

EXTENDING A MODEL OF MONOPHONIC HIERARCHICAL MUSIC ANALYSIS TO HOMOPHONY

Phillip B. Kirlin and David L. Thomas

Department of Mathematics and Computer Science, Rhodes College

kirlinp@rhodes.edu, thodl15@gmail.com

ABSTRACT

Computers are now powerful enough and data sets large enough to enable completely data-driven studies of Schenkerian analysis, the most well-established variety of hierarchical music analysis. In particular, we now have probabilistic models that can be trained via machine learning algorithms to analyze music in a hierarchical fashion as a music theorist would. Most of these models, however, only analyze the monophonic melodic content of the music, as opposed to taking all of the musical voices into account. In this paper, we explore the feasibility of extending a probabilistic model developed for analyzing monophonic music to function with homophonic music. We present details of the new model, an algorithm for determining the most probable analysis of the music, and a number of experiments evaluating the quality of the analyses predicted by the model. We also describe how varying the way the model interprets rests in the input music affects the resulting analyses produced.

1. INTRODUCTION

Music analysis is primarily concerned with studying the structure of music compositions, both at the small- and large-scale levels. Hierarchical music analysis, best exemplified by *Schenkerian analysis*, illustrates the structure of a music composition by identifying hierarchical relationships among the notes of the music. These relationships collectively group the notes into a series of hierarchical levels that demonstrate the function of each note in the music in relation to other notes at various levels of the hierarchy.

One of the complicating factors of Schenkerian analysis is that there is no single established algorithm for performing the analysis. Instead, textbooks present guidelines and sample analyses from which students gradually learn the techniques, often through trial and error. Historically, there have been a number of research endeavors that attempted to replicate the Schenkerian analysis procedure: purely algorithmic efforts run into problems because of the

conflicting and ambiguous nature of the Schenkerian analysis rule set [4, 6] and up until recently, machine learning approaches often hit roadblocks due to the lack of a large standardized corpus of Schenkerian analysis upon which to train [5, 10, 11].

However, more recent efforts to create such a corpus of Schenkerian analysis have led to a data-driven system capable of learning to analyze music in a hierarchical fashion [7, 9]. This system, however, is only capable of hierarchically analyzing the monophonic main melody of the composition, with any other voices or harmonic parts contributing only auxiliary information to the algorithms. In this work, we study the practicality of extending this monophonic model of music analysis to support homophonic textures with a soprano part and a supporting bass line. We present evidence that there are homophonic patterns that can be harnessed by machine learning techniques, demonstrate the workings of a probabilistic model on homophonic input, and evaluate the system both for accuracy and for determining where mistakes are made.

Because Schenkerian analysis is one of the most comprehensive forms of music analysis available today [3], the uses of this work extend beyond the obvious application of studying computationally-produced analyses of music. Algorithms for calculating music similarity or identifying musical styles or genres could be enhanced with the probabilistic model described here, as could systems for music recommendation or new music discovery. At a more fundamental level, studying computational models of music analysis can lead us towards a better understanding of musical perception and structure [1].

2. MODELING MONOPHONY AND HOMOPHONY

Schenkerian analysis hypothesizes that music compositions are structured as a series of hierarchical levels defined by *prolongations*: situations where a note, chord, or melodic interval remains in control of a passage of music even though it may not be sounding constantly during that time. Consider the five-note descending phrase in Figure 1, occurring over G major harmony. In this melody, a music theorist would identify a prolongation over the first three notes D–C–B: the note C prolongs the passing motion from the D to the B. A similar prolongation occurs among the notes B–A–G. This places the notes D, B, and G — the notes being prolonged — at a higher structural



© Phillip B. Kirlin and David L. Thomas.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Phillip B. Kirlin and David L. Thomas. “Extending a Model of Monophonic Hierarchical Music Analysis to Homophony”, 16th International Society for Music Information Retrieval Conference, 2015.

level than the C or the A.



Figure 1. An arpeggiation of a G-major chord with passing tones. The slurs are a Schenkerian notation used to indicate the locations of prolongations.

However, there is another level of prolongation at work in this melody. The interval of a fifth between the first note D and the last note G is prolonged by the motion to and from the B in the middle. This leads to a three-level hierarchy of intervals as shown in the binary tree in Figure 2(a). An equivalent structure, illustrated in Figure 2(b), is known as a *maximal outerplanar graph* or *MOP*: this structure is equivalent to a binary tree of melodic intervals but represents the same information more succinctly [14]. We claim that any hierarchical analysis can be represented as a MOP, and therefore, illustrated as a fully-triangulated polygon.

