A Computational Cognitive Model for the Analysis and Generation of Voice Leadings

Peter M. C. Harrison & Marcus T. Pearce
Queen Mary University of London, London, United Kingdom

Voice leading is a common task in Western music composition whose conventions are consistent with fundamental principles of auditory perception. Here we introduce a computational cognitive model of voice leading, intended both for analyzing voice-leading practices within encoded musical corpora and for generating new voice leadings for unseen chord sequences. This model is feature-based, quantifying the desirability of a given voice leading on the basis of different features derived from Huron's (2001) perceptual account of voice leading. We use the model to analyze a corpus of 370 chorale harmonizations by J. S. Bach, and demonstrate the model's application to the voicing of harmonic progressions in different musical genres. The model is implemented in a new R package, "voicer," which we release alongside this paper.

Received: April 1, 2019, accepted September 27, 2019.

Key words: perception, auditory scene analysis, computational models, artificial intelligence, composition

Western music pedagogy traditionally emphasizes two aspects of compositional practice: harmony and voice leading. Harmony specifies a vocabulary of harmonic units, termed "chords," alongside conventions for combining these chords into chord sequences; voice leading describes the art of realizing these chords as collections of individual voices, with a particular emphasis on the progression of individual voices from chord to chord.

Huron (2001, 2016) has argued that Western voice-leading practice is largely driven by the goal of manipulating the listener's psychological processes of auditory scene analysis. Auditory scene analysis describes how the listener organizes information from the acoustic environment into perceptually meaningful elements, typically corresponding to distinct auditory sources that can be related to real-world objects (Bregman, 1990). In Baroque music, voice-leading practice is often consistent with the principle of promoting the perceptual independence of the different musical voices. For example, Baroque composers tended to avoid parallel octaves between independent voice parts, presumably because parallel octaves cause the two voice parts to temporarily "fuse" into one perceptual voice, an incongruous effect when the voices are elsewhere perceived as separate voices (Huron, 2001, 2016). However, perceptual independence is not a universal musical goal: for example, octave doubling has long been accepted in Western music as a technique for creating the percept of a single voice with a reinforced timbre. This technique was taken further by composers such as Debussy, who often constructed entire musical textures from parallel motion while freely disregarding traditional prohibitions against parallel fifths and octaves (e.g., *La Cathédrale Engloutie*, 1910, L. 117/10). In such cases, we might hypothesize that Debussy purposefully adopted parallelism to minimize the perceptual independence of the underlying voices, hence creating a unitary textural stream (Huron, 2016).

Here we seek to develop a computational cognitive model of voice leading. This model is intended to simulate how a composer might choose between various candidate voice leadings on the basis of their consequences for music perception. One goal of constructing such a model is to create a formal basis for testing voice-leading theories on large datasets of music compositions. A second goal is to create a tool for generating voiced versions of unseen chord sequences, with potential applications in music composition and music cognition research.

A computational cognitive model of voice leading could adopt various levels of explanatory depth. For example, a researcher might introduce a model that takes the musical surface as input, simulates the process of auditory scene analysis, and quantifies the extent to which individual voices are recognized as independent auditory streams. If this model successfully predicted composers' decisions, this would support the hypothesis that voice leading is ultimately driven by the goal of maximizing the perceptual independence of musical voices. A second researcher might

Music Perception, volume 37, issue 3, pp. 208–224, issn 0730-7829, electronic issn 1533-8312. DOI: https://doi.org/10.1525/mp.2020.37.3.208

agree that voice-leading practices were originally shaped by perceptual principles, but hypothesize that experienced composers pay little attention to auditory scene analysis in practice, and instead construct their voice leadings from knowledge of voice-leading practice accrued through musical experience. Correspondingly, this second researcher might build a data-driven model that learns to construct voice leadings by emulating voice-leading practice in representative musical corpora, without any reference to auditory scene analysis.

Neither of these approaches is necessarily more "correct" than the other, but both do serve different goals. From a cognitive modeling perspective, the auditory scene analysis model better addresses the ultimate causes of voice-leading practices, explaining how compositional practice may have been shaped by general perceptual principles. In contrast, the data-driven model might better simulate the psychological processes of an individual composer. From a music generation perspective, the auditory scene analysis model is unlikely ever to approximate a particular musical style perfectly, since it neglects cultural contributions to voice-leading practice. In contrast, the data-driven model might effectively approximate a given musical style, but fail to distinguish perceptually grounded principles from culturally grounded principles, and hence fail to generalize usefully to other musical styles.

Here we adopt an approach intermediate to these two extremes. We do not try to build a comprehensive model of auditory scene analysis, and we do not construct a solely data-driven model. Instead, we construct a model that characterizes voice-leading acceptability as an interpretable function of various features that might reasonably be considered by an experienced composer, such as voice-leading distance, parallel octaves, and interference between partials. This level of abstraction is useful for interpretation: it means that we can inspect the model and understand what it has learned about voice-leading practice. This interpretability is also useful for music generation, as it allows the user to manipulate particular aspects of the model to achieve particular musical effects.

Following Huron (2001, 2016), we ground our model's features in both music theory and auditory perception. Music theory tells us about voice-leading rules that composers may have been explicitly taught during their music training, as well as voice-leading rules that analysts have inferred from their study of musical practice. Auditory perception tells us what implications these features may have for the listener, and helps to explain why particular musical styles adopt particular voice-leading practices.

The resulting model is well-suited to both corpus analysis and music generation. Applied to a music corpus, the model provides quantitative estimates of the importance of different voice-leading principles, as well as p values for estimating the statistical reliability of these principles. Applied to novel chord progressions, the model can generate voice leadings based on these different voice-leading principles, with the user having the freedom to use parameters derived from a reference corpus or alternatively to use hand-specified parameters in order to achieve a desired musical effect.

Importantly, the model does not assume a universal ideal for voice-leading practice. According to the model, voice-leading practice in a particular musical style is characterized by a set of regression weights that determine the extent to which composers promote or avoid certain musical features, such as parallel octaves and interference between partials. Depending on the musical style, the contribution of a given feature might reverse entirely; for example, parallel octaves are avoided in Bach chorales, but are commonplace in certain compositions by Debussy. The model's main assumption is that a common set of perceptual features underpin voice leading in diverse musical styles, an assumption that seems plausible in the context of the proposed relationship between voice-leading practice and auditory scene analysis (Huron, 2001, 2016).

