2017-04-04

Analysis of analysis: importance of different musical parameters for Schenkerian analysis

Phillip B. Kirlin & Jason Yust (2016) Analysis of analysis: Using machine learning to evaluate the importance of music parameters for Schenkerian analysis, Journal of Mathematics and Music, 10:2, pp. 127-148. https://doi.org/10.1080/17459737.2016.1209588

https://hdl.handle.net/2144/39133

Submitted exclusively to the *Journal of Mathematics and Music*
Last compiled on November 16, 2015

Analysis of analysis: Using machine learning to evaluate the importance of music parameters for Schenkerian analysis

Phillip B. Kirlin[a]* and Jason Yust[b]

[a]*Department of Mathematics and Computer Science, Rhodes College, Memphis, USA;*
[b]*School of Music, Boston University, Boston, USA*

()

While criteria for Schenkerian analysis have been much discussed, such discussions have generally not been informed by data. Kirlin (2014b) has begun to fill this vacuum with a corpus of textbook Schenkerian analyses encoded using data structures suggested by Yust (2006), and a machine learning algorithm based on this dataset that can produce analyses with a reasonable degree of accuracy. In this work, we examine what musical features (scale degree, harmony, metrical weight) are most significant in the performance of Kirlin's algorithm.

Keywords: Schenkerian analysis, machine learning, harmony, melody, rhythm, feature selection

1. Introduction

Schenkerian analysis is widely understood as central to the theory of tonal music. Yet, many of the most prominent voices in the field emphasize its status as an expert practice rather than as a theory. Burstein (2011, 116), for instance, argues for preferring a Schenkerian analysis "not because it demonstrates features that are objectively or intersubjectively present in the passage, but rather because I believe it encourages a plausible yet stimulating and exciting way of perceiving and performing [the] passage." Rothstein (1990, 298) explains an approach to Schenker pedagogy as follows:

> Analysis should lead to better hearing, better performing, and better thinking about music, not just to "correct" answers. [. . .] I spend lots of class time—as much as possible—debating the merits of alternative readings: not primarily their conformance with theory, though that is discussed where appropriate, but their relative plausibility as models of the composition being analyzed.

Schachter (1990) illustrates alternative readings of many works and asserts that a full musical context is essential to evaluating them. Paring the music down to just aspects of harmony and voice leading, like "the endless formulas in white notes that disfigure so many harmony texts," he claims, leaves the difference between competing interpretations undecidable. In publications such deliberation typically occurs at a high level. It rarely addresses the implicit principles used to deal with many details of the musical surface. As Agawu (2009, 116) says, "the journey from strict counterpoint to free composition makes an illicit or—better—mysterious leap as it approaches its destination."

*Corresponding author. Email: kirlinp@rhodes.edu

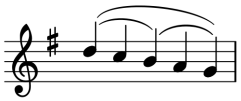
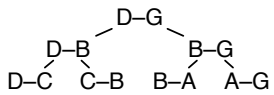
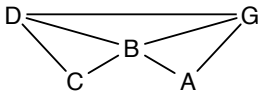


Figure 1. A melodic line illustrating prolongations.



(a) A binary tree of melodic intervals.



(b) A maximal outerplanar graph, or MOP.

Figure 2. The prolongational hierarchy of a G-major chord with passing tones represented as two equivalent data structures.

As with any complex human activity, the techniques of artificial intelligence may greatly advance our understanding of how Schenkerian analysis is performed and what kinds of implicit cognitive abilities and priorities support it. The present work builds upon the research of Kirlin (2014a; 2015) which models Schenkerian analysis using machine learning techniques. By probing Kirlin's algorithm we address a question of deep interest to Schenkerian analysts and pedagogues: what role do different aspects of the music play in deliberating between possible analyses of the same musical passage? Because the activity of Schenkerian analysis involves such a vast amount of implicit musical knowledge, it is treacherous to litigate this question by intuition, without the aid of computational models and methods.

The second section explains the machine learning algorithm we used, which is essentially that of Kirlin (2014b). The third section explains an experiment to test which musical features the algorithm relies upon most heavily to produce accurate analyses. The fourth section provides the results of that experiment, and the fifth provides further exploratory analysis of data produced by the experiment.

2. A machine learning algorithm for Schenkerian analysis

Schenkerian theory is grounded in the idea that a tonal composition is organized as a hierarchical collection of *prolongations*, where a prolongation is understood, for our purposes, as an instance where a motion from one musical event, L, to another non-adjacent event, R, is understood to control the passage between L and R, and the intermediate events it contains. A prolongation is represented in Schenkerian notation as a slur or beam.

Consider Figure 1, a descending melodic passage outlining a G major chord. Assuming this melody takes place over G-major harmony, this passage contains two passing tones (non-harmonic tones in a melodic line that linearly connect two consonant notes via stepwise motion), the second note C and the fourth note A. These tones smoothly guide the melody between the chord tones D, B, and G. In Schenkerian terminology, the C prolongs the motion from the D to the B, and the A similarly prolongs the motion from the B to the G.

The hierarchical aspect of Schenkerian analysis comes into play when we consider a prolongation that occurs over the entire five-note passage. The slurs from D to B and B

to G identify the C and A as passing tones. Another slur from D to G, which contains the smaller slurs, shows that the entire motion outlines the tonic triad from D down to G. The placement of slurs may reflect the relatively higher stability of the endpoints (between chord tones over non-chord tones, and between more stable members of the triad, root and fifth, over the third), or they may reflect a way in which the local motions (passing-tone figures) group into the most coherent larger-scale motion (arpeggiation of a triad).

This hierarchy can be represented visually by the tree in Figure 2(a): this diagram illustrates the hierarchy of melodic intervals present in the composition and the various prolongations identified above. An equivalent representation, known as a maximal outerplanar graph, or MOP (Yust 2006, 2009, 2015), is shown in Figure 2(b). Binary trees of intervals and MOPs are duals of each other in that they represent identical sets of information, though the MOP representation is more succinct.

From a mathematical perspective, a MOP is a complete triangulation of a polygon. Each triangle in a MOP represents a single melodic prolongation among the three notes of the music represented by the three endpoints of the triangle. Because MOPs are oriented temporally, with the notes shown in a MOP always ordered from left to right as they are in the musical score, we can unambiguously refer to the three endpoints of a triangle in a MOP as the left (L), center (C), and right (R) endpoints. Each triangle in a MOP, therefore, represents a prolongation of the melodic interval from L to R by the intervals from L to C and C to R.

2.1. *A probabilistic interpretation of MOPs*

Our goal is to develop an algorithm with the ability to predict, given a musical composition, the "correct" Schenkerian analysis for that composition. We develop this algorithm using the following probabilistic perspective.

