Using Supervised Learning to Uncover Deep Musical Structure

Phillip B. Kirlin
Department of Mathematics and Computer Science
Rhodes College
Memphis, Tennessee 38112

David D. Jensen
School of Computer Science
University of Massachusetts Amherst
Amherst, Massachusetts 01003

Abstract

The overarching goal of music theory is to explain the inner workings of a musical composition by examining the structure of the composition. Schenkerian music theory supposes that Western tonal compositions can be viewed as hierarchies of musical objects. The process of Schenkerian analysis reveals this hierarchy by identifying connections between notes or chords of a composition that illustrate both the small- and large-scale construction of the music. We present a new probabilistic model of this variety of music analysis, details of how the parameters of the model can be learned from a corpus, an algorithm for deriving the most probable analysis for a given piece of music, and both quantitative and human-based evaluations of the algorithm's performance. This represents the first large-scale data-driven computational approach to hierarchical music analysis.

Introduction

Music analysis is largely concerned with the study of musical structures: identifying them, relating them to each other, and examining how they work together to form larger structures. Analysts apply various techniques to discover how the building blocks of music, such as notes, chords, phrases, or larger components, function in relation to each other and to the whole composition. *Schenkerian analysis* is a particular style of music analysis. This widely-used theory of music posits that compositions are structured as hierarchies of musical events, such as notes or intervals, with the surface level music at the lowest level of the hierarchy and an abstract structure representing the entire composition at the highest level. This type of analysis is used to reveal *deep structure* within the music and illustrate the relationships between various notes or chords at multiple levels of the hierarchy.

For more than forty years, music informatics researchers have attempted to construct computational systems that perform automated music analysis, as having systems capable of analyzing music is helpful for both academic and applied reasons. Music naturally lends itself to scientific study because it is a "complex, culturally embedded activity that is open to quantitative analysis" (Cook 2005), and approaching musical studies from a scientific standpoint brings a certain empiricism to the domain which historically has been difficult or impossible due to innate human biases (Volk, Wiering, and van Kranenburg 2011; Marsden 2009).

Despite the importance of Schenkerian analysis in the music theory community, computational studies of Schenkerian analysis are few and far between due to the lack of a definitive, unambiguous set of rules for the analysis procedure, limited availability of high-quality ground truth data, and no established evaluation metrics (Kassler 1975; Frankel, Rosenschein, and Smoliar 1978). More recent models, such as those that take advantage of new representational techniques (Mavromatis and Brown 2004; Marsden 2010) or machine learning algorithms (Gilbert and Conklin 2007) are promising but still rely on a hand-created set of rules.

In contrast, the work presented here represents the first purely data-driven approach to modeling hierarchical music analysis. We use the largest set of encoded music analyses in existence to develop a probabilistic model of the rules of music analysis, deploy this model in an algorithm that can identify the most likely analysis for a piece of music, and evaluate the system using multiple metrics, including a study with human experts comparing the algorithmic output head-to-head against published analyses from textbooks.

There are many potential uses of this work. Aside from the straightforward option of analyzing new music, the probabilistic model and algorithm described here could be used in systems for calculating music similarity, such as in music recommendation or new music discovery; intelligent tutoring systems for teaching music composition or Schenkerian analysis itself; or in music notation software.

The MOP Representation

Schenkerian analysis views a musical composition as a series of hierarchical levels. During the analysis process, structural levels are uncovered by identifying *prolongations*, situations where a musical event (a note, chord, or harmony) remains in control of a musical passage even when the event is not physically sounding during the entire passage. Consider the five-note melodic sequence shown in Figure 1, a descending sequence from D down to G. Assume that an analyst interprets this passage as an outline of a G-major chord, and wishes to express the fact that the first, third, and fifth notes of the sequence (D, B, and G) are more structurally important in the music than the second and fourth notes (C and A). In this situation, the analyst would interpret the C and A as *passing tones*: notes that serve to transition smoothly between the preceding and following notes by filling in the space in between. From a Schenkerian aspect, we would say that there are two prolongations at work here: the motion from the D to the B is prolonged by the note C, and the motion from the B to the G is prolonged by the note A.