In its broader definitions, the art of voice leading includes processes of embellishment and elaboration, whereby an underlying harmonic skeleton is extended through the addition of musical elements such as passing notes, neighbor notes, suspensions, and appoggiaturas (Huron, 2016). These additions can contribute much to the interest of a musical passage. However, they add a whole layer of complexity to the voice-leading task, potentially contributing a new "surface" harmonic progression that should itself obey certain syntactic conventions. It is difficult to model such processes while maintaining a strict division between harmony and voice leading. In this paper, therefore, we omit processes of embellishment and instead formalize voice leading as the task of assigning pitch heights to pitch classes prescribed by a fixed harmonic progression. This process might also be termed "voicing"; we retain the term "voice leading" to emphasize how we are interested not only in the construction of individual chord voicings but also in the way that these voicings lead consecutively from one voicing to the next.

Voice leading is typically taught in the context of musical styles where each note is explicitly assigned to a particular voice part, such as Baroque chorale harmonizations. However, voice leading can also be important

in other styles: for example, effective voice leading is considered essential to jazz music, despite the fact that jazz harmony is often played on the piano or guitar, where explicit voice assignment is lacking (Tymoczko, 2011). We wish for our model to generalize to such styles, and therefore we do not include explicit voice assignment in the algorithm. Instead, the algorithm infers voice assignments solely from the pitch content of the musical passage, and uses these inferred assignments to evaluate voice-leading rules.

There are several published precedents for voice-leading modeling. Models specifically of voice leading are quite rare (see Hörnel, 2004, for one such model), but many models do exist for melody harmonization, a compositional task that often involves a voice-leading component (see Fernández & Vico, 2013, for a review). Generally speaking, these models are grounded more in artificial intelligence research than cognitive science research; as a result, there is little emphasis on auditory perception, model interpretability, or corpus analysis. Many of the models are neural networks, which can potentially capture very complex musical principles but typically possess low interpretability (Hild, Feulner, & Menzel, 1984; Hörnel, 2004). Others are rule-based, providing a formal instantiation of the researcher's music-theoretic knowledge without necessarily testing this knowledge against musical practice (Ebcioğlu, 1988; Emura, Miura, & Yanagida, 2008). Both the neural-network approaches and the rule-based approaches seemed ill-suited to our cognitive modeling goals. Moreover, the models generally lack publicly available implementations, which restricts their utility to potential users. We address these concerns in the present work, developing a cognitively motivated voice-leading model and releasing a publicly available implementation in the form of `voicer`, an open-source software package for the R programming language (R Core Team, 2018).

Model

We suppose that a chord sequence can be represented as a series of N tokens, $(x_1, x_2, \ldots, x_N)$, where each token constitutes a *pitch-class chord*, defined as a pitch-class set with known bass pitch class. For example, a IV-V-I cadence in C major would be written as $((\mathbf{5}, 0, 9), (\mathbf{7}, 2, 11), (\mathbf{0}, 4, 7))$, where boldface denotes the bass pitch class. Further, we suppose that we have a candidate generation function, C, which generates a set of candidate voicings for a given pitch-class chord. For example, we might have $C((\mathbf{0}, 4, 7)) = \{\{48, 52, 55\}, \{48, 52, 67\}, \{48, 64, 67\}, \ldots\}$, where each voicing is expressed as a set of MIDI note numbers. Our aim is to model the process by which the musician assigns each pitch-class chord x_i a voicing $X_i \in C(x_i)$.

We suppose that the probability of choosing a voicing X_i varies as a function of certain features of X_i as evaluated with respect to the previous voicing, X_{i-1}. We write f_j for the jth of these features, and define a *linear predictor* $L(X_i, X_{i-1})$ as a weighted sum of these features, where the regression weight of feature f_j is denoted w_j.

$$L(X_i, X_{i-1}) = \sum_j w_j f_j(X_i, X_{i-1}) \tag{1}$$

The linear predictor summarizes the desirability of a particular voicing, aggregating information from the different features. As with traditional regression models, the regression weights determine the contribution of the respective features; for example, a large positive value of w_j means that voicings are preferred when they produce large positive values of f_j, whereas a large negative value of w_j means that large negative values of f_j are preferred.

We suppose that the probability of sampling a given chord voicing is proportional to the exponentiated linear predictor, with the normalization constant being computed by summing over the set of candidate voicings, $C(x_i)$:

$$P(X_i | X_{i-1}, x_i) = \begin{cases} \frac{e^{L(X_i, X_{i-1})}}{\sum_{X \in C(x_i)} e^{L(X, X_{i-1})}} & \text{if } X_i \in C(x_i), \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

This is a sequential version of the *conditional logit* model of McFadden (1974), originally introduced for modeling discrete-choice decisions in econometrics.

The probability of the full sequence of voicings can then be expressed as a product of these expressions:

$$P(X_1, X_2, \ldots, X_N | x_1, x_2, \ldots, x_N) = \prod_{i=1}^{N} P(X_i | X_{i-1}, x_i). \tag{3}$$

where X_0 is a fixed start symbol for all sequences.

Once the candidate voicing generation function C and the features f_i are defined, the regression weights w_i can be optimized on a corpus of chord sequences using maximum-likelihood estimation. Here we perform this optimization using iteratively reweighted least squares as implemented in the `mclogit` package (Elff, 2018). The resulting regression weights quantify the contribution of each feature to voice-leading practice.

Features

Our feature set comprises 12 features that we hypothesized should be useful for the voice-leading model. We designed these features to cover the 13 traditional rules reviewed in Huron's (2001) perceptual account of voice leading (see also, Huron, 2016).

Voice-leading distance. The voice-leading distance between two chords may be defined as the sum distance moved by the implied voice parts connecting the two chords. A chord progression that minimizes voice-leading distance is said to have "efficient" voice leading. Efficient voice leading promotes auditory stream segregation through the pitch proximity principle, which states that the coherence of an auditory stream is improved when its tones are separated by small pitch distances (Huron, 2001, 2016). Correspondingly, we expect our voice-leading model to penalize voice-leading distance when applied to common-practice Western music. We compute voice-leading distance using the minimal voice-leading algorithm of Tymoczko (2006) with a taxicab norm, modified to return pitch distances instead of pitch-class distances. This algorithm generalizes effectively to chords with different numbers of pitches by supposing that several voices can start or end on the same pitch. For example, the optimal voice-leading between C4-E4-G4 and B3-D4-F4-G4 is found to be C4 → B3, C4 → D4, E4 → F4, G4 → G4, which corresponds to a voice-leading distance of $1 + 2 + 1 = 4$ semitones.