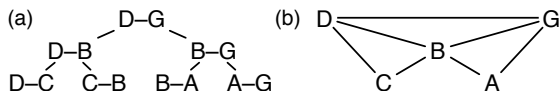


Figure 2. The prolongational hierarchy of a G-major chord with passing tones represented as (a) a tree of melodic intervals, and (b) a MOP.

By combining a corpus of musical excerpts and corresponding MOP analyses with a supervised machine learning algorithm, it is possible to learn a probabilistic model over MOP structures. This model admits a $O(n^3)$ algorithm for determining the most probable MOP for a new piece of music [9].

In the original MOP model of prolongation, a single triangle describes the elaboration of a parent melodic interval by two child intervals. This model, as first conceived, can only represent monophonic note sequences. As it would be desirable to enable hierarchical music analysis of all the voices within a composition, it is worth exploring extensions to represent multi-voice musical textures. One possibility is using a separate MOP to represent the structure of each voice in the music: this would allow for independent analyses of each voice. This representation, however, would increase the computational complexity of the algorithm for determining the most probable MOP analysis from $O(n^3)$ to $O(n^6)$ for a two-voice composition [13].

Instead, we investigate MOPs that store multiple pitches in a single vertex. In particular, we study MOPs that store up to two pitches per vertex, with the pitches derived from separate soprano and bass voices. We call these new MOPs *interval MOPs*, so named because the two pitches stored in a vertex form a harmonic interval between the soprano and bass parts. Where it is necessary to differentiate between the two varieties of MOPs, we will call the original type of MOP a *monophonic MOP*.

2.1 The Interval MOP Model

Consider the five-note descending melodic pattern from Figure 1, now augmented with a bass line, as in Figure 3(a). The equivalent interval MOP is shown alongside, in Figure 3(b). Clearly, the triangles within an interval MOP have the same prolongational interpretations as in monophonic MOPs. One will observe, however, that there can be potential conflicts in the prolongational structure between different voices. For instance, consider the melodic voices in Figure 4(a), where the prolongational slurs imply that the MOP-like structure in Figure 4(b) is necessary to represent the prolongations in the soprano and bass parts. Unfortunately, the definition of a MOP prohibits any edge-crossings of this variety: such a crossing breaks the strict hierarchy necessary to maintain the mathematical (and as we will show later, computational) properties inherent in a MOP.

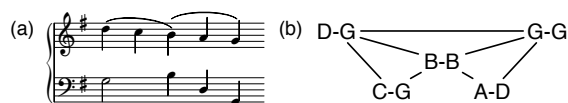


Figure 3. (a) A musical passage with independent soprano and bass parts, and (b) the corresponding interval MOP.

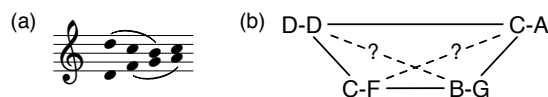


Figure 4. (a) A musical passage with conflicting soprano and bass prolongations and (b) the only way of representing both prolongations in a MOP-like structure, illustrating the conflicting edges that would arise.

Though note-against-note textures are easily represented in interval MOPs, some explanation is necessary for how to handle more complicated rhythms. When one voice has a change in pitch while another voice has a sustained pitch (i.e., oblique motion), the sustained note may be duplicated in the interval MOP. For example, consider the first and second beats in Figure 3(a): the bass note G is held for both of these beats and is duplicated in the first two vertices of the interval MOP in Figure 3(b).

Rests can be explicitly represented in an interval MOP. Suppose that the half note G in the bass part of of Figure 3(a) were a quarter note followed by a quarter rest. This would alter the interval MOP of Figure 3(b) to have the vertex C-G store the pair C-(rest) instead.

Every interval MOP contains two additional vertices, representing the *START* and *FINISH* of a composition. The *START* vertex is always (temporally) the first vertex in a MOP: it does not correspond to the first note in the music, but rather should be thought of as occurring before the start of the music. Similarly, the *FINISH* vertex is always the last vertex in a MOP and temporally occurs after the end of the music. These extra vertices are necessary to permit any pair of intervals in a MOP to represent the most

abstract level of the musical hierarchy. Consider a MOP not containing such extra vertices: because prolongations are oriented temporally, with the left and right vertices of a prolongation always higher in the structural hierarchy than the middle vertex, the most abstract edge of the MOP would have to be between the first harmonic interval of the music and the last. Because the first and last harmonic intervals are not always the most musically important pair of events, including START and FINISH allow for any pair of harmonic intervals in the music to represent the most abstract level in the structural hierarchy [14].

