs@cs.utexas.edu

ABSTRACT

This work is a comparative study of the success of two common machine learning techniques, N-grams and Hidden Markov Models, for the classification of sheet music by composer. We implement a simple N-gram model to perform composer recognition between two contemporary artists, Mozart and Haydn. We also introduce a novel approach to authorship attribution using HMMs. Our models incorporate rhythmic duration and note-to-note intervals, and use entirely supervised learning. We introduce schemes to use information in the musical scores as annotations for our algorithms. Our methods show improvements over previous attempts at the same classification task using methods from NLP.

1. INTRODUCTION

Musicologists are frequently interested in what distinguishes one body of work from another, whether it be between contemporary composers, or composers from differing eras and ethnic backgrounds. The most common approaches to composer recognition rely on qualitative analysis of scores, performed by trained musicologists, who use various cues to discern the different aspects of a composer's unique style. This can be very difficult when the composers are from the same time period and/or geographic location. Various computational methods have been employed to automate such classification tasks, but many of these methods only achieve true success on musical works from different eras and geographic regions.

This work uses two common techniques from Natural Language Processing (NLP), N-grams and Hidden Markov Models (HMMs), to perform classification of musical works by composer. We believe that music is a kind of language, and using computational ideas which exploit the linguistic features of music can help improve classification results. We choose two composers who belonged to the same era and cultural background - W. A. Mozart and F. J. Haydn - to demonstrate the success of our methods on contemporaries who frequently were in very close contact with each other. Therefore, our approach cannot rely on chronological or cultural variability, but instead must identify key statistical features that distinguish the compositional style of the two composers. We restrict our corpora even further, focusing on the violin parts contained in the first movements of various string quartets of the two composers. This helps control for the obvious variation between large-scale works vs. chamber music, and between the different styles employed for different instruments in that time period.

Various attempts have been made to formulate musicology tasks as problems in natural language. Some approaches include probabilistic grammars [1] [5] [14], N-gram models [15], and Hidden Markov Models (HMMs) [2] [4] [13]. Bod [1] and Gilbert & Conklin [5] both introduce different PCFGs designed to represent music; however, none of this work was designed to facilitate authorship attribution. The specific task of music authorship attribution has mostly been attempted using non-NLP methods, including clustering-based approaches [3], support vector machines [7], techniques from signal processing [10], and neural networks [9]. Authorship attribution for music using NLP-based approaches is a promising avenue of research [2] [15].

Related work has relied on unsupervised learning for HMMs [2]. This is surprising, considering supervised approaches generally outperform the unsupervised settings. We introduce supervision as information extracted from musical scores to train an HMM.

In Section 2 we describe a high-level view of our approach, including a detailed description of our data, tokenization methods, and algorithms. Section 3 describes the experimental results for the classification of works by Mozart and Haydn, including accuracy, precision, and recall for each method implemented, as well as a discussion of the results and improvements upon previous work. We summarize our work and suggest ideas for future directions in section 4.

2. COMPOSER RECOGNITION WITH N-GRAM AND HMM

2.1 Tokenization

The first step in performing sequential analysis of a data set is to determine exactly how to convert the data into a sequence. The music in our data set is primarily mono-

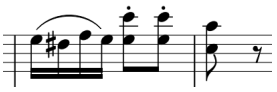

Figure 1. Excerpt from Mozart, K. 545, Movement 3

phonic; it can, for the most part, be seen as a chronological sequence of *pitches*, where each pitch has an associated *duration*. This suggests a natural way to tokenize each note in the music: the token is simply a pair (P, D) , where P is the pitch of the note, and D is its rhythmic duration.

However, there are some pitfalls to this approach that can be avoided by taking a slightly different viewpoint. Using absolute pitch and duration tends to result in a very large token vocabulary, with a high degree of variance. Previous work has found the *interval* between the current note and the previous note to be more effective than using absolute pitch [2]. The interval encodes a sequential relationship between consecutive notes, while absolute pitch encodes the relationship each note has with a fixed reference pitch (for instance, the key in which the piece is written). We take the viewpoint that an interval-based representation is preferable because it is (1) sequentially oriented, and (2) more general than the absolute pitch representation (in an interval representation, a C-D transition is identical to a D-E transition, while in an absolute representation, these would be distinct).