Melodic voice-leading distance. Efficient voice leading is likely to be particularly salient for the uppermost voice, on account of the high voice superiority effect (Trainor, Marie, Bruce, & Bidelman, 2014). We capture this hypothesis with a feature termed *melodic voice-leading distance*, defined as the distance between the uppermost voices of successive chords, measured in semitones. We expect our model to penalize melodic voice-leading distance when applied to common-practice Western music.

Pitch height. Harmonic writing in common-practice Western music commonly uses pitches drawn from a three-octave span centered on middle C (C4, 261.63 Hz) (Huron, 2001). This three-octave span corresponds approximately to the combined vocal range of male and female voices, and to the frequency range for which complex tones elicit the clearest pitch percepts (Huron, 2001). We address this phenomenon with three features. *Mean pitch height* computes the absolute difference between the chord's mean pitch height, defined as the mean of its MIDI note numbers, and middle C, corresponding to a MIDI note number of 60. *Treble pitch height* is defined as the distance that the chord's highest note spans above C5 (523.25 Hz), expressed in semitones, and returning zero if the chord's highest note is C5 or lower. Similarly, *bass pitch height* is defined as the distance that the chord's lowest note spans below C3 (130.81 Hz), expressed in semitones, and returning zero if the chord's highest note is C3 or higher. We expect our model to penalize each of these features.

Interference between partials. Any given chord may be realized as an acoustic spectrum, where the spectrum defines the amount of energy present at different oscillation frequencies. The peaks of this spectrum are termed *partials*, and typically correspond to integer multiples of the fundamental frequencies of the chord's constituent tones. Partials separated by small frequency differences are thought to elicit interference effects, in particular *masking* and *roughness* (Harrison & Pearce, in press). Masking, the auditory counterpart to visual occlusion, describes the way in which the auditory system struggles to resolve adjacent pitches that are too similar in frequency. Roughness describes the amplitude modulation that occurs from the superposition of two tones of similar frequencies. Both masking and roughness are thought to have negative aesthetic valence for Western listeners, potentially contributing to the perceptual phenomenon of "dissonance." Correspondingly, musicians may be incentivized to find voice leadings that minimize these interference effects.

Corpus analyses have shown that interference between partials provides a good account of chord spacing practices in Western music, in particular the principle that lower voices should be separated by larger pitch intervals than upper voices (Huron & Sellmer, 1992). Correspondingly, we introduce interference between partials as a voice-leading feature, operationalized using the computational model of Hutchinson and Knopoff (1978) as implemented in the *incon* package (Harrison & Pearce, in press). This model expands each chord tone into its implied harmonics, and sums over all pairs of harmonics in the resulting spectrum, modeling the interference of a given pair of partials as a function of their critical bandwidth distance and the product of their amplitudes. We expect our voice-leading model to penalize high values of this interference feature.

Number of pitches. The number of distinct pitches in a chord voicing must be greater than or equal to the size of the chord's pitch-class set. Larger chords can be produced by mapping individual pitch classes to multiple pitches. Instrumental forces place absolute constraints on this process; for example, a four-part choir cannot produce voicings containing more than four pitches, but can produce voicings with fewer than four pitches by

assigning multiple voices to the same pitches. Other stylistic principles place weaker constraints on this process, which we aim to capture with our voice-leading model. First, we suppose that the musical style defines an ideal number of pitches, and that this ideal can be deviated from with some penalty; for example, a four-part chorale preferentially contains four pitches in each chord voicing, but it is permissible occasionally to use voicings with only three pitches. We operationalize this principle with a feature called *Number of pitches (difference from ideal)*. Second, we suppose that there may be some additional preference for keeping the number of pitches consistent in successive voicings, and operationalize this principle with a feature called *Number of pitches (difference from previous chord)*. We expect the voice-leading model to penalize both of these features.

Parallel octaves/fifths. Octaves and fifths are pitch intervals spanning 12 semitones and 7 semitones respectively. Parallel octaves and parallel fifths occur when two voice parts separated by octaves or fifths both move by the same pitch interval in the same direction. Parallel motion tends to promote perceptual fusion, and this effect is particularly strong for harmonically related tones, such as octaves and fifths (Huron, 2016). The avoidance of parallel octaves and fifths in common-practice voice leading may therefore be rationalized as a mechanism for promoting the perceptual independence of the voices. Conversely, extended sequences of parallel octaves and fifths in the music of Debussy (e.g., *La Cathédrale Engloutie*, 1910, L. 117/10) may encourage listeners to perceive these sequences as single textural streams (Huron, 2016).

We capture this phenomenon using a Boolean feature termed *Parallel octaves/fifths (any parts)* that returns 1 if parallel octaves or fifths (or compound versions of these intervals; a compound interval is produced by adding one or more octaves to a standard interval) are detected between any two parts and 0 otherwise. Voice assignments are computed using Tymoczko's (2006) algorithm, meaning that the feature remains well-defined in the absence of notated voice assignments.

As noted by Huron (2001), parallel octaves and fifths are particularly salient and hence particularly prohibited when they occur between the outer parts. We capture this principle with a Boolean feature termed *Parallel octaves/fifths (outer parts)*, which returns 1 if parallel octaves or fifths are detected between the two outer parts and 0 otherwise.

Exposed octaves (outer parts). Exposed octaves, also known as "hidden octaves" or "direct octaves," occur when two voices reach an interval of an octave (or compound octave) by moving in the same direction. Injunctions against exposed octaves appear in many voice-leading textbooks, but the nature of these injunctions differs from source to source. For example, some say that the rule against exposed octaves applies to any pair of voice parts, whereas others say that the rule only applies to the outer parts; likewise, some say that exposed octaves are acceptable when either of the voices move by step, whereas others say that exposed octaves are only excused when the top line moves by step (see Arthur & Huron, 2016, for a review).

Auditory scene analysis provides a useful perspective on this debate. Like parallel octaves, exposed octaves combine similar motion with harmonic pitch intervals, and are hence likely to promote fusion between the constituent voices. Approaching the interval with stepwise motion may counteract this fusion effect by introducing a competing cue (pitch proximity) that helps the listener differentiate the two voice parts (Huron, 2001, 2016). This provides a potential psychological explanation for why exposed octaves might be excused if they are approached by stepwise motion.