Assume that we are given a sequence of notes N that we wish to analyze, and that all possible Schenkerian analyses of N can be enumerated as $A_1, \ldots, A_m$, for some integer m. We desire the the most probable analysis given the notes, which is $\arg\max_i P(A_i \mid N)$. Because a Schenkerian analysis can be represented in MOP form by its collection of triangles $T_1, \ldots, T_k$, for some integer k, we define $P(A_i \mid N)$ as $P(T_1, \ldots, T_k)$. In other words, to compute the probability of a certain analysis given a sequence of notes, we reduce this to computing the probability of observing a MOP analysis containing the specific set of triangles $T_1, \ldots, T_k$ that comprise analysis A_i.

Previous work (Kirlin and Jensen 2015) illustrates that it is possible to derive an estimate for the probability distribution above using machine learning. We begin by considering the SCHENKER41 dataset, the largest known corpus of machine-readable Schenkerian analyses in existence (Kirlin 2014a). This dataset contains 41 common-practice era musical excerpts and their corresponding Schenkerian analyses. The excerpts in the data set are derived from four sources: Forte and Gilbert's *Introduction to Schenkerian Analysis* and the accompanying solutions manual (1982b; 1982a) (10 excerpts), Cadwallader and Gagné's *Analysis of Tonal Music* (1998) (4 excerpts), Pankhurst's *SchenkerGUIDE* (2008) (24 excerpts), and an additional individual expert music analyst (3 excerpts).

Each excerpt in the data set can be translated into a MOP representing the prolongations present in the main melody of the excerpt. Though the conversion to MOPs is straightforward, it is not as simple to use the resultant triangle frequencies in the MOPs to derive an estimate for the full joint probability distribution $P(T_1, \ldots, T_k)$ using standard machine learning techniques due to the curse of dimensionality: the vast number of

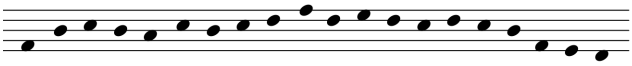


Figure 3. A melody with general interval information only.

combinations of triangles is simply too large. Instead, we make the simplifying assumption that each triangle in a MOP is independent of all other triangles in a MOP, which implies that $P(T_1, \ldots, T_k) = P(T_1) \cdots P(T_k)$. This assumption reduces the full joint distribution to a product of simpler, lower-dimensional distributions, which are easier to learn. An experiment verifies that this assumption largely preserves relative probability scores between two MOPs, which is a sufficient condition for our purposes to proceed (Kirlin and Jensen 2011).

We define the probability of an individual triangle appearing in a MOP as the probability of a given melodic interval being elaborated by the specific choice of a certain child note. Mathematically, we define $P(T_i) = P(M_i \mid L_i, R_i)$ where L_i, M_i, and R_i are the three endpoints of triangle T_i. We use random forests (Breiman 2001; Provost and Domingos 2003), an ensemble learning method, to learn this conditional triangle distribution $P(M_i \mid L_i, R_i)$. More precisely, we use features of the left and right endpoints to predict features of the middle point. Details of the mathematical formulation are described in Kirlin and Jensen (2015), but we will describe the various features that our probabilistic model takes into account. The major division of the features is between those that describe the left and right endpoints (L and R), and those that describe the middle point (M).

With an appropriate estimate of the probability of seeing any particular triangle in a MOP, we can calculate the probability of an entire MOP analysis given a sequence of notes using the ParseMop-C algorithm (Kirlin and Jensen 2015). This algorithm relies on the probabilistic interpretation of MOPs described above, along with the equivalence between MOPs and binary trees, to view each prolongation within a MOP as a production in a probabilistic context-free grammar. Under this interpretation, it is straightforward to use standard parsing techniques (Jiménez and Marzal 2000; Jurafsky and Martin 2009), to develop an $O(n^3)$ algorithm that can determine the most probable analysis for a given sequence of notes. Additionally, the grammar formalism allows us to restrict the predicted analyses to those that contain a valid *Urlinie* through specific sets of production rules. We can compare the predicted output analyses of ParseMop-C against the ground-truth analyses from the Schenker41 corpus to determine the performance of the algorithm, which we measure via *edge accuracy*, which is the percent of edges in an algorithmically-produced MOP that correspond to edges in the ground-truth MOP.

3. Musical Features

To understand how individual musical features might figure into the accuracy of these algorithmic analyses, consider the task of analyzing the "music" in Figure 3. This is the intervallic pattern of the melody of a simple four-measure phrase of real music. There is no information about the rhythm or harmony of the original music. We do not know what the key is or which note is the tonic. Registral information is also removed by octave reducing melodic intervals and inverting larger leaps. How likely would it be that one could reproduce a textbook analysis of the melody relying on this intervallic pattern only?

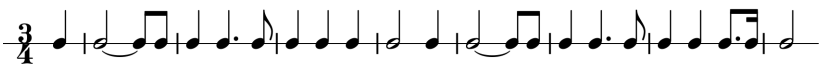


Figure 4. The meter and rhythm of a melody.

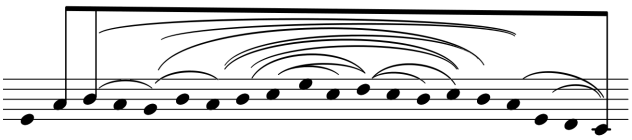


Figure 5. An analysis of the intervallic pattern in Figure 3, by a computer trained on just the intervallic patterns of textbook analyses.

Or what if one were given only the rhythm and meter of a melody, as in Figure 4? How accurately could we expect to predict a Schenkerian analysis of the passage?

In the experiment described below, we train the machine learning algorithm on Schenkerian analyses with differing amounts of information about the music, and therefore discover how important different aspects of the music (melodic, harmonic, metric) are to accurately reproducing Schenkerian analyses.

A MOP analysis of Figure 3 is equivalent to adding slurs between the notes such that no two slurs cross, until it is impossible to add any more. Figure 5 shows the solution arrived at by the algorithm trained on just the generic melodic intervals for a corpus of textbook analyses. An analysis of Schubert's Impromptu, Op. 142 No. 1, from the Cadwallader and Gagné (1998) textbook (Figure 6) has enough detail to be translated into the explicit and near-complete MOP shown in Figure 7. The MOP is a simplification, of course, reflecting just the basic melodic hierarchy implied by Cadwallader and Gagné's analysis. There are three places where their analysis is ambiguous: In measures 1–3, they show a double neighbor figure A♭-G-B♭-A♭, where a slur could be added from A to B♭ or from G to A♭, but neither is clearly implied by the analysis. In measures 5–8, they show C-B♭-A♭-C in stemmed notes. A slur from C to A♭ might be inferred but is not actually shown, so it is not included in the coded analysis. Finally, this example reflects a common problem in that Cadwallader and Gagné give the status of $\hat{3}$ to two different Cs. A fully explicit analysis would choose between these as the true initiation of the fundamental line. The part of the analysis shown with beams is the "background" of the analysis; the algorithm is constrained to find such a $\hat{1}$-$\hat{2}$-$\hat{3}$ initial ascent as the background of the phrase (but not precisely where these notes occur). But the rest of the analysis reflects intervallic patterns learned from a corpus.

