VOICE-LEADING SCHEMA RECOGNITION USING RHYTHM AND PITCH FEATURES

Christoph Finkensiep[1] **Ken Déguernel**[1] **Markus Neuwirth**[2,1] **Martin Rohrmeier**[1]

[1] École Polytechnique Fédérale de Lausanne [2] Anton Bruckner University Linz

christoph.finkensiep@epfl.ch, ken.deguernel@epfl.ch, markus.neuwirth@epfl.ch, martin.rohrmeier@epfl.ch

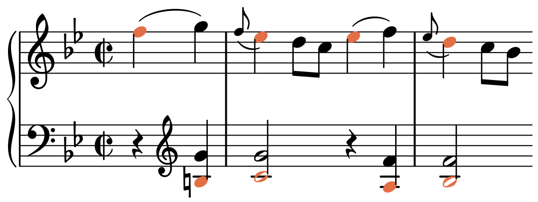
ABSTRACT

Musical schemata constitute important structural building blocks used across historical styles and periods. They consist of two or more melodic lines that are combined to form specific successions of intervals. This paper tackles the problem of recognizing voice-leading schemata in polyphonic music. Since schema types and subtypes can be realized in a wide variety of ways on the musical surface, finding schemata in an automated fashion is a challenging task. To perform schema inference we employ a skipgram model that computes schema candidates, which are then classified using a binary classifier on musical features related to pitch and rhythm. This model is evaluated on a novel dataset of schema annotations in Mozart's piano sonatas produced by expert annotators, which is published alongside this paper. The features are chosen to encode music-theoretically predicted properties of schema instances. We assess the relevance of each feature for the classification task, thus contributing to the theoretical understanding of complex musical objects.

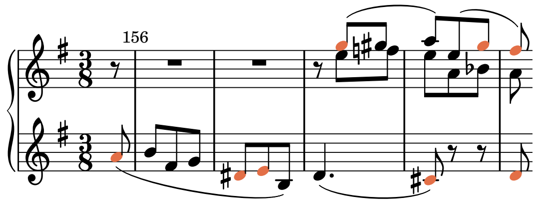
1. INTRODUCTION

Voice-leading schemata are frequently used patterns that can be found across historical periods, ranging from Renaissance, Baroque, and Classical to modern tonal music; examples include such well-known schemata as the Lamento, the Pachelbel, the descending-fifths sequence, and cadences [8, 10, 4, 11]. A schema serves as a template for contrapuntal structure that can be elaborated in multiple ways.

At present, there is only scant quantitative evidence about the frequency and diachronic distribution of schemata across history (e.g., [10, 2]); large-scale, machine-readable datasets on schemata are not yet available. For assessing the prevalence of schemata in a corpus of music, automated recognition of schema instances



(a) A (true) Fonte at the beginning of K281-iii.



(b) A possible candidate for a Fonte in K283-iii.

Figure 1: An example of a Fonte (a) with structural notes highlighted. The task is to decide whether proposed instances such as (b) are true instances or not.

can be a time- and cost-efficient alternative to manually labelled data. However, there are two key challenges for computational approaches when seeking to uncover note patterns in music: (1) the multidimensional (polyphonic) structure of music as opposed to, for example, the sequential structure of written text [17]; (2) the highly flexible nature of these patterns, given that the structural notes in the individual voices can be elaborated in a wide variety of ways.

Voice-leading schemata can be defined as configurations of two or more voices that move together through a sequence of stages, forming particular patterns of successive vertical intervals that occur within a specific tonal context. Consider the example of the Fonte (e.g., [10]): The Fonte is a four-stage pattern involving at least two voices. The bass moves through the scale degrees $\sharp\hat{1} \to \hat{2} \to \hat{7} \to \hat{1}$ of a major scale, while the soprano follows the pattern $\hat{5} \to \hat{4} \to \hat{4} \to \hat{3}$, thus producing the following sequence of vertical intervals: tritone $\to$ minor third $\to$ tritone $\to$ major third. The schema prototype can be elaborated in actual compositions in many different ways. For instance, the notes belonging to one stage can be displaced in time; any number of elaboration notes can be inserted between the structural notes of one stage and between stages... An example illustrating the surface realization of a Fonte is shown in Figure 1a. While containing the correct interval pattern is a central property of any schema instance, it is not sufficient: the selected notes must also provide the contrapuntal template for its context, so that the notes contained in the time-span covered by the schema candidate can be meaningfully interpreted as ornamentations of the selected notes. Figure 1b shows a candidate for a Fonte instance. The task at hand is to decide whether or not such a candidate is a true schema instance.