Arthur and Huron (2016) investigated the perceptual basis of the exposed octaves rule, and found that stepwise motion had little effect on perceptual fusion. However, they did find tentative evidence that stepwise motion reduces fusion in the specific case of the uppermost voice moving by step. They explained this effect by noting that fusion comes from the listener interpreting the upper tone as part of the lower tone, resulting in a single-tone percept at the lower pitch. Approaching the lower pitch with stepwise motion presumably reinforces this lower pitch, and therefore has limited consequences for the fusion effect. In contrast, approaching the higher pitch with stepwise motion may encourage the listener to "hear out" this upper pitch, therefore reducing the fusion effect (Arthur & Huron, 2016).

Further work is required before the perceptual basis of exposed octaves is understood fully. For now, we implement a Boolean feature that captures the most consistently condemned form of exposed octaves: those that occur between the outer parts with no stepwise motion in either part. We term this feature *Exposed octaves (outer parts)*. Future work could implement different variants of this feature to capture the different nuances discussed above.

Part overlap. Ascending part overlap occurs when a voice moves to a pitch above that of a higher voice from the preceding chord. Similarly, descending part overlap occurs when a voice moves to a pitch below that of a lower voice from the preceding chord. According to Huron (2001), composers avoid part overlap because it interferes with pitch-based auditory stream segregation,

making it harder for listeners to identify the constituent voices in a chord progression. Correspondingly, we define a Boolean feature termed *Part overlap* that returns 1 when part overlap is detected and 0 otherwise. This feature uses Tymoczko's (2006) algorithm to determine voice assignments for each pitch.

Analysis

We now use our model to analyze a dataset of 370 chorale harmonizations by J. S. Bach, sourced from the virtual music library *KernScores* (Sapp, 2005).[1] These chorales provide a useful baseline application for the model: they are relatively stylistically homogeneous, they have a consistent texture of block chords, and they are considered to be a touchstone of traditional harmonic practice.

These chorales were originally notated as four independent voices. For our analyses, it is necessary to translate these independent voices into sequences of vertical sonorities. We achieve this using *full expansion* (Conklin, 2002): we create a new sonority at each timepoint when a new note onset occurs, with this sonority comprising all pitches already sounding or starting to sound at that timepoint. Because of embellishments such as passing notes and appoggiaturas, these sonorities do not correspond to chords in the conventional sense; deriving a conventional chord sequence would require the services of either a music theorist or a harmonic reduction algorithm (e.g., Pardo & Birmingham, 2002; Rohrmeier & Cross, 2008). We therefore use the term "sonority" to identify the collections of pitch classes identified by the full-expansion algorithm.

Our sequential features (e.g., voice-leading distance) are undefined for the starting sonority in each chorale. We therefore omit all starting sonorities from the model-fitting process. An alternative approach would be to set all sequential features to zero for these starting sonorities.

One of our features—"Number of pitches (difference from ideal)"—is intended to capture the default number of pitches in each voicing for a particular musical style. Since all the chorales in our dataset have four voices, all of which tend to sing throughout the chorale, we set the ideal number of pitches to four.

[1] The collection was originally compiled by C. P. E. Bach and Kirnberger, and later encoded by Craig Sapp. The encoded dataset omits chorale no. 50, the only chorale not in four parts. This dataset is available as the 'bach_chorales_1' dataset in the hcorp package (https://github.com/pmcharrison/hcorp). Source code for our analyses is available at https://doi.org/10.5281/zenodo.2613563.

The model supposes that each sonority has a finite set of candidate voicings. For a given sonority, we enumerate all candidate voicings that satisfy the following conditions:

a) All pitches must range between C2 (65.41 Hz) and B5 (987.77 Hz) inclusive;
b) The voicing must represent the same pitch-class set as the original sonority;
c) The voicing and the original sonority must share the same bass pitch class;
d) The voicing must contain between one and four distinct pitches, reflecting the fact that the chorales were originally written for four voice parts.

Before beginning the analysis, it is worth acknowledging two simplifications we have made when modeling Bach's composition process. First, Bach took his soprano lines from pre-existing chorale melodies, and only composed the lower parts; in contrast, our model recomposes the melody line as well as the lower parts. Correspondingly, our model is not really a simulation of chorale harmonization, but rather a simulation of the Bach chorale style itself. Second, our model assumes that the sonorities are fixed in advance of constructing the voice leadings, which is arguably unrealistic given that the sonorities derived from full expansion include embellishments that are themselves motivated by voice leading, such as passing notes. This simplification is useful for making the analysis tractable, but future work could investigate ways of modeling interactions between harmony and voice leading.

PERFORMANCE

Having fitted the voice-leading model to the corpus, we assess its performance by iterating over each sonority in the corpus and assessing the model's ability to reproduce Bach's original voicings. Different performance metrics can be defined that correspond to different methods for sampling voicings. One approach is to select the voicing with the maximum probability according to the model: in this case, the model retrieves the correct voicing 63.05% of the time. A second approach is to sample randomly from the model's probability distribution: in this case, the model has an average success rate of 44.63%. A third approach is to sample voicings from the model in descending order of probability, until the correct voicing is recovered: on average, this takes 2.55 samples, corresponding to 2.14% of the available voicings. Given that there are on average 102.96 available voicings for each sonority, these figures suggest fairly good generative choices.

MOMENTS

The "Moments" portion of Table 1 describes feature distributions in the original corpus. For example, the first entry indicates that the mean voice-leading distance between successive voicings is 5.96, with a standard deviation of 4.29. Given that these chorales are each voiced in four parts, this implies that each voice part moves on average by 1.49 semitones between each voicing.

It is interesting to examine features corresponding to strict rules that we might expect never to be violated in Bach's work. For example, parallel octaves and fifths are often taught to music students as unacceptable violations of common-practice style, yet our analysis identifies such voice leadings in 1.09% of Bach's progressions. These cases often correspond to passages where Bach introduced voice crossings to avoid parallel progressions (e.g., Figure 1); such voice crossings have no impact on our algorithm, which recomputes all voice leadings using Tymoczko's (2006) algorithm. We decided not to remove such cases, because voice reassignment arguably only partially eliminates the aesthetic effect of these parallel progressions, and because we wish the algorithm to generalize to textures without explicit voice assignment.