The algorithm accurately identifies some local passing motions in the music given just the basic intervallic information. Evaluated by the proportion of shared slurs between the computer analysis and the textbook one, however, the computer does not do especially well on this example, mostly because it buries the C of measure 4 in the middle of a B♭-C-D♭ passing motion, and thus most of its slurs cross over a note that is especially prominent in the textbook analysis. This passing motion, which is perfectly reasonable given only intervallic information, is quite implausible when we see the metric and harmonic status of the three notes involved. On other examples, however, the algorithm performs surprisingly well with such minimal musical information, as we shall see below.

Given a slightly richer musical object, including the scale-degree number of each note (i.e., it has the reference point of a key), and some harmonic context (Roman numeral without inversion) the algorithm produces the analysis of Figure 8. The new information

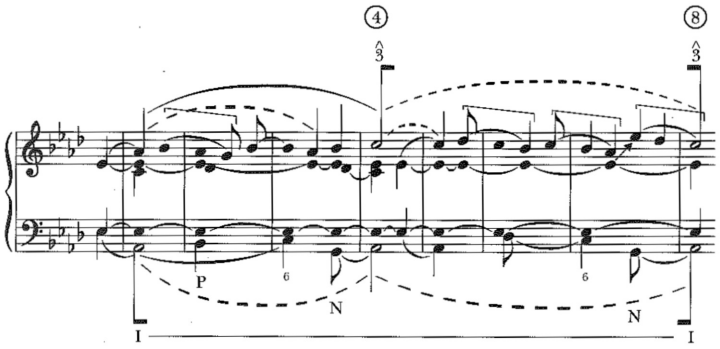


Figure 6. Cadwallader and Gagné (1998)'s analysis of Schubert's Impromptu, Op. 142 No. 2.

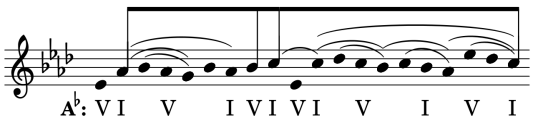


Figure 7. An encoded version of the analysis in Figure 6.

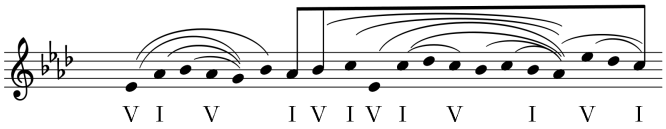


Figure 8. Computer analysis of the melodic (interval and scale degree) and harmonic information of the Schubert Impromptu.

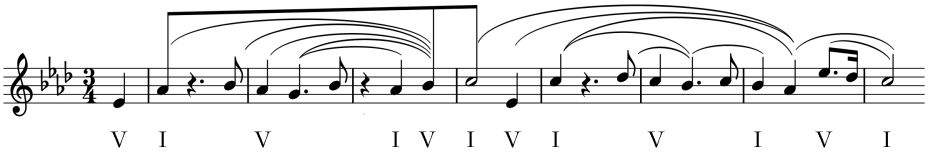


Figure 9. Computer analysis of the melodic, harmonic, and metrical data of the Schubert Impromptu.

allows it to avoid certain blatant errors: for instance, now that it knows that the third note (B♭) occurs over a tonic chord, not a V, it avoids assigning it a major structural role. However, the algorithm makes another decision that turns out to be a mistake, shifting the first note of the structure ahead to note 7. It apparently identifies an arpeggiation of V as a likely introductory approach the first structural note, but that turns out to be implausible because of the rhythm. When it has additional metrical information (specifically, which notes are accented relative to others), the algorithm corrects this mistake, shifting the structural beginning back to the metrically strong note 2, as shown in Figure 9. This result is then quite similar to Cadwallader and Gagné's analysis.

This single example shows how different aspects of the music—melodic pattern, harmony, rhythm—play different roles in determining the plausibility of a Schenkerian analysis. The principal goal of the experiment reported here is to answer the broad question of what features of the music are most essential to accurately reproducing human analyses.

The main result (reported in Section 4) is an ordering of various musical features according to how much the performance of the machine learning algorithm depends upon them. However, the significance of different features is not simple and additive, and the experiment also produces a large amount of data that can be further mined for information about how the features interact with one another. Some exploration and interpretation of that data is reported in Section 5.

We define 18 features in all that are available to the algorithm in the full model. Six of these are features of the middle note:

- SD-M: The scale degree of the note (represented as an integer from 1 through 7, qualified as raised or lowered for altered scale degrees),
- RN-M: The harmony present in the music at the time of onset of the center note (represented as a Roman numeral from I through VII),
- HC-M: The category of harmony present in the music at the time of the center 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.)
- CT-M: Whether the note is a chord tone in the harmony present at the time (represented as a true/false value),
- MS-LMR: The metrical strength of the center note's position as compared to the metrical strength of note L, and to the metrical strength of note R.
- Int-LMR: The melodic intervals from L to M and from M to R, generic (scale-step values) and octave generalized (ranging from a unison to a seventh).

Note that two features of the middle note are different than the others, in that they are influenced by the left and right notes. These are therefore distinguished as "LMR" features, as opposed to simple "M" features.

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

- SD-LR: The scale degree (1–7) of the notes L and R (two features).
- Int-LR: The melodic interval from L to R, with intervening octaves removed.
- IntI-LR: The melodic interval from L to R, with intervening octaves removed and intervals larger than a fourth inverted.
- IntD-LR: The direction of the melodic interval from L to R; i.e., up or down.
- RN-LR: The harmony present in the music at the time of L or R, represented as a Roman numeral I through VII (two features).
- HC-LR: 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).
- CT-LR: Whether L or R was a chord tone in the harmony present at the time (two features).
- MS-LR: A number indicating the beat strength of the metrical position of L or R. The downbeat of a measure is 0. For duple or quadruple meters, the halfway point of the measure is 1; for triple meters, beats two and three are 1. This pattern continues with strength levels of 2, 3, and so on (two features).
- Lev1-LR: Whether L and R are consecutive notes in the music.
- Lev2-LR: Whether L and R are in the same measure in the music.
- Lev3-LR: Whether L and R are in consecutive measures in the music.