To tackle the problem of schema detection, this paper provides two contributions. First, we propose a novel dataset with hand-annotated schemata found in Mozart's piano sonatas (Section 3). Second, we present a binary classifier that recognizes true schema instances among a set of proposed schema candidates based on rhythm and pitch features related to regularity, complexity, salience, and harmonic context (Section 4). We evaluate the impact of these features on the classification task using a logistic regression (Section 5).

2. RELATED WORK

Automated discovery and recognition of musical patterns is a topic of ongoing interest in the MIR community [15, 5, 17, 3, 9, 12, 16]. Voice-leading schemata as a specific class of patterns have so far received only little attention; they have been approached with computational methods only very recently. For instance, Symons [23] has developed an algorithm that recognizes schemata in a small corpus, pointing out the importance of rhythmic regularity. Finkensiep et al. [7] tackle the problem of temporal displacement and free polyphonic textures using a two-dimensional extension of skipgrams, which have previously been proposed by Sears et al. [21, 22]. Recently, Katsiavalos et al. [13] have presented a system that uses heuristics-based time-span reduction to discover and recognize schemata.

Several studies aimed at finding cadences, which can be viewed as a subcategory of voice-leading schemata, and evaluated the features relevant for the classification task. Bigo et al. [1] evaluated a set of 44 features linked with the moment of cadential arrival, which are integrated using a support-vector machine for classifying beats as belonging to a cadence or not. Sears et al. [21] use skipgrams on vertical slices to find cadences using a figured bass-like representation of the notes in each slice. Duane [6] approaches cadences directly as voice-leading patterns by trying to recognize and learn them from melodic motion.

3. DATASET

Our dataset is based on the full set of Mozart's piano sonatas encoded in MusicXML format. These 18 sonatas with 3 movements each (thus 54 movements in total) have been composed between 1774 and 1789 and constitute a prominent sample of the classical style. The pieces in the dataset contain 103,829 notes in total distributed over 7,500 measures, with 244 hand-annotated true instances (0.13%) and 190,994 automatically generated false instances (99.87%) for the selected schema types and subtypes (see Table 1).

Schema	Variant	Occurrences
Do-Re-Mi	`.2`	5
Fenaroli	`.2(.min)`	10 (3)
	`.2.flipped(.min)`	43 (8)
	`.2.melcanon(.min)`	6 (2)
	`.2.basscanon.min`	1
Fonte	`.2`	49
	`.2.flipped`	2
	`.2.majmaj`	8
Indugio	`.2`	9
	`.2.voiceex`	5
Lamento	`.2`	2
Lully	`.2`	2
Morte	`.2`	1
Prinner	`.2`	32
Quiescenza	`.2`	46
	`.2.diatonic`	6
Sol-Fa-Mi	`.2`	4

Table 1: List of schemata with their variants and number of occurrences in the Mozart's Piano Sonata dataset.

3.1 Schema Formalization and Lexicon

For the present study, we selected 10 schema types and 20 subtypes (listed in Table 1) which have been suggested in the literature [10, 4, 20, 19]. The approach presented here assumes that a schema type consists of (1) a fixed number of voices; (2) a fixed number of stages, whereby each stage contains one note per voice; and (3) a characteristic interval pattern between these notes. The prototype for each schema variant (or subtype) is specified using a formal notation. For instance, the prototype of the two-voice Fonte is encoded as:

```
"fonte.2": [["a1", "P5"],
            ["M2", "P4"],
            ["M7", "P4"],
            ["P1", "M3"]]
```

where ".2" indicates the two-voice variant of the Fonte. Each note is given as an interval to some arbitrary reference point. Since all possible transpositions of the schema are considered by the matcher, it is not necessary to know the reference key.

Schema instances are encoded as nested arrays of notes in the same form as the corresponding prototypes. Instances may deviate from the shape of the prototype if (a) a note that would repeat its predecessor (e.g. the second $\hat{4}$ in the Fonte) is held over or missing, or (b) two adjacent voices merge and are represented by a single note on the surface.