WEIGHTS

The "Weights" portion of Table 1 lists optimized weights for each feature, alongside the corresponding standard errors and p values. Consider the voice-leading distance weight, which takes a value of -0.37: this means that increasing voice-leading distance by one unit modifies a voicing's predicted probability by a factor of $exp(-0.37) = 0.69$.

Similar reasoning applies to Boolean features, which can only take two values: "true" (coded as 1) or "false" (coded as 0). For example, part overlap has a weight of -0.67, meaning that a voicing with overlapping parts is $exp(-0.67) = 0.51$ times less likely to occur than an equivalent voicing without overlapping parts. Part overlap is therefore a moderate contributor to voice-leading decisions: something to be avoided but not prohibited. Parallel octaves and fifths, meanwhile, are

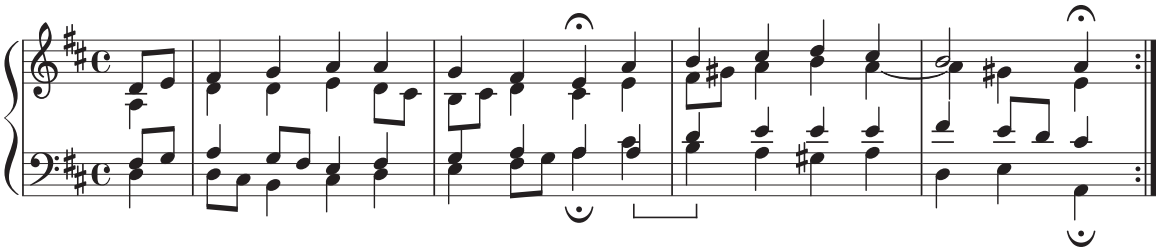

FIGURE 1. J. S. Bach, *Mach's mit mir, Gott, nach deiner Güt'*, BWV 377, bb. 1–4. The two chords immediately after the first fermata imply parallel fifths and octaves that have been only partly mitigated by swapping the bass and tenor parts.

TABLE 1. *Descriptive and Inferential Statistics for the 12 Voice-leading Features as Applied to the Bach Chorale Dataset*

	Moments		Weights			Feature importance	
Feature	*M*	*SD*	Value	*SE*	*p*	Max. probability	Random sample
Voice-leading distance	5.961	4.295	−0.375	0.003	< .001	.553	.379
Melodic voice-leading distance	1.133	1.486	−0.240	0.006	< .001	.164	.100
Treble pitch height (distance above C5)	0.742	1.430	−0.237	0.008	< .001	.103	.052
Bass pitch height (distance below C3)	0.801	1.814	−0.173	0.006	< .001	.072	.038
Interference (Hutchinson & Knopoff, 1978)	0.189	0.076	−8.653	0.231	< .001	.068	.037
Parallel octaves/fifths (Boolean)	0.011	0.104	−2.489	0.062	< .001	.056	.033
Number of pitches (difference from ideal)	0.080	0.278	−1.321	0.035	< .001	.056	.032
Mean pitch height (distance from C4)	2.506	1.793	−0.128	0.005	< .001	.031	.018
Part overlap (Boolean)	0.028	0.164	−0.669	0.041	< .001	.013	.013
Parallel octaves/fifths (outer parts; Boolean)	0.001	0.023	−2.323	0.270	< .001	.008	.004
Exposed octaves (outer parts; Boolean)	0.001	0.037	(0.204)	(0.164)	.214	.000	.000
Number of pitches (difference from previous)	0.125	0.336	(0.008)	(0.034)	.822	.000	.000

Note. Moments provides the mean and standard deviation of feature values in the Bach chorale dataset. *Weights* provides the regression weights for each feature, alongside corresponding standard errors and *p* values. *Feature importance* provides permutation-based importance metrics for each feature.

almost prohibited. Parallel progressions between outer parts are penalized particularly heavily; such progressions reduce a voicing's probability by a factor of $exp(-2.49 - 2.32) = 0.01$.

FEATURE IMPORTANCE

It is difficult to make meaningful comparisons between the weights of continuous features, because each must be expressed in the units of the original feature. This problem is addressed by the permutation-based feature importance metrics in Table 1. These metrics operationalize feature importance as the drop in model performance observed when the trained model is evaluated on a dataset (in this case the Bach chorale corpus) where the feature is randomly permuted (see e.g., Fisher, Rudin, & Dominici, 2018).[2] Table 1 presents two feature importance metrics corresponding to two previously presented performance metrics: the accuracy of maximum-probability samples and the accuracy of random samples. Both metrics indicate that voice-leading efficiency, particularly in the melody line, is the primary contributor to model performance.

It is worth noting that a large feature weight can accompany a small feature importance. For example, parallel fifths/octaves between the outer parts yields a relatively large weight of -2.32, but a relatively small feature importance of 0.01 (maximum-probability sampling). This can be rationalized by the observation that parallel fifths/octaves between the outer parts is essentially prohibited in common-practice voice leading (hence the large weight), but this rule only excludes a tiny proportion of possible voice leadings (hence the small feature importance).

It is also worth noting how each feature's importance will necessarily depend on which other features are present. For example, the weight attributed to "mean pitch height (distance from C4)" is likely to be attenuated by voice-leading distance, because if the previous voicing already had an good mean pitch height, and the next voicing only differs by a small voice-leading distance, then the next voicing is guaranteed to have a fairly good mean pitch height. As a result, the "mean pitch height" feature only needs to give a slight nudge in the appropriate direction to prevent mean pitch height from wandering over time.

STATISTICAL SIGNIFICANCE

Two features received regression weights that did not differ statistically significantly from zero: *Exposed octaves (outer parts)* and *Number of pitches (difference from previous)*. The lack of statistical significance for the exposed-octaves feature is particularly interesting, given how commonly Western music pedagogy prohibits these progressions. Examining the *Moments* column of Table 1, it is clear that such progressions are extremely rare in the chorale dataset, which is surprising given the minimal contribution of the corresponding feature. This suggests that these progressions are being penalized by other features. Three such features seem particularly relevant: *Voice-leading distance*, *Melodic voice-leading distance*, and *Mean pitch height*. According to our definitions, exposed octaves only occur when both outer parts move by three or more semitones; such large movements are likely to be heavily penalized by the voice-leading distance features. Furthermore, the two voices must progress in similar motion, thereby inducing a significant change in mean pitch height. Assuming that the previous voicing was already at a suitable mean pitch height, this is likely to take the voicing to an unsuitable mean pitch height, resulting in penalization by the *Mean pitch height* feature. In sum, therefore, it seems plausible that the exposed-octaves feature is made redundant by the other features.