In the remainder of this paper, we study the feasibility of using interval MOPs to represent Schenkerian analyses for two-voice homophonic compositions. We do not consider polyphonic textures with completely independent voices due to the likelihood of encountering conflicting prolongational structures, such as in Figure 4(a), and the inability of interval MOPs to represent such structures, as discussed earlier. We show that there are patterns that arise in the encoding of music analyses in the interval MOP structure, we illustrate algorithms for harnessing these patterns and identifying the probabilistically most likely interval MOP analysis for new pieces of music, and we conclude with experiments showing how (1) accurately these algorithms can reproduce ground truth analyses and (2) what sorts of errors the algorithms make.

3. CONSTRUCTING INTERVAL MOPS FROM REAL-WORLD ANALYSES

Earlier work in computational Schenkerian analysis has verified that there are regularities in the prolongations that humans identify during the analysis procedure. Specifically, if we recall that each triangle in a MOP corresponds to a three-note prolongation, then it has been shown that various types of triangle occur more frequently than others [9]. However, in order to confirm this finding for interval MOPs, we first require an algorithm to convert a pair of monophonic MOPs — one representing the soprano line and one representing the bass line — into a single interval MOP. The strategy we use is to first align the notes of the monophonic MOPs to create an initial completely-untriangulated interval MOP consisting of corresponding pairs of notes between the soprano and bass MOPs. A pair of notes is created any time there is an temporal overlap between a soprano note and a bass note, so an individual note may appear multiple times in an interval MOP. Next, interior edges are added from the original soprano MOP in corresponding locations in the interval MOP; this has the same effect as copying every prolongation from the soprano MOP to the interval MOP. Lastly, all edges are added from the original bass MOP to the interval MOP that can be added without creating conflicts (overlapping edges). We prioritize the soprano prolongations because the soprano voice is more easily heard in the overall music and usually is more melodically significant.

We ran an experiment to verify the appropriateness of using interval MOPs as a representation of a multi-voice Schenkerian analysis. We used an updated version of the

SCHENKER41 corpus: a data set containing 41 excerpts of common practice period music and corresponding Schenkerian analyses. All of the music in the corpus is for a solo keyboard instrument or for voice with keyboard accompaniment, is in a major key, and does not modulate. All of the excerpts are between two and sixteen measures in length, but most are either four or eight measures long. 39 of the Schenkerian analyses in the corpus are taken from textbooks and two analyses were sourced from a local expert music analyst [7]. We translated all the musical excerpts from monophonic MOPs to interval MOPs using the algorithm described above. Because we are interested in confirming that there are patterns in prolongational data as represented by interval MOPs, we examined how often every type of triangle appeared in the converted interval MOPs.

Specifically, we calculated the frequencies of all triangle types in the corpus in order to test the statistical significance given the null hypothesis that the corpus analyses represented as interval MOPs resemble randomly-constructed MOPs in their triangle frequencies. Determining the expected frequency of a triangle in a MOP under this null hypothesis is straightforward precisely because of the mathematical underpinnings of the MOP formulation.

Assume we have a polygon with n vertices, numbered clockwise from 0 to $n-1$, and we are interested in the number of times that the triangle between vertices x , y , and z ($x < y < z$) appears across all complete triangulations of this polygon. We observe that any triangle drawn inside a polygon necessarily divides the interior of the polygon into four regions: the triangle itself, plus the three regions outside the triangle but inside the polygon, as in Figure 5 (though it is possible for some of these regions to be degenerate line segments). Any complete triangulation of the polygon that contains $\triangle xyz$ must necessarily completely triangulate the three regions outside of the triangle, and we simply multiply the number of ways of triangulating each of those three regions to obtain the total number of complete triangulations that contain $\triangle xyz$.

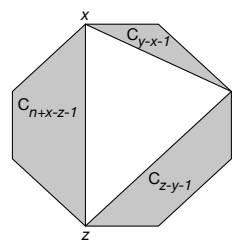


Figure 5. The number of times $\triangle xyz$ appears in all possible triangulations of the octagon can be calculated from the sizes of the shaded regions.