3.2 Data Production

The dataset consists of two parts: manual annotations by experts and automatically retrieved candidates for schema instances, i.e. sets of notes with an interval pattern conforming to a schema variant. Both the annotations and the computed candidates share the same encoding format, namely a nested lists of note IDs (one sublist per stage, one note per voice) that corresponds to note elements in a MusicXML representation of the scores. While the manual annotations provide the true instances of the dataset, the false instances consist of all skipgram candidates that do not appear in the annotations. [1] The complete dataset is available on github. [2]

3.2.1 Expert Annotations

Two annotators (the third author and Adrian Nagel) provided their analyses by using a web-based annotation app that was specifically developed for the annotation process. The app displays a score using the Verovio toolkit [18], and allows the user to select individual notes from the musical score to mark schema instances. Instances are automatically checked for conformance with the schema prototype in the lexicon, while permitting the deviations described in Section 3.1. The annotators also considered additional criteria such as harmonic signature, phrase structure, pattern repetition, and form-functional context. [3]

3.2.2 Computing Candidates with Skipgrams

In order to compute all candidates of schemata for the classifier, we base our work on the generalized skipgram model proposed in [7], which enumerates two-dimensional note configurations that occur within certain temporal bounds. We use this algorithm to find configurations with a maximal note displacement of 1 bar per stage and a maximal distance of 1 bar between the onsets of two adjacent stages. The configurations are filtered for a specific interval pattern during enumeration regardless of the local keys. This method provides us with all candidates for a schema instance within a reasonable window. However, due to the exhaustive search and a high number of possible note combinations, our resulting dataset is extremely unbalanced. Because of the high combinatoric complexity, we restrict this study to two-voiced schema variants. Furthermore, we reduced the number of candidates to at most 25 per group of temporally overlapping candidates using a previous version of the model presented here.

4. FEATURES AND SCHEMA CLASSIFICATION

4.1 Musical Features

By using precomputed schema candidates that are known to have the correct interval structure (which is all information that we consider for a specific schema type), the problem is narrowed down to deciding whether or not the candidate consists of the structurally important notes. To this end, we have defined a set of features with regard to rhythmic, pitch, and metric information, inspired in part by previous work [10] and that we wish to evaluate with the classifier. These features attempt to measure the *recognizability* of the candidate as a structural pattern, assessing, for example, its complexity, salience, or regularity in various musical dimensions. For a schema candidate C that consists of a number of stages n_s and a number of voices n_v, let $C_{s,v}$ denote the note from stage s and voice v. Each note is represented by an onset, an offset, and a pitch. Whenever pairs of notes are compared, K denotes the numbers of compared note pairs (excluding pairs with missing notes).

The first feature can be considered a rough estimate of the *harmonic or modal signature* of the schema candidate. We define the *harmonic profile* of a candidate as the distribution of pitch-classes (relative to the transposition of the match) of notes that overlap with the time span of the candidate (excluding the matched notes themselves). The `profiledist` is defined as the euclidean distance between a match's pitch profile and the average pitch profile of all true instances of the same schema. Thus, this feature uses training data to derive the prototype profiles instead of defining a harmonic signature a priori.

Three features address the *regularity* of pitch and rhythm between pairs of stages. `rreg` measures the average rhythmic dissimilarity between each pair of stages. For a pair of stages, the rhythmic dissimilarity is defined as the sum of the temporal distance of the notes of the same voice, given the best alignment possible when projecting one stage unto the other. `mreg` is defined very similarly, but here the alignment offset is fixed to whole beats to preserve metric position. Finally, `preg` measures the average pitch dissimilarity between each pair of stages. Similar to rhythmic dissimilarity of a pair of stages, pitch dissimilarity is defined as the sum of the pitch distances of the notes of the same voice, given the best pitch alignment possible when projecting one stage unto the other. These features are defined as

$$\star\texttt{reg} = \frac{1}{K} \sum_{\substack{(s,s')\in stage\\ s\neq s'}} \min_{\delta} \left(\sum_{v=1}^{n_v} |\mu(C_{s,v}) - \mu(C_{s',v}) - \delta| \right), \tag{1}$$

where μ corresponds to onset for `rreg` and `mreg`, and to pitch for `preg`. For `mreg`, δ is restricted to integer multiples of a beat.