The non-significant contribution of the feature *Number of pitches (difference from previous)* is arguably unsurprising given the corpus being modeled. Each of these chorales is written for four voices, and so the primary pressure on the number of pitches in the sonority is likely to be the goal of providing these four voices with distinct lines; deviations from this four-pitch norm are generally rare and quickly resolved. This phenomenon can be captured by the feature *Number of pitches (difference from ideal)*, making the feature *Number of pitches (difference from previous)* unnecessary. However, this latter feature may become more important in corpora where the number of voices is less constrained, such as in keyboard music.

Generation

The probabilistic model developed in the previous section can be directly applied to the automatic generation of voice leadings for chord sequences. Given a prespecified chord sequence, the model defines a probability distribution over all possible voice leadings for that chord sequence, which factorizes into probability distributions for each chord voicing conditioned on the previous chord voicing. It is straightforward to sample from this factorized probability distribution: simply iterate from the first to the last chord in the sequence, and sample each voicing according to the probability distribution

[2] Note that the feature is only permuted in the test dataset, not the training dataset.

defined by the conditional logit model, using the sampled voicing at position i to define the feature set for chord voicings at position $i + 1$.

If our goal is to approximate a target corpus as well as possible, then this random sampling is a sensible approach. However, if our goal is to generate the best possible voice leading for a chord sequence, then we must identify some objective function that characterizes the quality of a chord sequence's voice leading and optimize this objective function.

Here we propose optimizing the sum of the model's linear predictors. As defined previously, the linear predictor characterizes a given chord voicing as a weighted sum of feature values, with this linear predictor being exponentiated and normalized to estimate the probability of selecting that voicing. The linear predictor might be interpreted as the attractiveness of a given voicing, as inversely related to features such as voice-leading distance and interference between partials.

Optimizing the sum of the linear predictors is subtly different to optimizing for probability. Optimizing for probability means maximizing the ratio of the exponentiated linear predictors for the chosen voicing to the exponentiated linear predictors for the alternative voicings. This maximization does not necessarily entail high values of the linear predictor; in perverse cases, high probabilities may be achieved when the chosen voicing is simply the best of a very bad set of candidates. We wish to avoid such cases, and to identify chord voicings that possess good voice-leading attributes in an absolute sense, not simply relative to their local competition.

The space of all possible voice leadings is large: given 100 candidate voicings per chord, a sequence of 80 chords has 10^{160} potential voice-leading solutions. It is clearly impractical to enumerate these voice leadings exhaustively. A simple "greedy" strategy would be to choose the chord voicing with the highest linear predictor at each chord position; however, this is not guaranteed to maximize the sum of linear predictors across all chord positions. Instead, we take a dynamic-programming approach that deterministically retrieves the optimal voice-leading solution while restricting the number of linear predictor evaluations to approximately a^2n, where a is the number of candidate voicings for each chord and n is the number of chords in the sequence. This approach simplifies the computation by taking advantage of the fact that none of our features look back beyond the previous chord's voicing. See *Appendix* for details.

Several of the features, such as voice-leading distance and part overlap, are undefined for the first chord in the sequence. Correspondingly, the first chord of each sequence was excluded from the model-fitting process described in *Analysis*. When generating from the model, however, it is inappropriate to exclude these chords from the optimization. Instead, we set all context-dependent features to zero for the first chord of each sequence (in fact, any numeric constant would have the same effect). The initial chord voicings are then optimized according to the context-independent features, such as interference between partials and mean pitch height.

Figure 2 demonstrates the algorithm on the first ten sonorities of the chorale dataset: *Aus meines Herzens Grunde*, BWV 269.[3] For comparison purposes, Figure 2A displays J. S. Bach's original voice leading, and Figure 2B displays a heuristic voice leading where the bass pitch class is played in the octave below middle C and the non-bass pitch classes are played in the octave above middle C, after Harrison and Pearce (2018a). Figure 2C displays the voice leading produced by the new algorithm, using regression weights as optimized on the original corpus, and generating candidate chords according to the same procedure as described in *Analysis*. Unlike the heuristic algorithm, the new algorithm consistently employs four notes in each chord, creating a richer voice leading that is more representative of the original chorale harmonization. The new algorithm successfully avoids the two parallel fifths produced in the last two bars by the heuristic algorithm, and achieves considerably smoother voice leading throughout.

In chorale harmonizations the soprano line is typically constrained to follow the pre-existing chorale melody. We can reproduce this behavior by modifying the candidate voicing generation function so that it only generates voicings with the appropriate soprano pitches. Figure 2D displays the voice leading produced when applying this constraint. Our implementation also supports further constraints such as forcing particular chord voicings to contain particular pitches, or alternatively fixing entire chord voicings at particular locations in the input sequence.

We were interested in understanding how the trained model would generalize to different musical styles. In harmony perception studies, it is often desirable to present participants with chord sequences derived from pre-existing music corpora, such as the McGill Billboard corpus (Burgoyne, 2011) and the iRb corpus (Broze &

[3] Source code is available at https://doi.org/10.5281/zenodo.2613563. Generated voice leadings for all 370 chorales are available at https://doi.org/10.5281/zenodo.2613646.