The number of ways of triangulating each of the three regions is directly related to the size of each region, which we can calculate from the values of the vertices x , y , and z . The sizes (number of vertices in the polygons) of these regions are $y - x + 1$, $z - y + 1$, and $n + x - z + 1$, respectively. Precisely because edges in MOPs cannot cross, the number of triangles that will appear in each of these

regions is solely a function of the size of each region: the number of ways to triangulate each region is the Catalan number for the size of each region minus two, and therefore the complete calculation for the expected frequency is $C_{y-x-1} \cdot C_{z-y-1} \cdot C_{n+x-z-1}$, where $C_i = \frac{1}{i+1} \binom{2i}{i}$.

We ran binomial tests for each type of triangle by comparing the expected frequencies of the triangles with the observed frequencies in the corpus. We found that there were 48 different types of triangles that were possible in the corpus of interval MOPs, where a triangle type was defined by categorizing the three harmonic intervals between the endpoints as either *consonant*, *dissonant*, *single* (for single notes), or *not applicable* (for MOP vertices containing a START or FINISH vertex). We checked for triangles that were statistically significant at the 5% level. By using the Šidák correction, we found that only triangles that had a p -value of less than 0.001 would be considered significant; there were six triangles that matched this criteria. These triangles are described in Figure 6.

4. A PROBABILISTIC INTERPRETATION OF INTERVAL MOPs

Now that we have verified that there are statistically significant prolongational patterns in the corpus of interval MOPs, we may continue towards our goal of developing an algorithm to harness the patterns in such a way as to be able to analyze new compositions. We proceed in a manner similar to that which was used in the original probabilistic model of monophonic MOPs [9].

Given two monophonic sequences of notes, a soprano line $S = s_1, s_2, \dots, s_n$, and a bass line $B = b_1, b_2, \dots, b_m$, our goal is to calculate the most probable analysis A for these notes, which means maximizing $P(A | S, B)$. An interval MOP is defined by the set of triangles T_1, T_2, \dots, T_k within, and thus we define

$$P(A | S, B) = P(T_1, T_2, \dots, T_k).$$

This full joint probability distribution cannot be efficiently estimated using the amount of training data available to us, so we decompose it into a product of probabilities of individual triangles:

$$P(T_1, T_2, \dots, T_k) \approx P(T_1) \cdot P(T_2) \cdots P(T_k).$$

In other words, we assume that each triangle in an interval MOP is independent of the other triangles. An earlier experiment [8] verifies that this does not appreciably alter the probabilistic rankings of the MOP analyses.

We define the individual probability of a triangle within an interval MOP analysis in terms of random variables representing the three endpoints of the triangle:

$$P(T_i) = P(C_i | L_i, R_i).$$

The three random variables in this distribution each represent either a harmonic interval or a single soprano or bass note with a rest in the other voice. The endpoints are unambiguously named because MOPs are oriented by the temporal dimension: later notes always appear to the right of earlier notes.

Our goal is to use the SCHENKER41 data set to estimate $P(C | L, R)$, but this is impractical due to the high-dimensional nature of the random variables involved: we would like to use melodic, harmonic, and rhythmic features of the triangle endpoints, and a data set of 41 analyses does not give us enough data to do this by directly counting triangle frequencies and normalizing them into a probability distribution. Instead, we use random forests [2], a type of ensemble classifier, to estimate this probability. Specifically, we create a large collection of decision trees, with each tree designed to predict a certain feature of the middle point C , trained on a subset of the features of the left and right endpoints L and R . The predictions of all the trees for a given feature of C are then aggregated and normalized into a probability distribution [12].

4.1 Features

We use a set of twenty-seven features to represent a triangle. Specifically, we use eighteen features solely involving the left and right endpoints (L and R) to predict nine features for the center point (C). These features are:

- The category of the interval involving the soprano and the bass note, listed either as *Cons* (Consonant) or *Dis* (Dissonant) (three features, one each for L , C , and R).
- For a given note in an interval, the scale degree (1-7) of the note (six features).
- The harmony present in the music at the time of the interval as a Roman numeral (six features). These harmonic labels, provided by experts, are included in the SCHENKER41 corpus.
- The broader category of harmony present in the music at the time of the interval, such as *tonic* or *dominant* (six features).
- For a given note in an interval, whether the note was a chord tone in the harmony present at the time (six features).