We used the ParseMop-C algorithm with leave-one-out cross-validation to compute new MOP analyses for all the pieces in the Schenker41 corpus longer than four measures, then calculated the overall edge accuracy for all the MOPs combined. We then

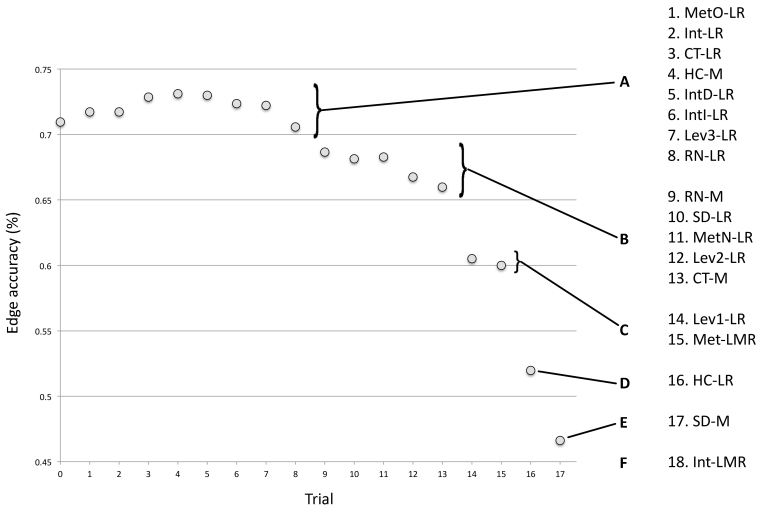


Figure 10. Performance of the algorithm on trials 0–17 and the feature removed prior to each trial.

cycled through each feature or pair of related features from the training data and re-ran the experiment to find the feature that, when omitted, decreased the overall accuracy the least. This feature was then permanently removed and another cycle was performed, until only one feature remained. For the entire set of 18 features we then had 18 trials, numbered 0–17, where trial 0 included all 18 features in the full model, and trial 17 (trivially) included only the last remaining feature.

4. Main results

The main result of the experiment is an ordering of the features by *expendability*: how useful the feature is in the context of other features. Figure 10 shows which feature was dropped on each trial. At certain points in the process there are larger changes in baseline edge accuracy, most notably on trials 14, 16, and 17. This suggests that between these points are more significant differences in feature importance, and the features can be sorted into groups between these critical points.

As a check on the robustness of this result we also averaged, across all trials including the given feature, the decrease in performance observed after removing that feature. These data are shown in Figure 11. The ordering obtained by this measure is perfectly consistent with the grouping of features in Figure 10, suggesting that this grouping provides a fairly robust partial ordering of the features by expendability.[1] The ordering within the large groups, A and B, is not consistent between the two measures of expendability.

One striking aspect of the result is that, if we sort the features by type—melodic, harmonic, metrical, and temporal—the five features in groups C–F represent all four types, with a duplication for melodic features (Int-LMR and SD-M). In addition, the five features in group B also represent all types, with an additional harmonic feature (CT-M

[1]These two measures are confounded: for the features that remain in the model for longer, there are more trials to average over, and the drop in performance predictably gets higher as the number of features in the full model gets smaller. The similarity of the two rankings can therefore be partly featured to this confound, but not entirely so. The average rise of feature importance from trial to trial is just 0.12%, and this is mostly due to the last four trials. On trials 0–13 the average change is 0.02%.

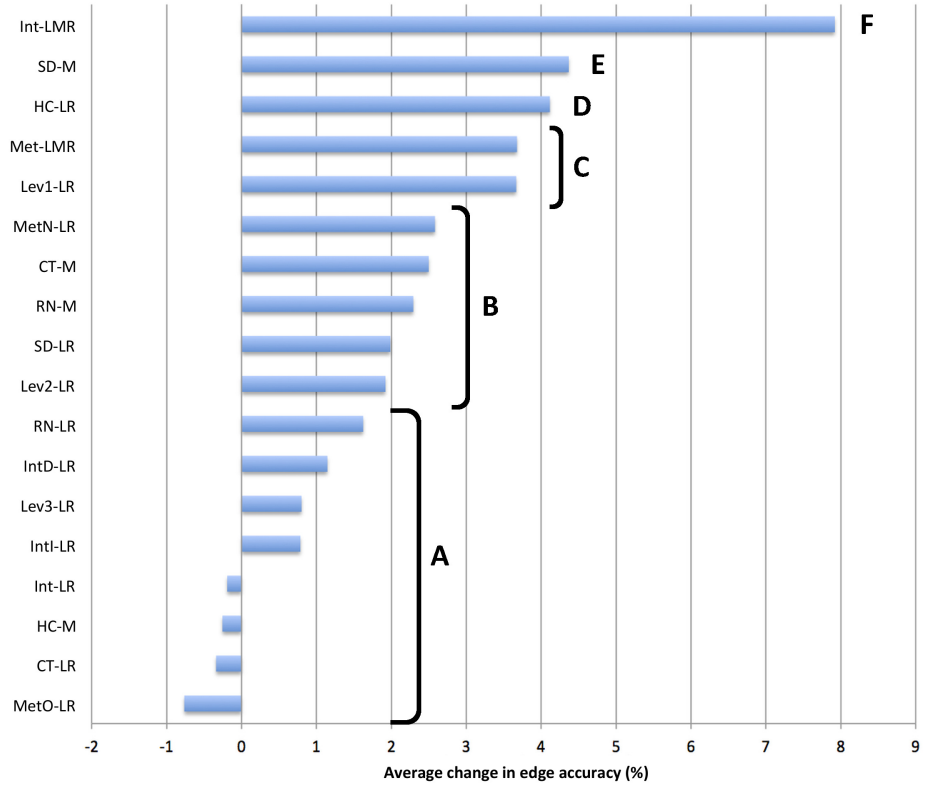


Figure 11. Average feature importance for all features over the course of the experiment.

and RN-M). Since the ordering of groups C–F appears robust, while that within group B is not, we can broadly characterize the results as follows:

- Melodic, harmonic, metrical and temporal information all contribute to the analytical process.
- Of these, melodic features are overwhelmingly the most important, harmonic features the next most important, and metrical and temporal features the least important.
- For each of the feature types, except melodic, the majority of the necessary information is captured by a single feature. Yet, in all cases, some additional valuable information is provided by one or two other features (group B). In the case of melodic features, these can be subdivided into purely intervallic information (Int-LMR) and tonally anchored information (SD-M).

The result therefore lends itself to discussion by feature:

4.1. *Melodic features*

Melodic features proved by far to be the primary analytical consideration. The algorithm obtains over 50% edge accuracy, well above the chance level of 28%, on melodic features alone (SD-M and Int-LMR). Of these, the intervallic pattern of the melody, represented by Int-LMR, is the most essential. Across all trials, the algorithm consistently relied upon this feature considerably more than any of the others. This means that decision making in Schenkerian analysis relies heavily upon interval patterns: preference for stepwise motion,

and particularly showing directed stepwise progressions (linear progressions).

It might seem surprising that SD-M, while also very important, takes a distant second to Int-LMR. It is important to recognize though, that because this feature applies to the middle note alone, it is much more useful in combination with Int-LMR than by itself (in which case the algorithm must make decisions with no information about the left and right notes). Furthermore, the importance of the tonal anchoring provided by SD-M may be tempered by the fact that the algorithm is constrained to find an appropriate background (i.e., one that ends on $\hat{1}$ in most cases).