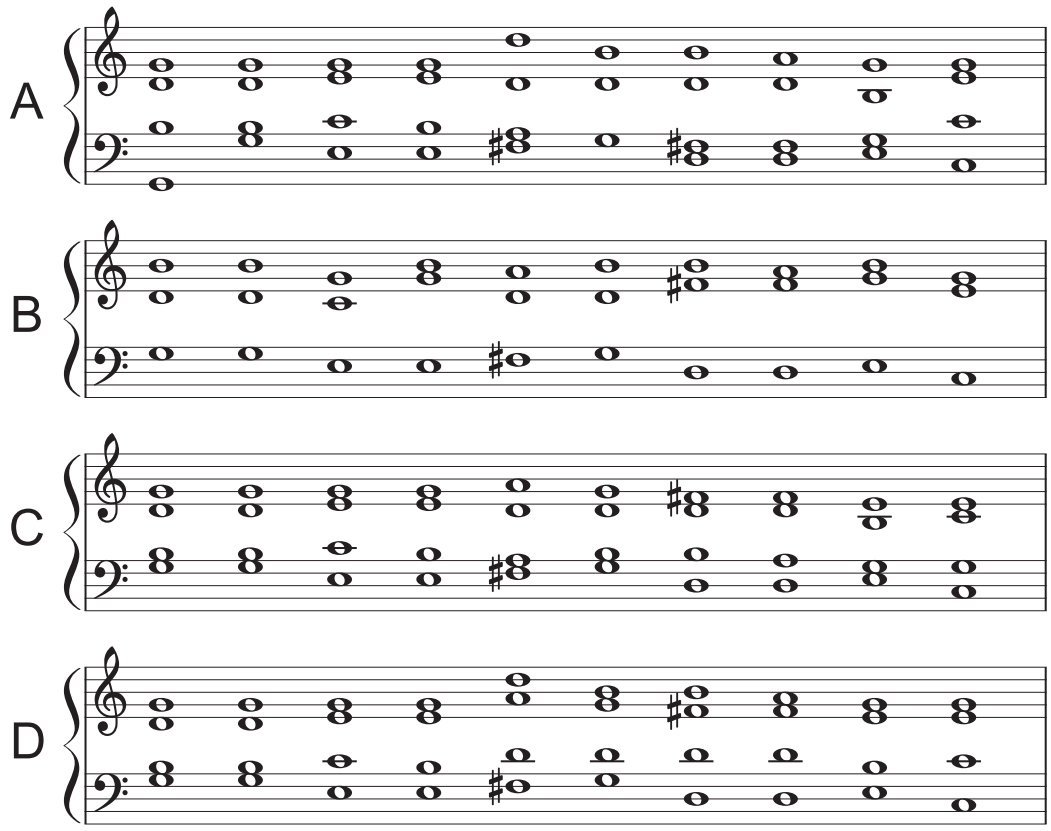

FIGURE 2. Example voice leadings for J. S. Bach's chorale *Aus meines Herzens Grunde* (BWV 269), chords 1–10. A) Bach's original voice leading. B) Heuristic voice leading. C) New algorithm. D) New algorithm with prespecified melody.

Shanahan, 2013). Unfortunately, these corpora just provide chord symbols, not fully voiced chords, and so the researcher is tasked with creating voice leadings for these chord sequences. We had yet to identify suitable datasets of voiced chord sequences for popular or jazz music, and therefore wished to understand whether Bach chorales would be sufficient for training the algorithm to generate plausible voice leadings for these musical styles.

From an auditory scene analysis perspective, there are clear differences between Bach chorales and popular/jazz harmony. The chorales consistently use four melodically independent voices, and Bach's voice-leading practices are consistent with the compositional goal of maximizing the perceptual independence of these voices while synchronizing text delivery across the vocal parts (Huron, 2001, 2016). In contrast, harmony in popular and jazz music is often delivered by keyboards or guitars, both of which produce chords without explicit voice assignment, with the number of distinct pitches in each chord often varying from chord to chord. Correspondingly, voice independence seems likely to be less important in popular/jazz harmony than in Bach chorales. Nonetheless, we might still expect popular/jazz musicians to pay attention to the perceptual independence of the outer parts, since these voices are particularly salient to the listener even when the voice parts are not differentiated by timbre. We might also expect popular/jazz listeners to prefer efficient voice leadings, even if they are not differentiating the chord progression into separate voices, because efficient voice leading helps create the percept of a stable textural stream (Huron, 2016). In summary, therefore, there are reasons to expect some crossover between voice-leading practices in Bach chorales and voice-leading practices in popular/jazz music.

Figures 3 and 4 demonstrate the application of the chorale-trained model to the first ten chords of two such corpora: the Billboard popular music corpus (Burgoyne, 2011), and the iRb jazz corpus (Broze & Shanahan, 2013). We use both datasets as translated to pitch-class notation by Harrison and Pearce (2018b), and use the same model configuration as for the Bach chorale voicing.

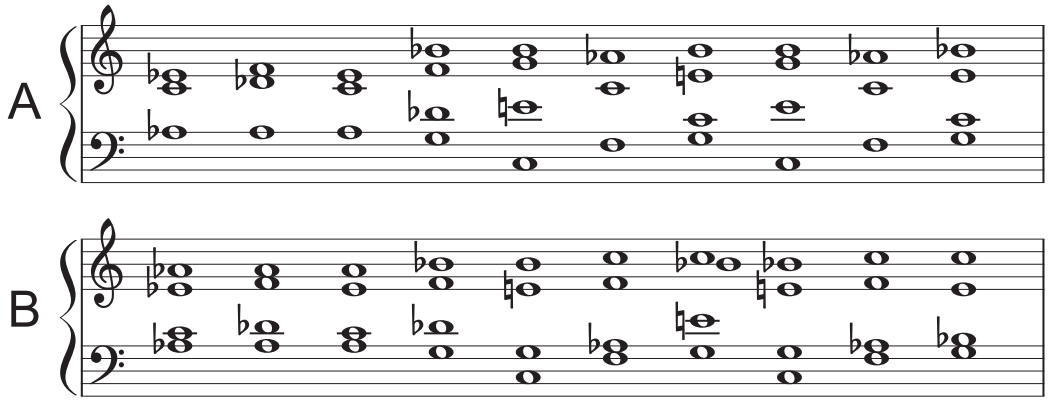

FIGURE 3. Example voice leadings for the first 10 chords of James Brown's *I don't mind*. A) Heuristic voice leading. B) New algorithm.

Figures 3A and 3B correspond to the first ten bars of the popular corpus, from the song *I don't mind* by James Brown. Figure 3A displays the heuristic algorithm described earlier, and Figure 3B displays the new algorithm's voicing. Unlike the heuristic algorithm, the new algorithm maintains four-note voicings at all times, producing a richer and more consistent sound. The voice-leading efficiency is also considerably improved, particularly in the melody line.