Wołkowicz, Kulka, and Kešelj use a similar construction for the duration - instead of using the actual, fixed length of each note, they use the ratio [1]

$$\frac{\text{current note length}}{\text{previous note length}}.$$

This approach ensures that the encoding is tempo-independent, and is also time-signature independent (the difference between cut time and $\frac{2}{4}$, for instance, is purely notational; they sound identical, but look very different in a score). We found that this representation was preferable to absolute duration.

Therefore, the two basic attributes of each note that we will be concerned with is the interval, I, between the current note and the previous note, and the *relative duration*, D, between the current note and the previous note. For chords and rests, we use a special token as the pitch ("Chord" and "Rest," respectively). To illustrate this tokenization, consider a fragment from the third movement of Mozart's Piano Sonata in C Major, K. 545, shown in Figure 1. This fragment can be represented with the following sequence of (interval, relative duration) pairs: (-1, 1.0), (2, 1.0), (-1, 1.0), (Chord, 2.0), (Chord, 1.0), (Chord, 1.0), (Rest, 1.0).

For the various experiments performed, we also introduced tokens for each measure boundary to determine if this additional information improved performance. If measure boundaries are included, the sequence in Figure 1 becomes (-1, 1.0), (2, 1.0), (-1, 1.0), (Chord, 2.0), (Chord, 1.0), (Barline, Barline), (Chord, 1.0), (Rest, 1.0).

[1] The authors actually use a quantized approximation of this ratio.

```
score = 0.0
for ngram in model.ngrams:
    diff  = model.prob(ngram)
            - testPiece.prob(ngram)
    den   = model.ngramProb(ngram)
            + testPiece.prob(ngram])
    score += 4.0 - (2.0 * (diff/den)) ** 2
return score
```

Figure 2. One of the two algorithms used to compute the similarity between a composer's model and a test piece, based on Wołkowicz et. al. [15]

2.2 N-gram Model

For our N-gram model, we represent each note as a pair (I, D), where I is the is the interval between the current note and the previous note, and D is the relative duration between the current note and the previous note. A separate N-gram model is used for each composer. The trained models are used to make classification decisions for each piece in the test set.

We tested two different N-gram methods for comparing test pieces to each model. The first, which we refer to as the "generative" model throughout this paper, is the frequently adopted approach, computing the probability that each model would generate that particular piece. For each value of N (we tested $N = 1, 2, 3$, and 4), we also manually incorporate lower values of N for smoothing. Naturally, we place the majority of the weight on the highest value of N.

The second N-gram comparison measure we used is a similarity measure given in Wołkowicz, Kulka, and Kešelj [15], which have demonstrated good performance for tasks similar to ours. We list the algorithm in Python-like pseudocode in Figure 2.

2.3 Hidden Markov Model

The Hidden Markov Model (HMM) is a very common statistical model for sequential data, where each token in a sequence has a hidden "state." They have been used very effectively for part-of-speech tagging in NLP [8]. HMMs have been used previously for music classification [2], but this study used unsupervised HMMs, taking the intervals to be the observable sequence, and deducing a set of hidden states automatically.

Our formulation of Hidden Markov Models for composer recognition is novel. Instead of simply taking a string of intervals as the observable sequence and training unsupervised HMMs to classify the test pieces, we take the intervals to be the observable sequences, and interpret the *relative duration* of each interval as its hidden state. This has the advantage of attaching an actual understandable meaning to the hidden states, as well as the obvious benefit that every musical piece comes pre-annotated. To evaluate the score of each model on a given test piece, a separate HMM is for each composer, and classification decisions on each musical work are made by choosing the model which yields the highest probability for the actual hidden state

sequence. In other words, we calculate the probability that each model would assign the *actual* sequence of relative durations to that piece's (observed) interval sequence.

3. EXPERIMENTAL EVALUATION

3.1 Data & Methodology

In order to ensure as much similarity between the two data sets as possible, we restricted our focus to the string quartets of two different composers, Mozart and Haydn. More specifically, we used the *first violin parts* of the *first movements* of Mozart's and Haydn's string quartets. String instrument parts are simpler to analyze as sequential data than, say, keyboard music, since string parts are primarily monophonic (chords do sometimes occur, but only occasionally).

We made use of a digitized sheet music corpus, made available through the `music21` toolkit. The corpus contains music collected from the Kern Scores library and various other sources, and includes 21 string quartets by Mozart and 59 by Haydn.[2] We also made use of the MAchine Learning for LanguagE Toolkit (MALLET) to implement the HMMs for our experiments [11].