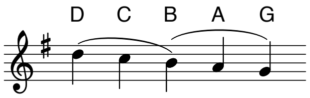

Figure 1: An arpeggiation of a G-major chord with passing tones. The slurs are a Schenkerian notation used to indicate the locations of prolongations.

However, there is another level of prolongation evident in this passage. Because the two critical notes that aurally determine a G chord are the G itself and the D a fifth above, a Schenkerian would say that the entire melodic span from D to G is being prolonged by the middle note B. Therefore, the entire intervallic hierarchy can be represented by the tree structure shown in Figure 2(a), though it can be more concisely represented by the equivalent structure in Figure 2(b), known as a *maximal outerplanar graph*, henceforth known as a *MOP*.

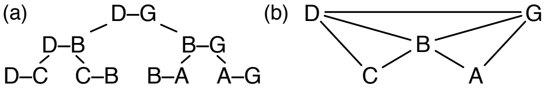

Figure 2: The prolongational hierarchy of a G-major chord with passing tones represented as (a) a tree of melodic intervals, and (b) a MOP.

MOPs were first proposed as elegant structures for representing prolongations in a Schenkerian-style hierarchy by Yust (2006). A single MOP is used to represent a set of prolongations identified in a monophonic sequence of notes, though Yust proposed some extensions for polyphony.

Formally, a MOP is a complete triangulation of a polygon, where the vertices of the polygon are notes and the outer perimeter of the polygon consists of the melodic intervals between consecutive notes of the original music, except for the edge connecting the first note to the last, which is called the *root edge*. Each triangle in the polygon specifies a prolongation. By expressing the hierarchy in this fashion, each edge (x, y) carries the interpretation that notes x and y are "consecutive" at some level of abstraction of the music. Edges closer to the root edge express more abstract relationships than edges farther away.

Probabilistic Model

We now propose a mathematical model of MOPs that allows estimation of the probability that a particular MOP analysis is the best one for a given piece of music.

Each triangle in a MOP triangulation has three endpoints, which we will denote by L, R, and C, corresponding to the left parent note, the right parent note, and the child note, respectively. The assignment of these labels to the notes of a triangle is unambiguous because MOPs are "oriented" by virtue of the temporal dimension: the left endpoint is always the earliest note of the three, while the right endpoint is always the latest. This corresponds exactly to our interpretation of a musical prolongation as described earlier: a prolongation always occurs among exactly three notes, where the middle (child) note prolongs the motion from the left note to the right.

We now define a probability distribution over triangles defined by their three endpoints; we call this the *conditional triangle distribution* $P(C \mid L, R)$. This distribution tells us how likely it is for a given melodic interval (from L to R) to be prolonged by a given child note (C). This distribution has some elegant mathematical properties that relate to the algorithms described in the next section.

We are interested in this distribution because it can be used to build a probabilistic model for an entire analysis in MOP form. Assume we are given a sequence of notes $N = (n_1, n_2, \ldots, n_L)$. We may define the probability of a MOP analysis A as $P(A \mid N)$. Because a MOP analysis is defined by the set of triangles T_{all} comprising the MOP, we will define

$$P(A \mid N) := P(T_{\text{all}}) = P(T_1, T_2, \ldots, T_m).$$

We will derive a method for estimating this probability by using corpus of ground-truth Schenkerian analyses. In particular, we use the SCHENKER41 data set (Kirlin 2014a), which contains 41 excerpts of music and a corresponding Schenkerian analysis for each excerpt. The corpus consists of music from the common practice period of European art music, and includes excerpts by J. S. Bach, G. F. Handel, Joseph Haydn, M. Clementi, W. A. Mozart, L. van Beethoven, F. Schubert, and F. Chopin. The analyses in the corpus are drawn from four sources: three Schenkerian analysis textbooks and one expert music theorist. There are only a handful of Schenkerian analysis textbooks; the three used in SCHENKER41 are the ones most commonly used in beginner-level courses. Historically, it has proven difficult to create corpora with ground-truth Schenkerian analyses due to the time-consuming process of collecting and encoding the analyses; at the moment, the SCHENKER41 data set is the largest digitized publicly-available data set of Schenkerian analyses.