We then define features corresponding to the *complexity* of the candidates in terms of displacement between pairs of notes. `rdsums` and `pdsums` respectively correspond to the average temporal and pitch distance between each note of the same stage. They are defined as

$$\star\texttt{dsums} = \frac{1}{K} \sum_{s=1}^{n_s} \sum_{\substack{(v,v')\in voice\\ v\neq v'}} |\mu(C_{s,v}) - \mu(C_{s,v'})|, \tag{2}$$

[1] This includes alternative versions of true instances with several possible note selections.

[2] `https://github.com/DCMLab/schema_annotation_data`

[3] As detailed in the schema-annotation guidelines (`https://github.com/DCMLab/schema_annotation_data/blob/master/manual/manual.pdf`).

where μ corresponds to onset for `rdsums` and to pitch for `pdsums`. Similarly, `rdsumv` and `pdsumv` respectively correspond to the average rhythmic and pitch distance between each note of the same voice. They are defined as

$$\star\texttt{dsumv} = \frac{1}{K}\sum_{v=1}^{n_v}\sum_{\substack{(s,s')\in stage\\ s\neq s'}}|\mu(C_{s,v}) - \mu(C_{s',v})|, \quad (3)$$

where μ correspond respectively to onset and pitch for `rdsumv` and `pdsumv`.

Another perspective at pitch displacement is provided by `vdist`, which measures the average amount of octave jumps within a voice from one stage to the next, and is defined as

$$\texttt{vdist} = \frac{1}{K}\sum_{v=1}^{n_v}\sum_{s=1}^{n_s-1}\left\lfloor\frac{\text{pitch}(C_{s+1,v}) - \text{pitch}(C_{s,v})}{\text{octave}}\right\rfloor. \quad (4)$$

While a certain complexity may be necessary to make a regular pattern recognizable in the first place, a more complex pattern can be more difficult to detect in the presence of other notes. For this reason, `onsets` counts the average number of distinct note onsets in the context of each stage. A low number of onsets allows the stages to be rhythmically displaced while still being recognizable as a unit. Given the number of distinct note onsets D_s for each state s, we have

$$\texttt{onsets} = \frac{1}{n_s}\sum_{s=1}^{n_s} D_s. \quad (5)$$

Finally, we define two features representing the *salience* of the candidate. First, we consider `dur`, which corresponds to the sum of all note durations in the candidate,

$$\texttt{dur} = \sum_{s,v} \text{offset}(C_{s,v}) - \text{onset}(C_{s,v}). \quad (6)$$

Then we consider `mweight`, a feature based on metric weight. We define our metric weight function as follows:

$$\text{mw}(C_{s,v}) = \begin{cases} 2 & \text{if onset}(C_{s,v}) \text{ is on a strong beat.} \\ 1 & \text{if onset}(C_{s,v}) \text{ is on a weak beat.} \\ \frac{1}{2^p} & \text{if onset}(C_{s,v}) \text{ is on a subbeat,} \end{cases}$$

where p is the number of prime factors needed to express the subbeat. Given that function, `mweight` corresponds to the average metric weight of each note of the candidate:

$$\texttt{mweight} = \frac{1}{K}\sum_{s=1}^{n_s}\sum_{v=1}^{n_v} \text{mw}(C_{s,v}). \quad (7)$$

4.2 Classification and Evaluation Method

The features described above are used as an input to a logistic regression model, a simple binary classifier model that uses a linear combination of the input features and applies a sigmoid to that score, yielding a value between 0 and 1 that indicates the probability of the input to be a true instance. Since a logistic regression is a special case of a neural network without hidden layers, this approach can be naturally extended to include more layers, allowing for more complex, non-linear feature combinations. However, preliminary experiments have shown that non-linear models (such as simple neural networks and support-vector machines) do not increase model performance and instead lead to overfitting, so we exclude them here.

The input data consists of expert annotations and skip-gram candidates, produced as described in Section 3.2. To get consistent temporal information about the notes, we unfold all repetitions and jumps notated in the scores. Repeated occurrences of notes are disambiguated by selecting for every schema candidate those note occurrences that have a consistent temporal order and cover a minimal time span. Finally, matches that have incomplete stages (due to implicit notes, as described above) are converted into complete instances with missing notes marked explicitly.

To evaluate the model's performance, we use 5-fold cross validation. [4] The pitch histograms used for `profiledist` are computed on the respective training set of each run. In order to get an unbiased model, we follow the advice given in [14] and balance our dataset by upsampling the true instances to match the number of false instances. The model is trained on the balanced training data using the Julia package `GLM.jl` [5] and applied to both balanced and unbalanced test data. In addition, a prior-corrected version of the model (see [14]) is applied to the unbalanced data.