/*Ben write about the preposessing steps – for eg how you handled some of the parsing blah*/

We ran twenty iterations of each method, and averaged the results. For each iteration, we took twenty random pieces by Mozart, and twenty random pieces by Haydn, and split them into training (approx. 70%) and testing (approx. 30%) sets. We chose to use the number of measures in each piece as the size of the piece; therefore, the number of pieces in the training and testing sets were somewhat dependent on the number of measures in the individual pieces that were randomly selected. We measure performance of each method using the metrics precision, recall, F_1 score and overall accuracy. We report statistical significance by computing a 95% confidence interval for each reported metric.

3.2 Results and Discussion

The overall results of our comparative analysis are given in Table 1. For the N-gram methods, we only display the most successful value of N in terms of overall F-measure. We also compare the success of the four different N-gram methods for different values of N, shown in Figure 3. The $\pm$ values are 95% confidence intervals.

For $N > 4$, we observed a sharp decline in performance, indicating that the method peaks for very small values of N. The generative N-gram measures peaked at $N = 2$, and the similarity N-gram measures peaked at $N = 3$.

The most successful method in terms of overall accuracy was N-gram with similarity measure and no bar lines, for $N = 3$. This method obtained an accuracy of 70.6%, and F scores of 64.3% and 74.3% for Mozart and Haydn, respectively. Considering the difficulty of this particular classification task, we consider this to be a very good result, especially considering the work of Chair & Vercoe [2], who achieved a best-case two-way classification accuracy of 77% between Irish and Austrian folk songs. Considering the fact that these folk songs originate from different geographic locations, and that Mozart and Haydn were close contemporaries who influenced each other heavily, an accuracy of 70.6% is a success.

[2] http://mit.edu/music21/doc/html/referenceCorpus.html

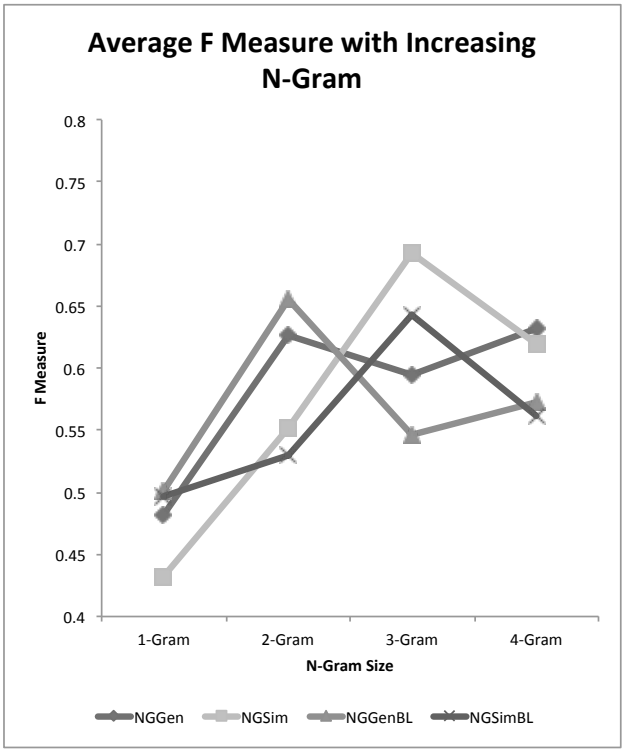

Figure 3. Comparative success of N-gram methods, with vs. without bar lines, and using generative vs. similarity measures

Overall, the N-Gram method using a similarity measure had a slight advantage over the generative N-Gram model. We suspect that this can be attributed to the fact that the similarity measure focuses on the N-grams that the composer model shares with the test piece, while generative probability is highly dependent on *all* N-grams seen in the training corpus, including the ones not present in the test piece.

However, it is important to note that N-gram is relatively unpredictable, in the sense that one has to first evaluate what values of N are optimal before having any confidence that the method will work for a particular set of composers, i.e. there is need for a validation set apart from a training set. For our purposes, that happened to be $N = 3$ for our most successful method, but there is no guarantee this will apply in the majority of cases. Furthermore, the N-gram methods had a tendency to be heavily biased toward one composer or another. The best values of N are depicted in Table 1, but for most of the other values, our classifier was heavily skewed toward either Mozart or Haydn.