We would like to use this corpus to train a mathematical model to estimate the probability $P(T_1, T_2, \ldots, T_m)$, but the curse of dimensionality prevents us from doing so. Directly using this joint probability distribution as the basis for the model would require many more ground-truth analyses than the 41 in the corpus — and almost certainly more than anyone has available — to get good estimates of the joint probabilities for every combination of triangles. Instead, as an approximation, we assume that the presence of a given type of triangle in a MOP is independent of the presence of all other triangles in the MOP. In other words,

$$\begin{aligned} P(A \mid N) := P(T_{\text{all}}) &= P(T_1, T_2, \ldots, T_m) \\ &= P(T_1) \cdot P(T_2) \cdots P(T_m). \end{aligned} \tag{1}$$

An experiment with synthetic musical data (Kirlin and Jensen 2011) demonstrates that this independence assumption largely preserves the ranking of candidate analyses by their $P(A \mid N)$ calculations.

Given the assumption of independence, we examine methods for using the SCHENKER41 corpus to obtain estimates for the probability of an individual triangle being found in an analysis. A straightforward way to estimate these quantities is to count their frequencies in the corpus and normalize them to obtain a probability distribution. This approach is not feasible due to (a) the size of the SCHENKER41 corpus and (b) the large number of triangle categories needing to be distinguished. The latter condition arises from the wealth of musical information available in the score to guide the analysis process, such as harmonic and metrical information.

In order to create a more accurate model that can handle more features with a smaller corpus, we use random forests (Breiman 2001), an ensemble learning method. Random forests create a collection of decision trees at training time. Each decision tree is only trained on a subset of the features available. The output of the random forest is normally the mode of the output of each individual tree, but we counted the frequencies of the outputs of all the trees and normalized them into a probability distribution instead (Provost and Domingos 2003).

It is straightforward to use random forests for obtaining estimates for the probabilities comprising the conditional triangle distribution $P(C \mid L, R)$: we use features of the left and right endpoints to predict features of the child endpoint in the middle. On the other hand, it is not so straightforward to incorporate multiple features per endpoint into this model. Although we would suspect that many harmonic, melodic, and metrical features of the three endpoints play a role in the Schenkerian analysis process, asking each individual decision tree in a random forest to predict multiple features in the output leads to another curse of dimensionality situation. Therefore, we will factor the conditional model using the rules of conditional probability. Assuming the features of the child note C — the note the random forest is trying to learn a probability distribution over — are denoted C_1 through C_n, we can rewrite $P(C \mid L, R)$ as

$$\begin{aligned} P(C \mid L, R) &= P(C_1, C_2, \ldots, C_n \mid L, R) \\ &= P(C_1 \mid L, R) \cdot P(C_2, \ldots, C_n \mid C_1, L, R) \\ &= P(C_1 \mid L, R) \cdot P(C_2, \mid C_1, L, R) \cdots \\ &\qquad P(C_n \mid C_1, \ldots, C_{n-1}, L, R). \end{aligned} \tag{2}$$

This formulation allows us to model each feature of the note using its own separate random forest. Specifically, we train six total random forests, with each forest learning one feature of the middle note C:

- C_6: The scale degree (1–7) of the note.
- C_5: The harmony present in the music at the time of the note, represented as a Roman numeral I through VII.
- C_4: The category of harmony present in the music at the time of the note, represented as a selection from the set *tonic* (any I chord), *dominant* (any V or VII chord), *predominant* (II, II^6, or IV), *applied dominant*, or *VI chord*. (Our data set did not have any III chords.)
- C_3: Whether the note is a chord tone in the harmony present at the time.
- C_2: The metrical strength of the note's position as compared to the metrical strength of note L.
- C_1: The metrical strength of the note's position as compared to the metrical strength of note R.

Note that the features are labeled C_6 through C_1; this ordering is used to factor the model as described in Equation 2. This ordering of the features is used because the features convey more specific musical information as one moves from from C_1 to C_6, and therefore it makes sense to allow the random forests which are learning the more specific features to use extra training information from the less specific features.