The code for the whole evaluation pipeline is provided online [6], including a notebook [7] that was used to generate all results and figures in this paper.

5. RESULTS AND DISCUSSION

5.1 Classification Performance

The overall performance of the model is shown in Table 2, aggregating over the predictions on all test sets. When applied to data balanced by upsampling, the model achieves a high classification performance with an F-score of 0.894 and a Matthews correlation coefficient (MCC) of 0.787. Since the model is trained on balanced data, applying it to unbalanced data simply scales the number of true positives and false negatives, resulting in a drastically reduced precision. Using the prior correction of the unbalanced dataset results in a very high accuracy; however, it introduces a bias to label matches as non-instances, which results in the false negatives dominating the false positives.

Figure 2 shows how the predicted probability of being a true instance is distributed for instances and non-instances (upper-left corner). Non-instances overwhelm-

[4] A 5-fold split was chosen to balance the number of folds and size of the resulting test set.

[5] `https://github.com/JuliaStats/GLM.jl`

[6] `https://github.com/DCMLab/schemata_code/tree/ismir2020`

[7] `https://github.com/DCMLab/schemata_code/blob/ismir2020/notebooks/ismir2020_classification.ipynb`

Condition	TP	TN	FP	FN	accuracy	precision	recall	F-score	MCC
Balanced	171,400	169,867	21,107	19,574	0.893	0.890	0.898	0.894	0.787
Unbalanced	219	169,867	21,107	25	0.889	0.010	0.898	0.020	0.089
Unbalanced (corrected)	15	190,923	51	229	0.999	0.227	0.061	0.097	0.118
Grouped	220	6,009	2,663	12	0.700	0.076	0.948	0.141	0.218
Grouped (corrected)	43	8,596	76	189	0.970	0.361	0.185	0.245	0.245

Table 2: Performance of the model in different conditions. TP = true positives, TN = true negatives, FP = false positives, FN = false negatives, MCC = Matthews correlation coefficient.