The melodic feature in the B group is SD-LR. For the most part, the scale degree of the left and right notes is predictable from the scale degree of the middle note and the intervals to and from the middle note. The reason that SD-LR still shows a moderate level of significance is that it provides an additional distinction between chromatically altered scale degrees and regular ones. While these are somewhat rare, where they do occur they are clearly an important consideration.

4.2. *Harmonic features*

Harmony also plays a substantial role in the analytical decision-making process. Adding HC-LR improves performance from 52% edge accuracy for the two melodic features alone to 60% accuracy. Part of this difference might be attributable to the fact that features of the middle note (like SD-M) are virtually useless without some LR features. Nonetheless, the algorithm relies significantly upon HC-LR in earlier trials also, as Figure 11 shows.

One question posed by the result is why HC-LR, rather than RN-LR, is identified as the crucial harmonic feature, since the latter is simply a slightly more specific version of the same information. (A high proportion of the harmonies in the corpus are I and V). Actually, we probably should read very little into this distinction; on trials that include both features (all within group A) the difference between removing either of them amounts to an average of 0.2%, and is not consistently in one direction. The high value for HC-LR in Figure 11 is due to the fact that it becomes more important in later trials, after RN-LR is removed. We should similarly hesitate to make much of the difference between RN-M and HC-M, which ultimately tips in the opposite direction.

The difference between left/right and middle harmony features is more significant. Two harmonic features of the middle note, RN-M and CT-M, are in group B. Intuitively, we would expect that the status of a note as chord tone or non-chord tone is of crucial importance as an analytical consideration. However, given the basic melodic features (Int-LMR and SD-M), it may be possible to accurately predict when the middle note is a non-chord tone on the basis of HC-LR. (The inclusion of CT-M in group B suggests that this cannot be predicted with perfect accuracy, however.) Some data described in the next section suggest that this accounts for the result. The HC-LR feature then has the added advantage that it distinguishes triangles where the left and right notes are in the same harmony from those where the harmony changes.

The relatively low significance of RN-M may also be surprising. After all, removing this feature means that the algorithm cannot give preference to certain harmonic successions (i.e., V-V-I over V-ii-I). However, it should be noted that successions of three different harmony types (Tonic, Dominant, and Predominant) will be relatively rare in comparison to the many local triangles that involve some kind of repetition (e.g., I-I-I, I-I-V, I-V-V, etc.). At the same time, distinctions about where harmonic changes should occur (or the left, as in I-I-V, on the right, I-V-V) may overlap with metrical considerations, since harmonic changes tend to coincide with strong metrical positions. Nonetheless,

the placement of RN-M in Group B does suggest that such distinctions are sometimes necessary.

4.3. *Metrical features*

Metrical features are of lesser importance than melodic or harmonic ones, but still play a crucial role, as indicated by the inclusion of Met-LMR in group C. Not surprisingly, the most important information is the metrical status of the middle note relative to the left and right notes, which allows the algorithm to generally prefer strong-weak-strong patterns. Of lesser importance is the metrical status of the left and right notes, which is included in group B. This feature allows the algorithm to distinguish weak-to-strong slurrings from strong-to-weak, and, because it is nominal rather than ordinal data, makes absolute-level distinctions ignored by Met-LMR.

4.4. *Temporal features*

The temporal features, though also lesser in importance than melodic and harmonic features, also play an important role. These features are relatively simple: they are all Boolean, and merely make rough distinctions between different temporal levels. The fact that the algorithm relies upon these features is an indication that Schenkerian analysis is not a purely recursive process—that is, rules apply differently at different levels. The most important of these features (group C) is the one that distinguishes the note-to-note level from everything else (Lev1-LR). Also important, to a lesser degree, is Lev2-LR, which allows the algorithm to set different probabilities for what may happen within a measure, and what may happen across measures.

4.5. *Other features (Group A):*

For many of the features in Group A, performance actually increased as they were removed. This seems counterintuitive, and can probably be essentially understood loosely speaking as statistical noise. Given a relatively small number of analyses in the corpus for a large amount of feature data, idiosyncratic features of individual examples may have an excessive influence on the outcome. We would expect such "noise" to be curtailed as the number of features is reduced. The fact that this group of features has essentially no discernable influence on the performance of the algorithm may largely be due to redundancy. For instance, MetO-LR is redundant with MetN-LR, and IntI-LR can be inferred from Int-LMR. CT-LR is mostly redundant with a combination of SD-LR and HC-LR or RN-LR. It also may be of little use because triangles with non-chord tones on the left or right are rare in the corpus. Similar observations may be made about all the interval features in Group A. The other feature in Group A is Lev3-LR. This feature is less useful because, given the length of the examples, relatively few triangles in the MOP will span more than two measures. And those that do will tend to be above the level of the given fundamental structure.

4.6. *Independent usefulness versus expendability*

While the main results equate the importance of features with their expendability, one might also understand importance more along the lines of "independent usefulness." In other words, rather than evaluating how much the algorithm relies upon a feature in

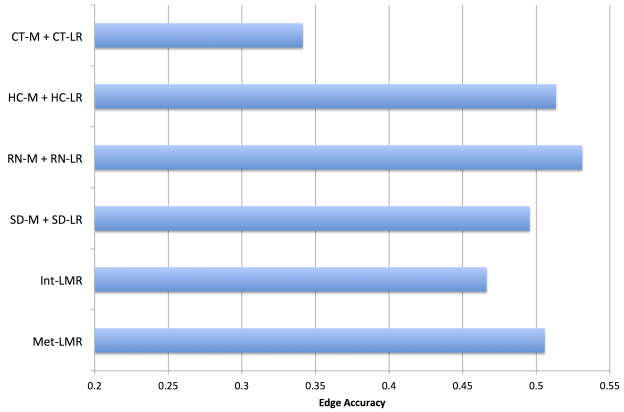


Figure 12. The edge accuracy of four models containing only a single LMR feature, or a single M and LR feature.

the presence of other features (expendability), we might consider how well the algorithm performs using each feature in the absence of the others. As a supplement to the expendability results, we also collected some results in which the algorithm was given an extremely limited amount of information about the music, only one or two features at a time. The results are shown in Figure 12. Since M features require LR features, and vice versa, we paired all similar M and LR features, and also included the LMR features by themselves.