We also used a variety of features for the left and right notes, L and R. These were:

- The scale degree (1–7) of the notes L and R (two features).
- The melodic interval from L to R, with intervening octaves removed.
- The melodic interval from L to R, with intervening octaves removed and intervals larger than a fourth inverted.
- The direction of the melodic interval from L to R; i.e., *up* or *down*.
- The harmony present in the music at the time of L or R, represented as a Roman numeral I through VII (two features).
- The category of harmony present in the music at the time of L or R, represented as a selection from the set *tonic, dominant, predominant, applied dominant,* or *VI chord* (two features).
- Whether L or R was a chord tone in the harmony present at the time (two features).
- A number indicating the beat strength of the metrical position of L or R. The downbeat of a measure is 0. For duple meters, the halfway point of the measure is 1; for triple meters, the one-third and two-thirds points are 1. This pattern continues with strength levels of 2, 3, and so on (two features).
- Whether L and R are consecutive notes in the music.
- Whether L and R are in the same measure in the music.
- Whether L and R are in consecutive measures in the music.

Random forests can be customized by controlling the number of trees in each forest, how many features are used per tree, and each tree's maximum depth. We use forests containing 1,000 trees with a maximum depth of four. We used Breiman's original idea of choosing a random selection of $m = \text{int}(\log_2 M + 1)$ features to construct each individual tree in the forest, where M is the total number of features available to us. In our case, $M = 16$, so $m = 5$.

Algorithms

We have a created an algorithm, known as ParseMop, that accepts a monophonic string of notes N as input and produces the most probable MOP analysis A by maximizing $P(A \mid N)$ according to Equations 1 and 2. ParseMop is based on the CYK algorithm used to parse context-free grammars, adapted to both (1) take probabilities into account, and (2) permit *ranking* the parses efficiently rather than just finding the single most probable parse (Jurafsky and Martin 2009; Jiménez and Marzal 2000). ParseMop uses dynamic programming to optimally triangulate successively larger sections of the MOP, and runs in $O(n^3)$ time, where n is the number of input notes.

We examined three variations of the ParseMop algorithm that differ in how they determine the correct analysis of the "upper levels" of the musical hierarchy. In particular, Schenkerian analyses usually incorporate one of three specific melodic patterns at the most abstract level of the musical hierarchy. Heinrich Schenker, the originator of the analysis method that bears his name, hypothesized that because of the way listeners perceive music centered around a given pitch (i.e., *tonal* music), *every* tonal composition could be viewed as an elaboration of one of the three possible patterns, and therefore should be reducible to that specific pattern. The three patterns are specific stepwise descending melodic sequences, each known as an *Urlinie*. The *Urlinie* concept is Schenker's attempt to show how seemingly very different pieces can grow out of a small set of basic melodic structures via prolongations (Pankhurst 2008). This idea has proved much more controversial than that of a structural hierarchy of notes within the music.

The first variation of the parsing algorithm, known as ParseMop-A, functions identically at all levels of the musical hierarchy and does not force the background structure of an analysis to conform to an *Urlinie* pattern or to any specific contour. Unsurprisingly, analyses produced by ParseMop-A often fail to find an *Urlinie* even when there is one in the ground-truth interpretation of a note sequence.

In contrast, the ParseMop-B algorithm accepts not only a sequence of notes as input, but also information specifying exactly which notes of the input sequence should be part of the background structure (the *Urlinie* and any initial ascent or arpeggiation). The dynamic programming formulation in ParseMop-B forces all parses to place these notes at the highest levels of the musical hierarchy, thus insuring that the algorithm will always produce a most-probable analysis with a correct background structure. This variation of ParseMop is, of course, only useful in situations where this background information can be determined beforehand.

The ParseMop-C algorithm is a compromise between the A and B algorithms to better reflect the real-world scenario of being able to identify the contour of the correct background musical structure for a piece ahead of time but not which specific notes of the piece will become part of that structure. While the input to ParseMop-B is a sequence of notes, some of which are specifically identified as belonging to the background structure, ParseMop-C accepts the same note sequence but along with only the *names* of the notes, in order, that belong to the background melodic structure. For example, given the sequence of notes $\underline{\text{E}}$–F–D–C–E–$\underline{\text{D}}$–B–$\underline{\text{C}}$ with the correct background structure underlined, ParseMop-B must be informed ahead of time that the first, sixth, and eighth notes of the input must appear at the uppermost levels of the musical hierarchy. ParseMop-C, on the other hand, is provided only with the information that the background must contain the notes E–D–C in that order, but will not know *which* E, D, and C are the "correct" notes.