We believe the Hidden Markov Model outlined in this

Method	Accuracy	Mozart			Haydn		
		Pr.	*Re.*	F_1	*Pr.*	*Re.*	F_1
NGGen (N=2)	$.652 \pm .101$	$.610 \pm .080$	$.854 \pm .097$	$.707 \pm .053$	$.784 \pm .158$	$.458 \pm .238$	$.546 \pm .217$
NGSim (N=3)	$.706 \pm .048$	$.738 \pm .074$	$.576 \pm .122$	$.643 \pm .095$	$.685 \pm .062$	$.817 \pm .054$	$.743 \pm .043$
NGGenBar (N=2)	$.668 \pm .050$	$.625 \pm .154$	$.692 \pm .142$	$.649 \pm .121$	$.718 \pm .054$	$.633 \pm .151$	$.663 \pm .078$
NGSimBar (N=3)	$.675 \pm .074$	$.773 \pm .153$	$.425 \pm .114$	$.546 \pm .123$	$.636 \pm .070$	$.892 \pm .077$	$.741 \pm .067$
HMM	$.586 \pm .091$	$.555 \pm .203$	$.625 \pm .072$	$.567 \pm .141$	$.647 \pm .087$	$.567 \pm .171$	$.590 \pm .089$
HMMBar	$.643 \pm .071$	$.646 \pm .060$	$.609 \pm .153$	$.622 \pm .099$	$.648 \pm .097$	$.675 \pm .072$	$.657 \pm .056$

Table 1. Results for each method method with 95% confidence intervals over 20 random random train-test splits. NGGen corresponds to results using the generative model, NGGenBar corresponds to results with the inclusion of bar lines, similar notation on NGSim (N-Gram with similarity measure), and HMM.

work has promise, despite its obvious deficiency when compared with N-gram methods. A quick glance at the results for the HMM does not reveal an obvious bias towards one composer or another. If this method were developed further, it could outperform both the N-gram approach and an unsupervised HMM approach. The fact that the inclusion of bar line markers improved the accuracy suggests that if we had more detailed data sets (which included things like phrase markers and accents), we might be able to achieve better results.

4. CONCLUSIONS & FUTURE WORK

We set out to investigate applicability of NLP methods to authorship attribution in musical works. To distinguish our research over prior work, we chose the difficult problem of distinguishing two composers who shared very similar backgrounds and who had direct influence on each other's style. We arrived at an acceptable model, which consistently showed encouraging results to dependably distinguish the two composers. We also discovered that despite the use of a sensible formulation of a Hidden Markov Model, this model failed to outperform simpler N-gram based approaches. In the process, we were made aware of the difficulty in representing music in linguistic terms; hence the absence of a standard linguistic viewpoint in the musicology community.

The absence of a small token & hidden state set is suspected to be the primary reason for our supervised HMM method's inability to outperform N-Gram methods. The chosen tokenization and the hidden states where far too many in number, compared to the data available to train, resulting in over-fitting of the training data. To enable structural and local constraints which the HMM method can take advantage of, we broke the music into individual measures, viewing each measure as a sentence. Unfortunately, this caused performance to drop instead of improve. We then attempted to incorporate more structure by marking end of measures as "Barline" tokens, which led to a noticeable improvement, comparable to our results for the generative N-gram approach. However, the HMM method was still outstripped by certain N-Gram configurations. The large tag set prohibited the use of a CRF, due to computing constraints. We attempted to reduce the tag set down to just five tags by thresholding relative duration levels; however, the resulting tag set was not capable of encoding fine idiosyncrasies present among the considered composers, and achieved generally poor results (hardly better than 50% accuracy). It is significant that for the majority instances where our HMMs failed to classify test pieces accurately, the scores predicted on the test pieces by the Mozart model vs. the Haydn model were often within a small delta, but *correct* classification usually were correct by much larger margins. This suggests that the right adjustment in our HMM formulation could lead to a notable improvement.

We are only aware of two previous attempts to distinguish between the music of Mozart and Haydn. The first is by Kaliakatsos-Papakostas and Epitropakis [9]. Our methods compare favorably to theirs, which obtained classification accuracy between 51.40% and 55.80% for these two composers using neural networks. The second, due to Hillewaere, Manderick, and Conklin [7], achieves a classification accuracy of 74.4% using support vector machines and 63.8% accuracy using N-Gram methods. Our data set was very similar to theirs, albeit more restricted, since we only analyzed the first movement of each string quartet. Since each study uses a slightly different data set and has slightly different goals, it is difficult to do a rigorous comparison of their success. We are interested in performing further cross-examinations of different techniques for identical classification tasks with identical data; we believe such work could help elucidate which approaches work best.