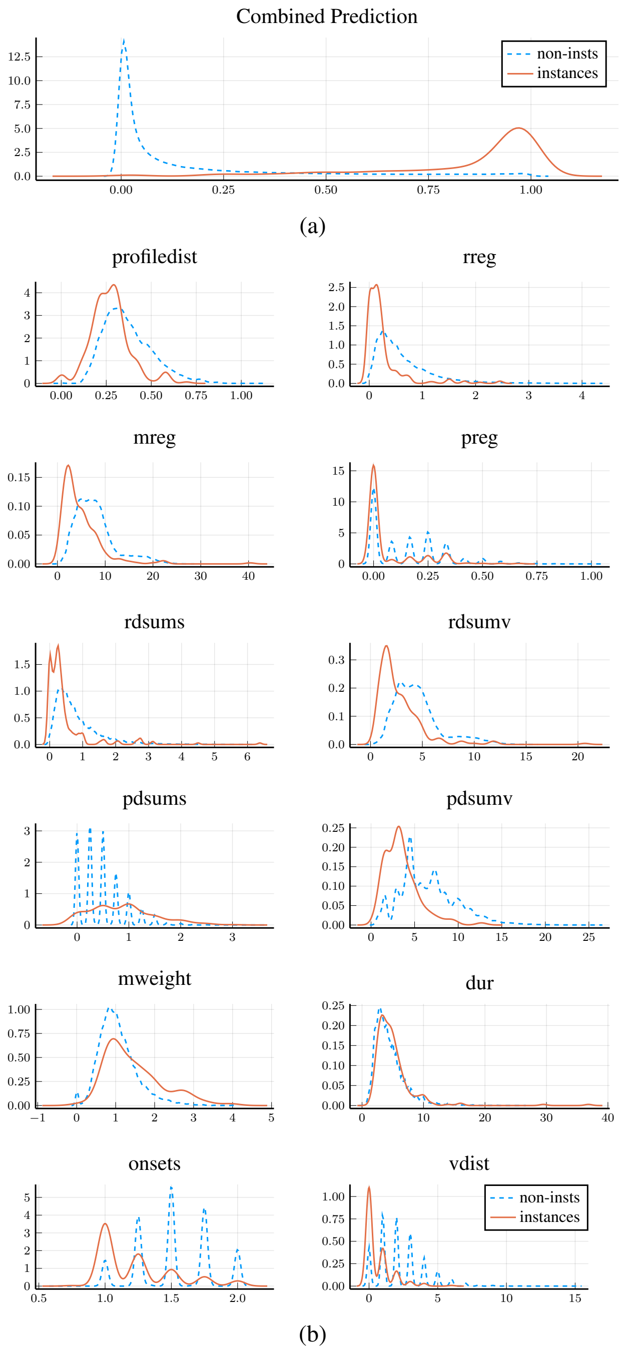


Figure 2: Distribution of model prediction (a) and feature values (b) over instances and non-instances as a kernel density estimate. The more the curves tend in opposite directions, the better the two classes are separated.

ingly receive low probabilities and instances are typically rated very high. This is in line with the model's good performance on raw data, but it also reveals why imbalance poses a serious problem: while the majority of non-instances are correctly discarded by the model, a minority remains indistinguishable from true instances under the model. When the skipgrams propose many more non-instances than instances, the small part of indistinguishable non-instances becomes huge in relation to the true instances. Note that simply reducing the number of matches does not necessarily improve the situation: taking away the matches with a rating < 0.5 still leaves us with the problematic cases.

A lot of non-instances are proposed as combinatoric variations around true instances. To test whether the problematic cases are variations of true instances or genuine non-instances, we group all matches according to temporal overlap (prior correction is based on the imbalance of the groups here). The results (Table 2) show that grouping drastically increases the performance compared to the ungrouped condition but does not get close to the performance on balanced data, indicating that there is still a significant number of indistinguishable true non-instances.

This effect of indistinguishability may be seen as an indicator that our list of features lacks those features that would help resolve the remaining cases and clearly separate the classes. However, it is not clear that finding such features is easily attainable. First, consider that while the existing features are already very informative, the information needed to distinguish the problematic cases would have to be much more precise. Even when the probability of getting a positive result for a non-instance is only 10^{-3}, a true instance proportion of 10^{-3} still leaves a 50% chance that a positive result is a false positive. Second, our annotators conformed to very strict standards in order to discard non-instances, restricting true instances to cases where the schema is a highly plausible template for the musical surface. Such judgments rely on implicit music-theoretical knowledge and intuition, which are difficult to model.

Finally, a look at some highly confident false positives suggests that if schema classification is defined as a binary task (a surface pattern is a schema instance or not), then the performance of this task can hardly been improved.

For example, the excerpt in Figure 1b may not look like a very plausible Fonte at first sight (and was not classified as such by the annotators). However, the last two bars clearly contain the correct contrapuntal pattern for

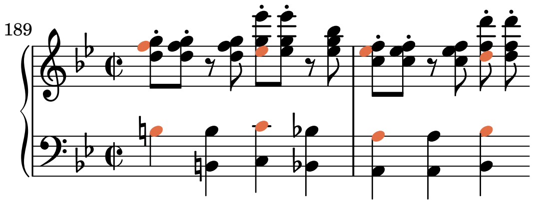


Figure 3: An ambiguous Fonte match (K333-iii). While intrinsically this is a highly plausible instance (interval pattern, tonal context, melodic parallelism), the context discards it, as the pattern is in fact part of a larger descending-fifths sequence.

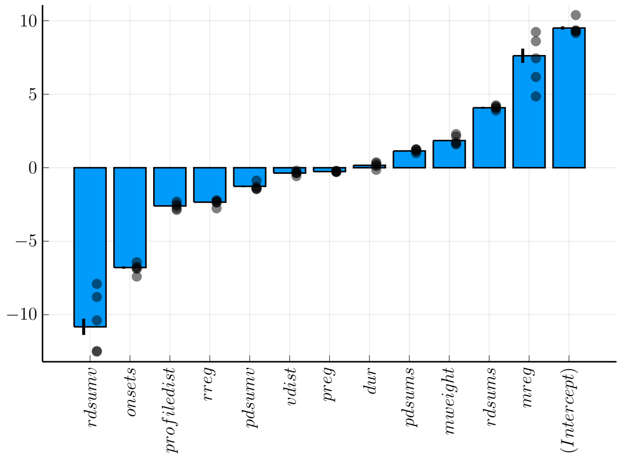


Figure 4: The parameters $\vec{\beta}$ of the model trained on the full upsampled dataset (bars) and normalized by multiplication with the average value of the respective feature. Error bars indicate the 95% confidence interval of the fit. Black points indicate the normalized parameters for each model trained during cross validation.