Evaluation and Discussion

We now evaluate the quality of our probabilistic model of music analysis by studying the analyses that the ParseMop algorithms produce for the music in the Schenker41 corpus. We also evaluate the utility of providing ParseMop with prior information about the structure of the *Urlinie*. Specifically, we show the results of four experiments, which (a) quantitatively compare analyses produced by ParseMop to corresponding analyses from textbooks, (b) show the locations within the analyses produced by ParseMop where the algorithm is more likely to make a mistake, (c) illustrate how the accuracy of the analyses produced by ParseMop changes as one moves through a list of analyses ranked by probability, and (d) use experienced music theorists to judge the analyses produced by ParseMop.

Due to the difficulty of finding additional musical excerpts with corresponding analyses to use as a testing data set, coupled with the small size of the corpus (41 musical excerpts), we used a leave-out-one cross-validation approach for training and testing in these experiments. Specifically, for each excerpt in the corpus, we generated a training set consisting of the music from the other 40 excerpts, trained the probabilistic model on these data, and used each ParseMop variant to produce a list of the top 500 analyses for the original excerpt that was left out.

Experts in Schenkerian analysis sometimes disagree on the "correct" analysis for a composition. It is therefore possible to have more than one musically-plausible hierarchical analysis of an excerpt; occasionally these analyses will differ radically in their hierarchical structures. However, due to limited data, our experiments rely on using a single analysis as ground truth.

Evaluation metrics

We define the *edge accuracy* of a candidate MOP as the percentage of edges in the candidate MOP that match with an edge in the equivalent reference MOP from Schenker41. We use this metric rather than the percentage of entire matching triangles because it is possible for two MOPs to have edges in common but no triangles in common.

Calculating the edge accuracy for a candidate MOP against a reference MOP is straightforward except for the possibility of untriangulated subregions inside the reference MOPs (candidate MOPs produced by ParseMop are always fully-triangulated). Humans experts sometimes leave out small details in an analysis when including such details would detract from the overall presentation of the musical hierarchy. Though this is common in textbooks, it results in reference MOPs in SCHENKER41 that are sometimes not fully triangulated.

To handle this situation, we modify our definition for "matching" edges as follows. A "matching" edge in a candidate MOP is an edge that either (1) matches exactly with an edge in the reference MOP, or (2) could fit in an untriangulated area of the reference MOP. In other words, edges are matches if they appear in the reference MOP or could hypothetically appear, if the reference MOP were completely triangulated.

While it may seem that compensating for untriangulated regions in this fashion could distort accuracy statistics, we take this into account by providing the edge accuracy for a randomized reference triangulation, providing a baseline level of accuracy for comparison. Furthermore, untriangulated regions account for only 183 of the 907 triangles in the corpus (about 20%). The baseline percentage of edge matches, along with the edge accuracy scores for the three PARSEMOP algorithms is shown in Figure 3. The relative accuracies, unsurprisingly, align with the amount of prior information given to PARSEMOP.

It is also instructive to examine, on a piece-by-piece basis, how much better PARSEMOP performs over the baseline algorithm. We can determine this value for a piece by normalizing the improvement over the baseline obtained by PARSEMOP relative to the hypothetical maximum improvement. In other words, if PARSEMOP produced an analysis matching 8 out of 10 edges and the baseline algorithm matched 6 out of 10 edges, then the normalized improvement would be 50%, because PARSEMOP improved by 2 edges out of a possible 4. A histogram of these normalized percents is shown in Figure 4.

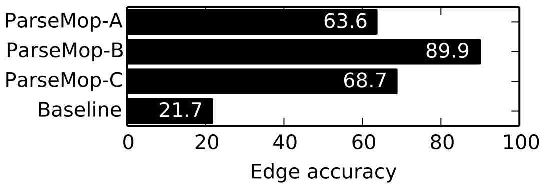

Figure 3: Edge accuracies for the three PARSEMOP algorithms and the baseline randomized algorithm.