The primary contribution of this work is a direct comparison of two different methods from NLP to the same composer recognition task. We found that under the particular configurations we chose, the use of an N-gram-based classifier outperformed a more sophisticated Hidden Markov Model; however, this does not imply that HMMs are inappropriate for this task. On the contrary, Chai and Vercoe [2] achieved a reasonable degree of success using unsupervised HMMs to distinguish between musical works hailing from different geographical regions. Our HMM formulation achieved results comparable to theirs, which obtained a 66% success rate when distinguishing between works of similar origin (German vs. Austrian). This suggests that since our data sets did not have the advantage of slight cultural variability, the use of a supervised

HMM might be as good, or even better, than an unsupervised HMM. We would like to apply the methods in this paper to this same folk song classification task, and see if our methods perform better or worse.

We plan on venturing deeper into NLP by implementing a probabilistic context-free grammar (PCFG) for music which will use a combination of phrase grouping markers (commonly known as "slurs"), rests, and accents to construct a parse tree for pieces. We are limited by that majority of the data we currently have access to only has the raw music data, without ornamentation or phrasing markers. We also would like to note the lack of success for our Hidden Markov Model indicated to us that using more advanced NLP concepts might not lead to a higher success rate, and therefore focusing on optimizing the solutions presented may be the more viable option.

5. REFERENCES

[1] Bod, R.: *Probabilistic Grammars for Music*. In Proceedings of the Belgian-Dutch Conference on Artificial Intelligence. (2001)

[2] Chai, W. and Vercoe, B.: *Folk Music Classification Using Hidden Markov Models*. International Conference on Artificial Intelligence. (2001)

[3] Cilibrasi, R., Vitànyi, P., and de Wolf, R.: *Algorithmic Clustering of Music Based on String Compression*. Computer Music Journal. (2004) 49-67

[4] Conklin, D.: *Music Generation from Statistical Models*. In Proceedings of the AISB 2003 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences. (2003) 30-35

[5] Gilbert, É., and Conklin, D.: *A Probabilistic Context-Free Grammar for Melodic Reduction*. In Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence (IJCAI). (2007) 83-94

[6] Cuthbert, M. S., Ariza, C. and Friedland, L.: *Feature Extraction and Machine Learning on Symbolic Music Using the* `music21` *Toolkit*. In 12th International Society for Music Information Retrieval Conference (ISMIR). (2011) 387-392

[7] Hillewaere, R., Manderick, B. and Conklin, D.: *String Quartet Classification with Monophonic Models*. In 11th International Society for Music Information Retrieval Conference (ISMIR). (2010) 537-542

[8] Jurafsky and Martin: *SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Second Edition. McGraw Hill. (2008)

[9] Kaliakatsos-Papakostas, M., Epitropakis, M.G., and Vrahatis, M.N.: *Musical Composer Identification through Probabilistic and Feedforward Neural Networks*. Applications of Evolutionary Computation. (2010)

[10] Mandel, M. I. and Ellis, D. P. W.: *Song-Level Features and Support Vector Machines for Music Classification*. In 6th International Conference on Music Information Retrieval (ISMIR). (2005) 594-599

[11] McCallum, A. K.: *MALLET: A Machine Learning for Language Toolkit*. http://mallet.cs.umass.edu. (2002)

[12] Mosteller, F. and Wallace, D.L.: *Inference and disputed authorship: The Federalist*. AddisonWesley. (1964)

[13] Pollastri, E. and Simoncelli, G.: *Classification of Melodies by Composer with Hidden Markov Models*. Proceedings of the First International Conference on WEB Delivering of Music (WEDELMUSIC01). (2001)

[14] Salas, H. A. G., Gelbukh, A., Calvo, H., and Soria, F. G.: *Automatic Music Composition with Simple Probabilistic Generative Grammars*. Polibits, Vol. 44. (2011) 59-65

[15] Wołkowicz, J., Kulka, Z., Kešelj, V.: *N-Gram-Based Approach to Composer Recognition*. Archives of Acoustics, Vol. 33 Issue 1. (2008) 43-55

[16] Changsheng Xu: *Automatic Music Classification and Summarization*, IEEE Transactions on Speech and Audio Processing, Vol. 13 No. 3. (2005) 441-450