In some situations, certain features are not applicable. In the case that L or R is a START or FINISH vertex, the features are marked with invalid values to denote their ineligibility. Furthermore, in situations where L , C , or R is not an interval, but instead a single note, only half of the attributes per category listed above are applicable.

5. EVALUATION

As mentioned earlier, one reason for preferring interval MOPs to a more complicated representation for multi-voice prolongational hierarchies is the mathematical elegance of the structure, which makes it an efficient choice from which to infer probabilistic patterns. No less important is fact that computing the optimal triangulation of a polygon can be done in $O(n^3)$ time by using a standard dynamic programming algorithm. This is the basis of the existing PARSEMOP algorithms designed for monophonic MOPs; we adapt the algorithms to work with interval MOPs.

There are three variations of PARSEMOP; each variation is given different amounts of *a priori* information regarding the most abstract level of the hierarchical analysis

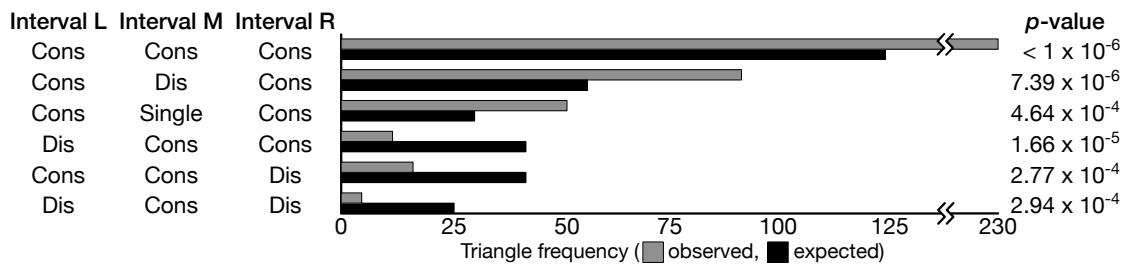


Figure 6. Types of triangles statistically significant at the 5% level.

being produced. Heinrich Schenker theorized that all tonal music compositions were derived from one of a small set of simple structures involving a short melodic progression harmonized in a specific way. Thus, the Schenkerian analysis process of finding prolongations theoretically always reveals one of these structures, known as the *fundamental structure* or *Ursatz*, at the background level.

All variants of the PARSEMOP algorithm accept the musical score as input, and are told which notes of the score constitute the soprano and bass lines. PARSEMOP-A has no conception of the *Ursatz* built into the algorithm, and therefore will not necessarily find one of the fundamental structures in the music when it runs. PARSEMOP-B, on the other hand, in addition to the musical score, is also informed as to which specific notes in the score should be placed into the background fundamental structure. This version of PARSEMOP, therefore, will always find the correct background structure. PARSEMOP-C is a compromise between the structurally-unaware PARSEMOP-A, and the overly-aware PARSEMOP-B: this version is informed as to which musical *itches* constitute the fundamental structure and in what order they should appear in the output, but the algorithm is not told the exact locations of the corresponding notes in the score.

We used leave-one-out cross-validation in conjunction with the SCHENKER41 corpus to evaluate how well the three PARSEMOP algorithms could reproduce the ground-truth analyses in the corpus. Specifically, for each of the 41 excerpts in the corpus, we trained our probabilistic model on the interval MOPs derived from the other 40 excerpts, and then used each PARSEMOP algorithm to derive the most probable analysis for the original piece omitted. We compared the algorithmically-produced MOPs to the ground-truth MOPs using a metric called *edge accuracy*, which is the proportion of internal edges in an interval MOP that correspond to an edge in the ground-truth interval MOP. We use this metric rather than proportion of triangles that match between two analyses because there are cases where two analyses can have edges in common, indicating some similarity, yet have no triangles in common.

Although occasionally music analysts may disagree on what the “correct” Schenkerian analysis should look like for a piece of music, the limited amount of data allowed us only one ground-truth analysis per musical excerpt.

Figure 7 shows the aggregate edge accuracy levels for the three PARSEMOP algorithms. For the sake of compar-

ison, we included the average edge accuracy as would be obtained by a baseline algorithm that analyzes music randomly: this hypothetical algorithm creates triangulations uniformly at random from the space of all possible complete triangulations.