Error locations

In addition to studying what kinds of errors PARSEMOP makes, it is worthwhile to identify where the errors are being made. In other words, we would like to know if PARSEMOP is making more mistakes at the surface level of the music or at the more abstract levels. We can quantify the notion of "level" by numbering the interior edges of a candidate MOP produced by PARSEMOP with increasing integers starting from zero, with zero corresponding to the most abstract interior edge. Perimeter edges are not assigned a level because they are always present in any MOP and therefore cannot be in error. These assigned integers correspond to the standard idea of vertex depth in a tree, and therefore can be regarded as *edge depths*. We will normalize these edge depths by dividing each integer by the maximum edge depth within the MOP, giving an error location always between 0 and 1, with 0 corresponding to the most abstract edges, and 1 corresponding to those closest to the musical surface.

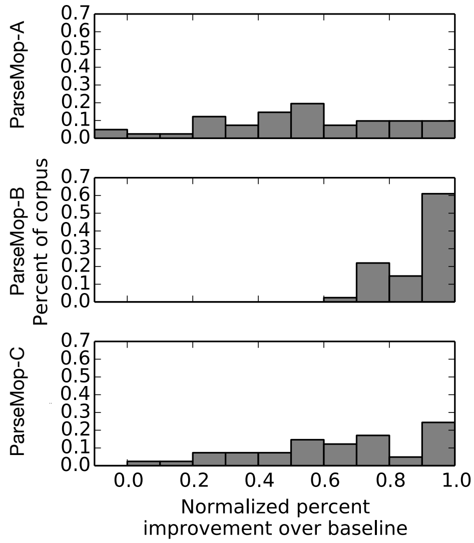

Figure 4: Histogram tallying, for each piece, the normalized percent improvement over the baseline algorithm.

Figure 5 shows the probability for the three PARSEMOP variants to include an incorrect edge at different normalized error depths. Unsurprisingly, the probability of making an error at the most abstract level corresponds exactly to how much extra information the PARSEMOP variant is given about the contour of the main melody of the musical excerpt in question: PARSEMOP-B has the lowest probability for an error at depths between 0.0 and 0.2 (the highest levels of the hierarchy), while PARSEMOP-A has the largest.

Maximum accuracy as a function of rank

So far we only have examined the accuracy of the top-ranked analysis produced by the PARSEMOP algorithms for each musical excerpt. However, it is instructive to examine the accuracies of the lower-ranked analyses as well, in order to investigate how accuracy relates to the ranking of the analyses. In particular, we are interested in studying the maximum accuracy obtained among the analyses at ranks 1 through n, where n is allowed to vary between 1 and 500. We would hope that analyses that are judged as being accurate are not buried far down in the rankings, especially when the top-ranked analysis is not perfectly accurate.

Figure 6 illustrates how the maximum accuracy changes for each musical excerpt as one moves through the ranked list. The relative quality of the results for the three PARSEMOP variants reflects the amount of *a priori* information provided to each variant. A few oddities in the graphs are worth mentioning. A single musical excerpt appears to

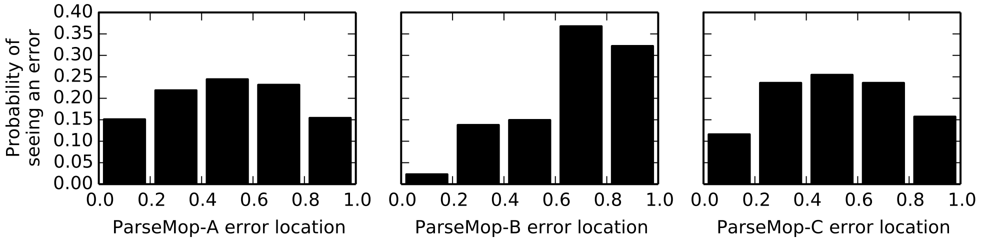

Figure 5: Histogram showing the locations of errors for the three PARSEMOP variants over all excerpts in the corpus.

present problems for PARSEMOP-A; it can be seen as the single line in the PARSEMOP-A graph that is clearly isolated below the other lines. This piece is a Mozart minuet that is particularly difficult to analyze due to repeated notes in the input and the pitches of the *Urlinie* being hidden in an inner voice. Interestingly, the isolated lowest line in the PARSEMOP-C graph corresponds to a different Mozart excerpt: this one is complicated to analyze due to its highly figurated texture and the notes of the *Urlinie* being often located on metrically weak beats, a rather uncommon situation that PARSEMOP likely learns to avoid identifying.