the stages 3 and 4. The beginning can be interpreted as a melodic unfolding of an Em chord that is ornamented by the notes of a B^7 chord, most clearly in the neighbor note d♯ to e (i.e., the bass for the stages 1 and 2 of a Fonte). Therefore, it can be argued that this section shares its contrapuntal structure with the Fonte, even though the typical parallelism is missing. Another, converse, example can be seen in Figure 3: in isolation, the pattern is a clear instance of a Fonte, but it is continued in the manner of a larger descending-fifths sequence, which, depending on the definition used, may discard it as a Fonte. A negative definition like this is very difficult to check under the current paradigm.

5.2 Feature Evaluation

Figure 4 shows the influence of each feature in a model trained on the full balanced dataset. Overall, schemata seem to expose a high regularity and low complexity compared to non-instance candidates. The strong negative factors `rdsumv` and `onsets` disregard candidates with a large temporal extension and a high degree of non-simultaneity. Metric regularity (i.e. rhythmic regularity aligned to the metrical grid) has a strong positive influence, indicating a preference for a regular temporal organization.

The preference for simultaneity of the notes in the same stage is somewhat contradicted by the moderately positive influence of the `rdsums`, the average note displacement within stages. This is particularly surprising when looking at the distribution of this feature over instances and non-instances (Figure 2b), which shows that instances generally show less displacement than non-instances. One possible explanation of this phenomenon is that the combination of both features (`onsets` and `rdsums`) expresses a general preferences for little displacement, but when the notes are non-simultaneous, then a higher distance is preferred, which may make the structural notes more recognizable.

Less important are features based on pitch (`profiledist`, `pdsum*`, `preg`, and `vdist`) as well as features that indicate basic salience (`dur` and `mweight`). Pitch features are likely of moderate to little importance because most of the relevant pitch-related information is already implied by the schema's interval structure. Interestingly, duration and metric weight (both properties that are taken from each note in isolation) play little to no role, which is confirmed in Figure 2b. This indicates that local properties do not mark notes as structural, this role seems to depend *only* on how the note is used in relation to other notes.

6. CONCLUSION

As the results presented above show, distinguishing between incidental and structural note configurations based on a small number of musically and cognitively motivated heuristics works well in the vast majority of cases. Even if a number of misclassifications remain, a closer look at these cases provides valuable insights into the problem at hand. First, the main limitation of our approach is that the model assesses suggested schema instances individually, without considering, or comparing it to, alternative interpretations. In many cases, the main reason for human experts to reject a candidate does not seem to be a lack of plausibility of the match itself, but rather the availability of a "better explanation", i.e. an alternative analysis of the match's context that identifies a more plausible contrapuntal scaffold. This result is in line with the reduction-based approach of Katsiavalos et al. [13]. Since the features used in this study already proved useful for independent classification, they likely benefit from a general structural-analysis approach, in which schema instances are recognized in reductions of the musical surface.

A second insight concerns the idea of schema itself and its relation to a classification task. From a cognitive perspective, a schema does not need to be instantiated unambiguously or even completely. It is sufficient if listeners recognize the schema as the template for the surface events, or if they understand the composer's intention to evoke the schema. In this regard, *discrete* binary classification into instances and non-instances may be as unattainable as it is undesirable, falling short of the complexity the relationship between schema and realization can exhibit.

7. ACKNOWLEDGEMENTS

The research presented in this paper is generously supported by the Volkswagen Foundation and Claude Latour. We also thank the anonymous reviewers for their helpful feedback.

8. REFERENCES

[1] L. Bigo, L. Feisthauer, M. Giraud, and F. Levé. “Relevance of Musical Features for Cadence Detection”. In: *Proceedings of the International Society for Music Information Retrieval Conference*. 2018.

[2] V. Byros. “Towards an “Archaeology” of hearing: schemata and eighteenth-century consciousness”. In: *Musica Humana* 1.2 (2009), pp. 235–306.

[3] E. Cambouropoulos, T. Crawford, and C. Iliopoulos. “Pattern Processing in Melodic Sequences: Challenges, Caveats and Prospects”. In: *Computers and the Humanities* 35 (2001), pp. 9–21.