These results show that independent importance of features is quite different from their expendability. By itself, harmony is the most useful feature, followed closely by meter, with the melodic information, by far the least expendable in the main experiment, being less useful than both of these. In other words, melodic features, while the most valuable overall, are also more dependent in their usefulness on some baseline harmonic and metrical information (in the form of HC-LR and Met-LMR). SD-M + SD-LR performs somewhat better (though not dramatically so) than Int-LMR, which means that tonal orientation does help—beyond what orientation is provided by the given *Urlinie*—even in the absence of harmonic information. The algorithm also does slightly better with Roman numerals than harmonic classes, which provides some evidence that distinctions between IV and II and between V and VII do in fact provide some small benefit. The one feature that proves virtually useless by itself (with performance close to chance) is the chord tone/non-chord tone distinction. We can speculate that this is largely do to the uselessness of CT-LR: there are few instances of *any* triangles in the corpus with non-chord tones on the left or right, so probabilities conditional upon these values of CT-LR are not very meaningful.

5. Exploratory data

The results in the previous section give a general picture of what kind of musical information is most important in producing a Schenkerian analysis. However, we also saw that the contribution of each feature to the analysis is highly dependent upon what other features are present. To get a sense of how the different features were interacting, we tracked the effect of each feature over the course of the experiment, by considering how much the removal of that feature affected performance at each stage. For instance, Figure 13 shows how the effect of removing the scale degree feature changes from trial to trial.

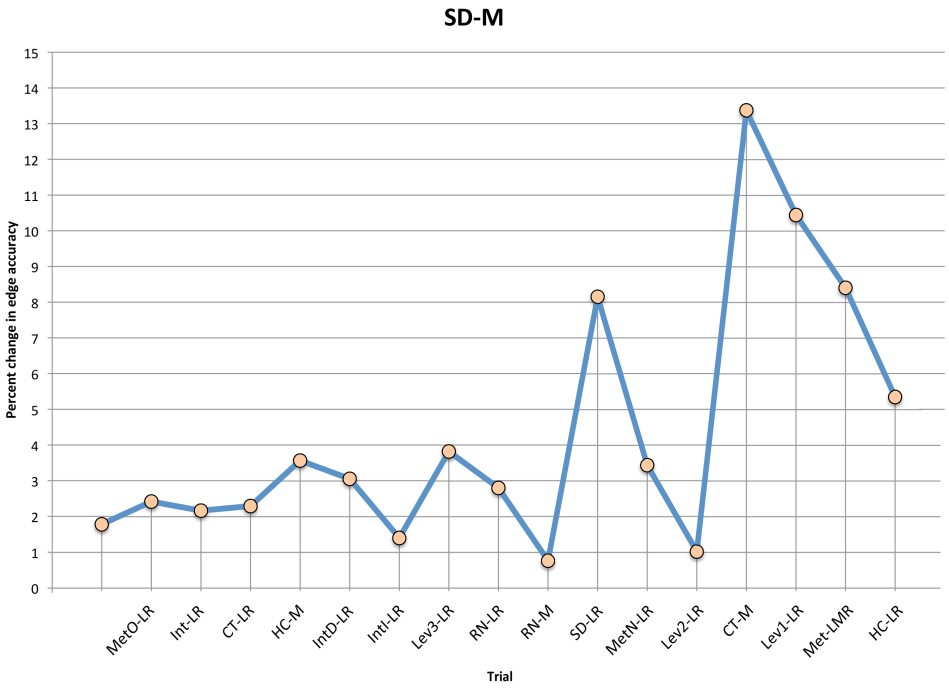


Figure 13. The effect of removing the SD-M feature on trials 0–16. The labels on the X-axis show the feature removed prior to the given trial. For instance, the peak at the trial for SD-LR means that the effect of removing SD-M changed to about 8% *after* the removal of SD-LR.

Were the relationship of this to other features equivocal, we might expect the values to stay roughly the same or to steadily increase from one trial to the next. The chart for SD-M is much more jagged, suggesting that it has non-trivial relationships with other features. For instance, when SD-LR is removed, SD-M suddenly spikes in importance. This suggests that the preceding low value probably reflected, not a lack of significance for scale degree information in general, but that there is considerable overlap between the two scale degree features, making one of them—but not both—expendable. This makes sense, because given that the algorithm has intervallic information about the melody (in the form of Int-LMR), either scale degree feature can anchor that intervallic information to a tonal center.

The most potentially meaningful data in these graphs are the places like this where there are large changes in the importance of some feature from one trial to the next. We can infer that these large changes result from some kind of interaction between the feature in question and the one removed from one trial to the next. Broadly speaking there are two possible types of interaction that such changes may reflect: First, some groups of features might provide redundant information, meaning that the removal of both has a much greater effect than the removal of one or the other alone (as in the case of the two scale degree features). There are some obvious cases of redundant information in the list of features that we included in the experiment. For instance, the MetO-LR feature adds no additional information that cannot be inferred from MetN-LR. Therefore after MetO-LR is removed, we should find that the algorithm is more sensitive to the removal of MetN-LR, which it is: removing it has no effect on edge accuracy in trial 1, but decreases edge accuracy from 64.1% to 62.7% on trial 2. In some cases, redundancies may involve larger groups of more than two features. For example, if one knows the scale

Table 1. Changes of feature importance from one trial to the next within one standard deviation of the mean (between −2.23% and +2.47%).

Feature	Trial	Feature removed	Edge accuracy	Difference from full	Change from previous trial
SD-M	13	CT-M	52.61	13.38	12.36
HC-LR	13	CT-M	52.74	13.25	7.77
SD-M	10	SD-LR	60.00	8.15	7.39
Met-LMR	14	Lev1-LR	60.00	0.51	−5.99
CT-M	10	SD-LR	67.52	0.64	−5.86
HC-LR	14	Lev1-LR	52.99	7.52	−5.73
Int-LMR	10	SD-LR	59.36	8.79	5.73
Int-LMR	15	Met-LMR	45.73	14.27	5.22
MetN-LR	10	SD-LR	68.28	−0.13	−4.71
SD-M	11	MetN-LR	64.84	3.44	−4.71
Int-LMR	13	CT-M	53.50	12.48	4.59
Int-LMR	12	Lev2-LR	58.85	7.90	−4.08
Met-LMR	13	CT-M	59.49	6.50	4.20
Int-LMR	8	RN-LR	66.62	3.95	−3.69
Met-LMR	11	MetN-LR	64.20	4.08	3.82
Int-LMR	14	Lev1-LR	51.46	9.04	−3.44
Int-LMR	5	IntD-LR	65.35	7.64	3.44
SD-M	16	HC-LR	46.62	5.35	−3.06
Met-LMR	10	SD-LR	67.90	0.25	−3.06
Int-LMR	11	MetN-LR	56.31	11.97	3.18
SD-M	14	Lev1-LR	50.06	10.45	−2.93
RN-LR	4	HC-M	72.36	0.76	−2.68
SD-LR	9	RN-M	68.15	0.51	−2.68
RN-LR	3	CT-LR	69.43	3.44	2.68
HC-LR	10	SD-LR	63.31	4.84	2.68
RN-M	8	RN-LR	68.66	1.91	−2.42
SD-M	12	Lev2-LR	65.73	1.02	−2.42

degree of the middle note (SD-M) then one can infer the scale degree of the left and right notes (SD-LR) by knowing the intervals between the middle and the left and right notes (Int-LMR).