Human-based evaluation

While it is useful to examine mathematical accuracy metrics, there is no substitute for having human evaluations of the PARSEMOP analyses. In particular, having experienced music theorists evaluate the analyses is indispensable because humans can make both qualitative and quantitative judgements that the accuracy metrics cannot.

We recruited three expert music theorists to assist with an experiment in which each expert graded pairs of analyses. The order of the musical excerpts in the corpus was randomized and for each excerpt, the graders were provided with the music notation of the excerpt itself, along with the corresponding PARSEMOP-C analysis and the textbook analysis. The graders were not given any information on the sources of the analyses; in particular, they did not know that the two analyses within each pair came from very different places. Furthermore, the order in which the two analyses of each pair were presented on the page was randomized so that sometimes the PARSEMOP analysis was presented first and sometimes second. Both analyses were displayed using a pseudo-Schenkerian notation scheme that uses slurs to illustrate prolongations and beams to show the notes of the main melody and other hierarchically-important notes. It is important to note that the textbook analyses were also presented using this notation style; this was done because the output of the algorithm which translates MOPs to pseudo-Schenkerian notation does not yet rival the true Schenkerian analytic notation used by humans, and so reproducing the textbook analyses verbatim would be too revealing to the graders.

The graders were instructed to evaluate each analysis in the way they would evaluate "a student homework submission." Each grader was asked to assign a letter grade to each analysis from the set A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F, according to a grading scheme of their own choosing. The goal of this experiment was to determine how much the PARSEMOP-C analyses differ in quality from their textbook counterparts. Figure 7 illustrates how, on average, the graders judged analyses when the A–F grades are converted to a standard 4.0 grading scale (A $= 4$, B $= 3$, C $= 2$, D $= 1$, F $= 0$. A "plus" adds an additional 0.3 points and a "minus" subtracts 0.3 points). The average differences (the amount the textbook analyses are preferred over the PARSEMOP analyses) indicate that the graders preferred the textbook analyses by somewhere between half a letter grade and a full letter grade. This relationship is also illustrated in Figure 8, which directly compares the grade of each PARSEMOP analysis with the grade of the corresponding textbook analysis.

Additional details of the evaluation procedures and further discussion of the results are available in Kirlin (2014b).

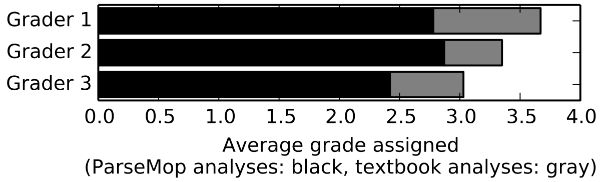

Figure 7: Average grades assigned by each grader to the textbook and PARSEMOP analyses.

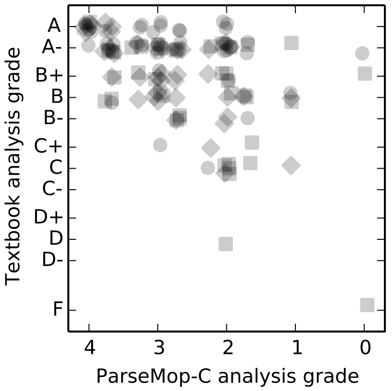

Figure 8: This graph plots grades (jitter added) for each PARSEMOP-C and textbook analysis pair. Different shapes correspond to different graders.