[4] W. E. Caplin. “Topics and Formal Functions: The Case of the Lament”. In: *The Oxford Handbook of Topic Theory* (2014). Ed. by D. Mirka, pp. 415–452.

[5] D. Conklin and M. Bergeron. “Discovery of Contrapuntal Patterns”. In: *Proceedings of the International Society on Music Information Retrieval*. 2010, pp. 201–206. ISBN: 978-90-393-5381-3.

[6] B. Duane. “Melodic Patterns and Tonal Cadences: Bayesian Learning of Cadential Categories from Contrapuntal Information”. In: *Journal of New Music Research* 48.3 (2019), pp. 197–216.

[7] C. Finkensiep, M. Neuwirth, and M. Rohrmeier. “Generalized Skipgrams for Pattern Discovery in Polyphonic Streams”. In: *Proceedings of the International Symposium on Music Information Retrieval*. 2018.

[8] A. Forte. *Tonal Harmony in Concept and Practice*. 3rd ed. New York: Holt, Rinehart and Winston, 1979. ISBN: 978-0-03-020756-3.

[9] M. Giraud, K. Déguernel, and E. Cambouropoulos. “Fragmentations with pitch, rhythm and parallelism constraints for variation matching”. In: *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2013, pp. 298–312.

[10] R. Gjerdingen. *Music in the Galant Style*. New York: Oxford University Press, 2007.

[11] S. Jan. “Using galant schemata as evidence for universal Darwinism”. In: *Interdisciplinary Science Reviews* 38.2 (2013), pp. 149–168.

[12] B. Janssen, W. B. de Haas, A. Volk, and P. van Kranenburg. “Finding repeated patterns in music: state of knowledge, challenges, perspectives.” In: *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. 2013, pp. 277–297.

[13] A. Katsiavalos, T. Collins, and B. Battey. “An Initial Computational Model for Musical Schemata Theory”. In: *Proceedings of the International Society on Music Information Retrieval*. 2019, pp. 166–172.

[14] G. King and L. Zeng. “Logistic Regression in Rare Events Data”. In: *Political Analysis* 9.2 (2001).

[15] O. Lartillot. “Automated Motivic Analysis: An Exhaustive Approach Based on Closed and Cyclic Pattern Mining in Multidimensional Parametric Spaces”. In: *Computational Music Analysis*. Ed. by D. Meredith. Springer, 2016, pp. 273–302.

[16] O. Lartillot. “Motivic pattern extraction in symbolic domain”. In: *Intelligent music information systems: Tools and methodologies*. IGI Global, 2008, pp. 236–260.

[17] D. Meredith, K. Lemström, and G. A. Wiggins. “Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music”. In: *Journal of New Music Research* 31.4 (Dec. 1, 2002), pp. 321–345. ISSN: 0929-8215. DOI: 10.1076/jnmr.31.4.321.14162.

[18] L. Pugin, R. Zitellini, and P. Roland. “Verovio: a library for engraving MEI music notation into SVG”. In: *Proceedings of the International Symposium on Music Information Retrieval*. 2014, pp. 107–112.

[19] J. Rice. “Adding to the Galant Schematicon: The Lully”. In: *Forthcoming in Mozart-Jahrbuch* (2014).

[20] J. A. Rice. “The Morte: a Galant Voice-Leading Schema as Emblem of Lament and Compositional Building-Block”. In: *Eighteenth-Century Music* 12.2 (2015), pp. 157–181.

[21] D. R. W. Sears, A. Arzt, H. Frostel, R. Sonnleitner, and G. Widmer. “Modeling Harmony with Skip-Grams”. In: *Proceedings of the International Society on Music Information Retrieval*. 2017, pp. 332–338. ISBN: 978-981-11-5179-8.

[22] D. R. W. Sears and G. Widmer. “Beneath (or beyond) the Surface: Discovering Voice-Leading Patterns with Skip-Grams”. In: *Journal of Mathematics and Music* 0.0 (July 14, 2020), pp. 1–26. ISSN: 1745-9737. DOI: 10.1080/17459737.2020.1785568. URL: https://doi.org/10.1080/17459737.2020.1785568 (visited on 07/29/2020).

[23] J. Symons. “A Cognitively Inspired Method for the Statistical Analysis of Eighteenth-Century Music, as Applied in Two Corpus Studies”. PhD thesis. Northwestern University, 2017.