The second type of interaction is the cooperation of two features. Sometimes, the information provided by one feature may not be useful without some information provided by another. In this type of situation, we should find that the importance of one feature *drops* significantly when the other feature is removed. Such interactions are harder to predict in advance, but the experiment generated some particularly interesting examples. For instance, in Figure 13 we see a drop in importance of SD-M upon the removal of MetN-LR and Lev2-LR. This suggests that the algorithm needs information about metric context to make use of SD-M.

To search the data systematically for the most prominent such interactions between parameters, we tallied every change of feature importance from one trial to the next and ordered them from largest to smallest in distance from the mean (where "feature importance" is the reduction of triangle accuracy that results from removing the given feature on the given trial). The mean change is 0.12%, indicating a tendency for feature importance to increase modestly on average as the model gets smaller. Table 1 includes all cases where the change is within one standard deviation (2.35%) of this mean.

5.1. *Removal of CT-M*

The largest two changes, and four of the largest 20, occurred between trials 12 and 13 after the removal of the CT-M feature. All of these are increases in sensitivity, which indicates redundancies. These redundancies apparently involve a large group of parameters. One would expect the CT-M feature to be very significant for assessing local melodic structure.

However, one can usually predict whether or not the middle tone is a chord tone given the harmonic class of the left and right notes (which will be the same if all three events happen within a single harmony) by knowing the scale degree of the middle note. The large change in sensitivity for these two features on trial 14 suggests that they are compensating for CT-M in this way after it is removed.

The other two features that increase in importance on trial 14 are Int-LMR and Met-LMR. These may compensate for the loss of CT-M by identifying intervallic and metric patterns characteristic of non-chord tone figures.

These data help us interpret the ordering of features arrived at in the experiment. One might be surprised that the CT-M feature does not persist somewhat longer, or that there is not a larger drop in accuracy on trial 14 after it is removed. However, a logical explanation is that the algorithm is able to compensate by essentially predicting whether a note is a non-chord tone on the basis of other features. The most important of these features, SD-M and HC-LR, are amongst the three features to last through trial 15, and the four features that are collectively associated with CT-M are precisely the last four features to be eliminated in the process.

5.2. *Removal of SD-LR*

Trial 10, after the removal of SD-LR, is the most disruptive overall shift in feature importance. The shift can be divided into two features with large increases in importance, and three that decrease. The increases are on SD-M and Int-LMR. The reason for these increases is obvious and already mentioned above: the value of SD-LR can be predicted from the combination of SD-M and Int-LMR. The three large decreases are on CT-M, MetN-LR, and Met-LMR.

The latter result can only be understood by considering what information SD-LR adds that cannot be compensated for by a combination of Int-LMR and SD-M. Because the intervals of the Int-LMR feature are generic, removing SD-LR means that there is no distinction between diatonic and chromatically altered notes on the left or right. Although chromatically altered notes are relatively rare in the corpus, there are at least five examples that include accented chromatic dissonance. The chromatic status of these note clearly overrides their metric position, making them unsuitable as left/right notes. Without this distinction, the predictive value of metric features is diluted. More generally, the distinction between chromatic and diatonic non-chord-tones is lost.

5.3. *Removal of Lev1-LR*

The next most prominent trial in Table 1 is trial 14, where the Lev1-LR feature is removed. All of the large changes on this trial are decreases in importance, meaning what we primarily see with Lev1-LR is that it is most useful in combination with other features. These are, in order, Met-LMR, HC-LR, Int-LMR, and SD-M—i.e., all the features remaining in the model at the point that Lev1-LR is removed. This result suggests that all of these features, which cover all the basic musical parameters that we studied, operate differently at the local, note-to-note, level than at deeper levels. In other words, this is strong evidence that the rules of note-to-note analysis differ in systematic ways from higher-level analytical reasoning.

5.4. *Removal of Met-LMR and Harmony Class L/R*

Not surprisingly, the last few trials lead to large adjustments in feature importance. (Note that trial 17 is excluded because there is only one feature left afterwards.) The removal of the last metric feature leads to a large positive adjustment in Int-LMR and a smaller negative one in SD-M. This probably simply reflects the fact that Int-LMR is the only other feature that provides information about the middle note relative to L/R context, and that information specific to the middle note (SD-M) becomes useless with no L/R context. The same reasoning may explain the negative shift in SD-M on trial 16, and in fact the overall negative trajectory of SD-M's importance over the last four trials. (See Figure 13.)

5.5. *Volatility of Int-LMR*

Many of the data points in Table 1 involve changes in the Int-LMR feature. There are some obvious reasons for this: this feature persists through all trials, including the later ones where larger shifts are to be expected. It also has the highest levels of feature importance in earlier trials, which also may partly explain why has a wider range of variability.

It is therefore convenient to consider the trajectory of Int-LMR's feature importance over the entire course of the experiment, as shown in Figure 14. The general upward trajectory is interrupted by three prominent dips. The first corresponds to the removal of two harmonic features, suggesting that the likelihood of certain interval patterns depends upon harmonic context. For instance, an arpeggiation might be a likely pattern when there is no change of Roman numeral, but should be avoided if there is a change. The other two dips occur with the temporal features, Lev2-LR and Lev1-LR, meaning that the likelihood of certain intervallic patterns changes between the very local levels (note-to-note and within-measure) and deeper levels.

5.6. *Other Group B trials (removal of RN-M and Lev2-LR)*

We have already noted large readjustments on trials 10 and 13 (removal of CT-M and SD-LR)). The other three Group B trials also appear on Table 10, with trial 11, removal of MetN-LR, being the most prominent. The increased reliance on Met-LMR after the removal of MetN-LR is likely due to the obvious redundancy between these two metric features, and the increase for Int-LMR is part of a general trend towards increasing reliance on that feature in later trials (see Fig. 14). The drop in importance for SD-M is more interesting: it suggests that the ability to locate melodic patterns relative to the key or prevailing harmony is most useful when we also have some absolute metric information (to supplement the relative metric information of Met-LMR), and are able to compare the metric status of the left and right vertices directly.