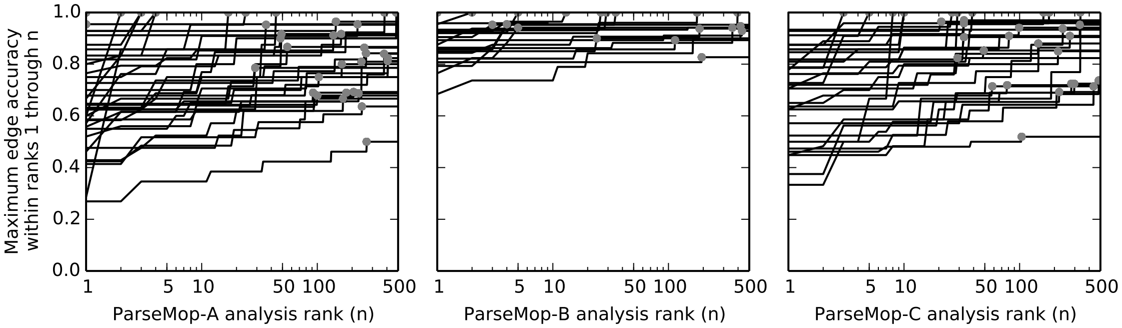

Figure 6: Each graph shows the maximum accuracy obtained within the top n analyses, as n increases from 1 to 500. The gray dots indicate the earliest rank at which the maximum accuracy (within the top 500 analyses) was obtained.

Summary and Future Work

In this paper, we have presented the first large-scale data-driven study of modeling Schenkerian analysis. We have shown a probabilistic model of the analysis procedure, an algorithm for identifying the most probable analysis for a piece of music, and studies illustrating the performance of the model and algorithm from both quantitative and human-centric perspectives.

The probabilistic model presented has extensive potential for further research and application. A logical next step for improving the model would be to include support for multiple voices, as the model currently assumes all of the notes of the input music constitute a single monophonic voice. A more sophisticated model capable of correctly interpreting polyphonic music — music with multiple notes sounding at once — would be extremely desirable.

References

Breiman, L. 2001. Random forests. *Machine Learning* 45(1):5–32.

Cook, N. 2005. Towards the compleat musicologist? Invited talk, Sixth International Conference on Music Information Retrieval.

Frankel, R. E.; Rosenschein, S. J.; and Smoliar, S. W. 1978. Schenker's theory of tonal music—its explication through computational processes. *International Journal of Man-Machine Studies* 10(2):121–138.

Gilbert, É., and Conklin, D. 2007. 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*, 83–94.

Jiménez, V. M., and Marzal, A. 2000. Computation of the n best parse trees for weighted and stochastic context-free grammars. In Ferri, F. J.; Iñesta, J. M.; Amin, A.; and Pudil, P., eds., *Advances in Pattern Recognition*, volume 1876 of *Lecture Notes in Computer Science*. Springer-Verlag. 183–192.

Jurafsky, D., and Martin, J. H. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, second edition.

Kassler, M. 1975. 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.

Kirlin, P. B., and Jensen, D. D. 2011. Probabilistic modeling of hierarchical music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 393–398.

Kirlin, P. B. 2014a. A data set for computational studies of Schenkerian analysis. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, 213–218.

Kirlin, P. B. 2014b. *A Probabilistic Model of Hierarchical Music Analysis*. Ph.D. Dissertation, University of Massachusetts Amherst.

Marsden, A. 2009. "What was the question?": Music analysis and the computer. In Crawford, T., and Gibson, L., eds., *Modern Methods for Musicology*. Farnham, England: Ashgate. 137–147.

Marsden, A. 2010. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research* 39(3):269–289.

Mavromatis, P., and Brown, M. 2004. Parsing context-free grammars for music: A computational model of Schenkerian analysis. In *Proceedings of the 8th International Conference on Music Perception & Cognition*, 414–415.

Pankhurst, T. 2008. *SchenkerGUIDE: A Brief Handbook and Website for Schenkerian Analysis*. New York: Routledge.

Provost, F., and Domingos, P. 2003. Tree induction for probability-based ranking. *Machine Learning* 52(3):199–215.

Volk, A.; Wiering, F.; and van Kranenburg, P. 2011. Unfolding the potential of computational musicology. *Proceedings of the International Conference on Informatics and Semiotics in Organisations* 137–144.

Yust, J. 2006. *Formal Models of Prolongation*. Ph.D. Dissertation, University of Washington.