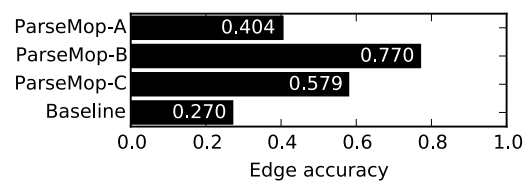


Figure 7. Edge accuracies for the three PARSEMOP algorithms and the baseline randomized algorithm.

Interestingly, the accuracy levels obtained by analyzing all possible analyses for a given piece of music do not follow a uniform or normal distribution. In fact, the distribution of edge accuracies as would be obtained by selecting a complete triangulation uniformly at random is quite skewed, as can be seen in Figure 8. This means that even though the PARSEMOP algorithms never break 80% accuracy, when compared against the baseline algorithm, they are doing quite well. In fact, we can use the distribution of edge accuracies from the baseline algorithm to judge each PARSEMOP algorithm’s accuracy against the null hypothesis that the PARSEMOP algorithm does no better than random. This results in p -values of 0.1022, < 0.0001 , and 0.0061 for the -A, -B, and -C varieties of PARSEMOP, respectively.

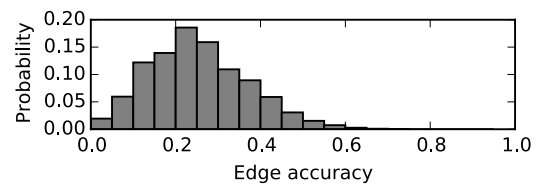


Figure 8. Distribution of edge accuracies under the baseline random algorithm.

We also analyzed where in the algorithmically-produced MOP analyses PARSEMOP was making mistakes. Specifically, for each non-perimeter edge in a PARSEMOP analysis that did not correspond to an edge in the ground MOP, we computed the *edge depth*: a number between 0 and 1 indicating how far down in the structural hierarchy the edge lies, with 0 being the most abstract level of the hierarchy

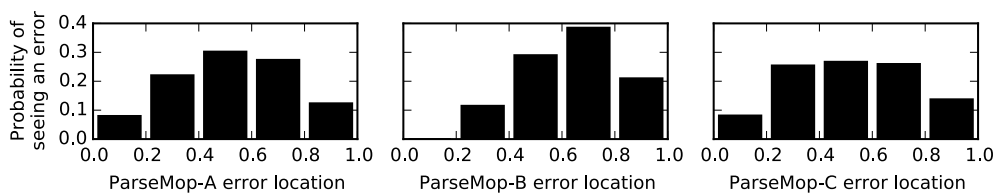


Figure 9. Probability distributions of the locations of errors in the PARSEMOP analyses.

and 1 being the surface level of the music. We produced probability distributions illustrating the hierarchical locations where PARSEMOP is most likely to make an error; these are shown in Figure 9. These distributions are unsurprising: PARSEMOP-A and -C make fewer errors at the extreme levels of the hierarchy due to the surface-level music constraining the low-level decisions at one end and fewer high-level decisions to be made at the other. Furthermore, PARSEMOP-B makes fewer mistakes at the most abstract level because the *Ursatz* has been supplied ahead of time.

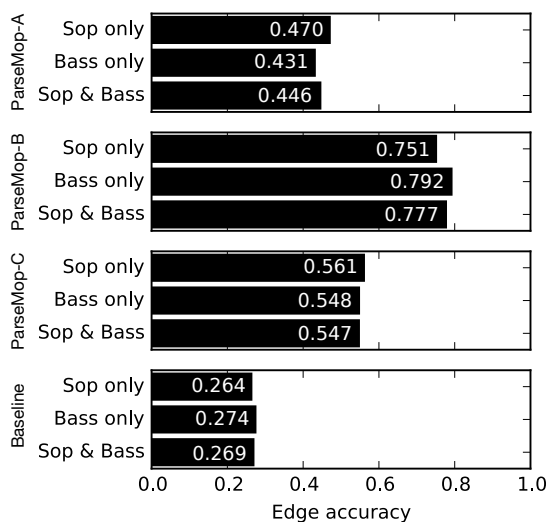


Figure 10. Edge accuracies after three varieties of rest adjustment for the three PARSEMOP algorithms and the baseline randomized algorithm.