Trial 12, where Lev2-LR is eliminated, appears on Table 1 twice because the two melodic features, Int-LMR and SD-M, both drop in importance on this trial. This means that different kinds of melodic patterns are likely across barlines than within a single measure, further evidence that melodic patterns especially are level-dependent. Finally, Trial 9, removal of RN-M, appears once on the table. In fact, there are many other shifts in feature importance on this trial that are sizable although they do not quite make it onto the table. In addition to SD-LR, three other features, Lev1-LR, HC-LR, and SD-M, all drop in importance by over 2%. This may indicate that certain combinations of scale degree progressions and harmonic progression are typical beyond the note-to-note level

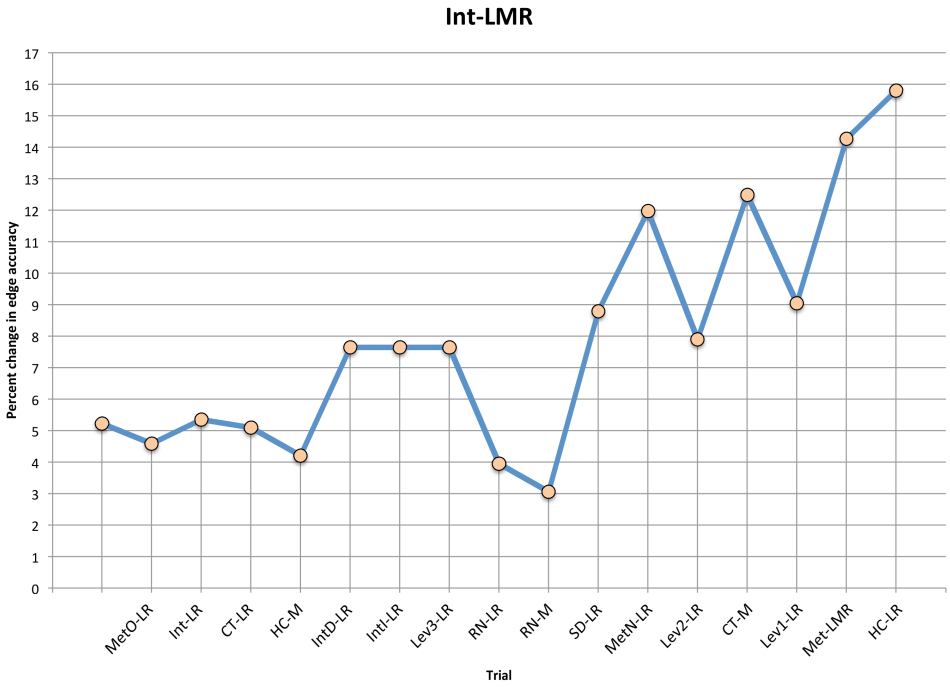


Figure 14. Changes in Int-LMR's feature importance through the trials.

of structure. The only increase in feature importance over 2% on this trial is for CT-M (2.29%), which is the only remaining harmonic property of the middle note.

5.7. *Group A trials*

It is unsurprising for the most part that there are few major readjustments in the early trials. Two positive shifts that do appear on the list in Table 1 reflect obvious redundancies: Int-LMR compensates for the removal of IntD-LR, and RN-LR (in combination with SD-LR) may predict the value of CT-LR. There are also three large drops in importance on Group A trials. One of these (Int-LMR on trial 8) is discussed above. Also on trial 8, RN-M drops in importance after the removal of RN-LR, and in fact drops from the model altogether on the next trial. Since HC-LR is still in the model at this point (but not HC-M), this may indicate that certain predictive Roman numeral progressions may be lost in the classification.

Finally, we found a drop in the value of RN-LR after the removal of HC-M on trial 4, indicating that the main condition on the harmony of the middle note is the harmony of the left and right notes. The size of this shift is surprising given that RN-M remains in the model at this point, and therefore little real information is lost by the removal of HC-M.

6. Conclusions

- Melodic, harmonic, and metrical features are all significant considerations in Schenkerian analysis.

- Overall, melodic intervals and melodic orientation to a key are the most essential factors for decision making in Schenkerian analysis.
- However, harmonic and metrical information is also necessary, and these are most useful in the absence of other features. I.e., analytical decisions based on harmony of meter tend to be more independent of melody than vice versa.
- The rules of Schenkerian analysis vary from level to level. In particular, there is strong evidence that the note-to-note level follows substantially different rules than larger-scale levels. This is especially true of how melodic patterns work in Schenkerian analysis, which varies not only at the note-to-note level, but also between within-measure and across-measure levels.
- The orientation of melodic patterns in a key is significant, but its implications vary with metrical context and analytical level.

Disclosure statement

The author has no conflict of interest.

References

Agawu, Kofi. 2009. *Music as Discourse: Semiotic Adventures in Romantic Music.* New York: Oxford University Press.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Burstein, L. Poundie. 2011. "Schenkerian Analysis and Occam's Razor." *Res Musica* 3: 112–121.

Cadwallader, Allen, and David Gagné. 1998. *Analysis of Tonal Music: A Schenkerian Approach.* Oxford: Oxford University Press.

Forte, Allen, and Steven E. Gilbert. 1982a. *Instructor's Manual for* Introduction to Schenkerian Analysis. New York: W. W. Norton and Company.

Forte, Allen, and Steven E. Gilbert. 1982b. *Introduction to Schenkerian Analysis.* New York: W. W. Norton and Company.

Jiménez, Víctor M., and Andrés Marzal. 2000. "Computation of the N Best Parse Trees for Weighted and Stochastic Context-Free Grammars." In *Advances in Pattern Recognition,* Vol. 1876 of *Lecture Notes in Computer Science* edited by Francesc J. Ferri, José M. Iñesta, Adnan Amin, and Pavel Pudil, 183–192. Springer-Verlag.

Jurafsky, Daniel, and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics.* 2nd ed. Prentice-Hall.

Kirlin, Phillip 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, Phillip B. 2014b. "A Probabilistic Model of Hierarchical Music Analysis." Ph.D. thesis, University of Massachusetts Amherst.

Kirlin, Phillip B., and David D. Jensen. 2011. "Probabilistic Modeling of Hierarchical Music Analysis." In *Proceedings of the 12th International Society for Music Information Retrieval Conference,* 393–398.

Kirlin, Phillip B., and David D. Jensen. 2015. "Using Supervised Learning to Uncover Deep Musical Structure." In *Proceedings of the 29th AAAI Conference on Artificial Intelligence,* 1770–1776.

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

Provost, Foster, and Pedro Domingos. 2003. "Tree Induction for Probability-Based Ranking." *Machine Learning* 52 (3): 199–215.

Rothstein, William. 1990. "The Americanization of Schenker Pedagogy?." *Journal of Music Theory Pedagogy* 4 (2): 295–300.

Schachter, Carl. 1990. "Either/Or." In *Schenker Studies,* Vol. 1 edited by Hedi Stiegel, 165–179. Cambridge: Cambridge University Press.

Yust, Jason. 2006. "Formal Models of Prolongation." Ph.D. thesis, University of Washington.

Yust, Jason. 2009. "The Geometry of Melodic, Harmonic, and Metrical Hierarchy." In *Proceedings of the International Conference on Mathematics and Computation in Music,* .

Yust, Jason. 2015. "Voice-Leading Transformation and Generative Theories of Tonal Structure." *Music Theory Online* 21 (4).