6. ACCOUNTING FOR RESTS

Though the results from the previous section indicate that the PARSEMOP algorithms using interval MOPs are doing relatively well when compared against the baseline algorithm, there is still plenty of room for improvement. One area we hypothesized that could be adversely affecting accuracy is the presence of rests in the soprano and bass parts. Recall that each vertex in an interval MOP holds both a soprano and bass note, but only for note pairs sounding at the same time. Notes may be paired with rests if a rest is “sounding” at the same time as a note in the other voice. This may be the wrong musical interpretation, however, in situations where rests are stylistic indications for performance (e.g., a substitute for staccato markings), rather than indications that a melodic line contains a true pause. Thus, we present a modification to the interval MOP construction algorithm that within a voice, extends each note through

any intervening rests up to the start of the next note within a voice. In essence, all rests are eliminated from the soprano and bass parts, and notes durations are increased to fill the gaps. There are three different versions of the rest adjustment algorithm that control which voices are adjusted: just the soprano, just the bass, or both voices adjusted.

The rest adjustment algorithm, when applied to all of the soprano lines in the corpus, modifies the durations of 102 notes out of a total of 931. When applied to the bass line, the algorithm elongates 316 notes out of a total of 908.

We re-ran the earlier cross-validation experiment with each of the three versions of the rest adjustment algorithm; the updated edge accuracies are shown in Figure 10. Interestingly, the only situation in which the rest adjustment algorithm had any large affect on the edge accuracy was for PARSEMOP-A, where it increased the edge accuracy from roughly 40% to between 44% and 47%. Effects on PARSEMOP-B and PARSEMOP-C were much smaller, and in some cases caused a slight decrease in accuracy. The effects on the p -values under the null hypothesis that the PARSEMOP analyses resemble analyses chosen at random were also small; these new p -values are shown in Table 1.

PARSEMOP variant:	A	B	C
Sop only	0.0420	< 0.0001	0.0090
Bass only	0.0853	< 0.0001	0.0133
Sop & Bass	0.0601	< 0.0001	0.0110

Table 1. p -values calculated under the null hypothesis that PARSEMOP analyses (with the rest adjustment algorithm) resemble analyses done randomly.

7. DISCUSSION

Overall, the results from this study are encouraging. The edge accuracies and their improvement over the random baseline algorithm imply that interval MOPs can successfully model a homophonic prolongational hierarchy. Interval MOPs maintain all of the mathematical and computational advantages of monophonic MOPs, including a straightforward learning algorithm and a computationally-efficient method for finding the most probable analysis for a new piece of music.

However, it is clear that interval MOPs cannot represent all of the prolongational situations that could arise in polyphonic textures, namely conflicting prolongations between voices. We plan on studying the feasibility of using independent MOPs for the soprano and bass; this will alleviate the representational issue, but may require an approximation algorithm for finding the most probable MOPs for new compositions in order to remain computationally tractable.

8. REFERENCES

- [1] Samer A. Abdallah and Nicolas E. Gold. Comparing models of symbolic music using probabilistic grammars and probabilistic programming. In *Proceedings of the Joint 11th Sound and Music Computing Conference and 40th International Computer Music Conference*, pages 1524–1531, 2014.
- [2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [3] Matthew Brown. *Explaining Tonality*. University of Rochester Press, 2005.
- [4] R. E. Frankel, S. J. Rosenschein, and S. W. Smoliar. Schenker’s theory of tonal music—its explication through computational processes. *International Journal of Man-Machine Studies*, 10(2):121–138, 1978.
- [5] Édouard Gilbert and Darrell Conklin. 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*, pages 83–94, Hyderabad, India, 2007.
- [6] Michael Kassler. Proving musical theorems I: The middleground of Heinrich Schenker’s theory of tonality. Technical Report 103, Basser Department of Computer Science, School of Physics, The University of Sydney, Sydney, Australia, August 1975.
- [7] Phillip B. Kirlin. A data set for computational studies of Schenkerian analysis. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 213–218, 2014.
- [8] Phillip B. Kirlin and David D. Jensen. Probabilistic modeling of hierarchical music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 393–398, 2011.
- [9] Phillip B. Kirlin and David D. Jensen. Using supervised learning to uncover deep musical structure. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1770–1776, 2015.
- [10] Alan Marsden. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289, 2010.
- [11] Panayotis Mavromatis and Matthew Brown. Parsing context-free grammars for music: A computational model of Schenkerian analysis. In *Proceedings of the 8th International Conference on Music Perception & Cognition*, pages 414–415, 2004.
- [12] Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, September 2003.
- [13] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.
- [14] Jason Yust. *Formal Models of Prolongation*. PhD thesis, University of Washington, 2006.