Figures 4A and 4B correspond to the first ten bars of the jazz corpus, from the composition *26-2* by John Coltrane. As before, Figures 4A and 4B correspond to the heuristic and new algorithms respectively. At first sight, the new algorithm produces some unusual voice leadings: for example, the tenor part jumps by a tritone between the fourth chord and the fifth chord. One might expect this inefficient voice leading to be heavily penalized by the model. However, the model considers this voice leading to be relatively efficient, as Tymoczko's (2006) algorithm connects the two voicings by approaching the lower two notes of the fifth chord (C, G) from the bass note of the previous chord (G), and approaching the second-from-top note in the fifth chord (E) from the second-from-bottom note in the fourth chord (D♭). This suggested voice assignment is indeed plausible when the extract is performed on a keyboard instrument, but it could not be realized by a four-part ensemble of monophonic instruments. For such applications, it would be worth modifying Tymoczko's (2006) algorithm to set an upper bound on the number of inferred voices.

We have implemented these algorithms in an open-source software package called `voicer`, written for the R programming language, and coded in a mixture of R and C++. The source code is available from the open-source repository https://github.com/pmcharrison/voicer and permanently archived at https://doi.org/10.5281/zenodo.2613565. Appendix B provides an introduction to the package.

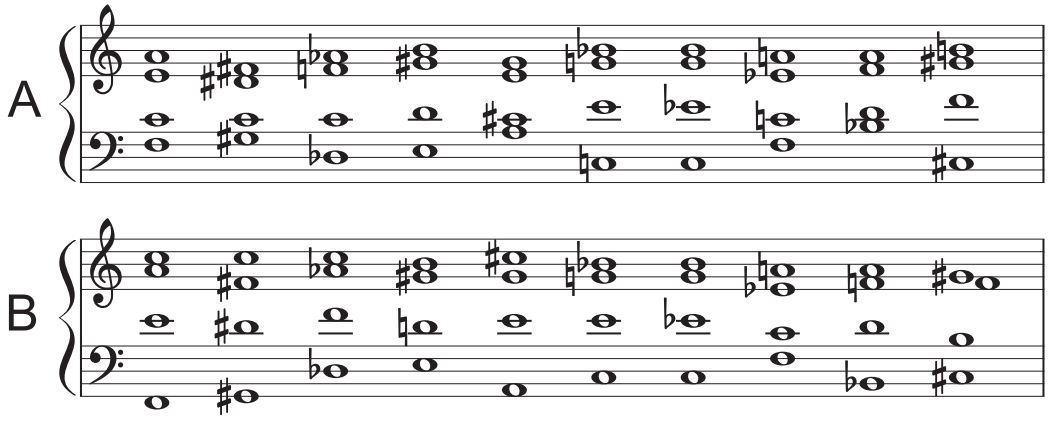

FIGURE 4. Example voice leadings for the first 10 chords of John Coltrane's *26-2*. A) Heuristic voice leading. B) New algorithm.

Discussion

We have introduced a new model for the analysis and generation of voice leadings. This model uses perceptually motivated features to predict whether a given voice leading will be considered appropriate in a particular musical context. Applied to a dataset of 370 chorale harmonizations by J. S. Bach, this model delivered quantitative evidence for the relative importance of different musical features in determining voice leadings. Applied to generation, the model demonstrated an ability to create plausible voice leadings for pre-existing chord sequences, and to generalize to musical styles dissimilar to the Bach chorales upon which it was trained.

Combining analysis with generation provides a powerful way to examine which principles are sufficient to explain voice-leading practice. While the analysis stage provides quantitative support for the importance of different musical features in voice leading, the generation stage can provide a litmus test for the sufficiency of the resulting model. Examining the outputs of the model, we can search for ways in which the model deviates from idiomatic voice leading, and test whether these deviations can be rectified by incorporating additional perceptual features into the model. If so, we have identified an additional way in which voice leading may be explained through auditory perception, after Huron (2001, 2016); if not, we may have identified an important cultural component to voice-leading practice. To this end, we have released automatically generated voicings for the full set of 370 Bach chorale harmonizations;[4] we hope that they will provide useful material for identifying limitations and potential extensions of the current approach.

The existing literature already suggests several additional features that might profitably be incorporated into the voice-leading model. One example is the "leap away rule," which states that large melodic intervals (leaps) are better situated in the outer voices than the inner voices, and that these intervals should leap away from the other voices rather than towards the other voices (Huron, 2016). This should be straightforward to implement computationally. A second example is the "follow tendencies rule," which states that the progressions of individual voices should follow the listener's expectations, which may themselves derive from the statistics of the musical style (*schematic expectations*), the statistics of the current musical piece (*dynamic expectations*), or prior exposure to the same musical material (*veridical expectations*) (Huron, 2016). Schematic expectations could be operationalized by using a dynamic key-finding algorithm to represent the sonority as scale degrees (e.g., Huron & Parncutt, 1993), and then evaluating the probability of each scale-degree transition with respect to a reference musical corpus; dynamic expectations could be operationalized in a similar manner, but replacing the reference corpus with the portion of the composition heard so far. Veridical expectations would require more bespoke modeling to capture the particular musical experience of the listener. An interesting possibility would be to unite these three types of expectation using Pearce's (2005) probabilistic model of melodic expectation (see also Sauvé, 2017). Further rules that could be implemented include the "semblant motion rule" (avoid similar motion) and the "nonsemblant preparation rule" (avoid similar motion where the voices employ unisons, octaves, or perfect fifths/twelfths) (Huron, 2016).

Our model also has practical applications in automatic music generation. For example, a recurring problem in music psychology is to construct experimental stimuli representing arbitrary chord sequences, which often involves the time-consuming task of manually constructing voice leadings. Our model could supplant this manual process, bringing several benefits including: a) *scalability*, allowing the experimental design to expand to large stimulus sets; b) *objectivity*, in that the voice leadings are created according to formally specified criteria, rather than the researcher's aesthetic intuitions; c) *reproducibility*, in that the methods can be reliably reproduced by other researchers.

Interpreted as a model of the compositional process, the model assumes that chords are determined first and that voice leading only comes later. This may be accurate in certain musical scenarios, such as when performers improvise from figured bass or from lead sheets, but it is clearly not a universal model of music composition. A more universal model might include some kind of alternation between composing the harmonic progression and composing the voice leading, so that the composer can revise the harmonic progression if it proves impossible to find a satisfactory voice leading.

The model also assumes a one-to-one mapping between the chords of the underlying harmonic progression and the chord voicings chosen to represent it. While this assumption may hold true for certain musical exercises, it is not universally valid for music composition. For example, an improviser playing from figured bass may choose to extend a single notated chord into multiple vertical sonorities, for example through arpeggiation or through the introduction of passing notes. It would be interesting to model this process explicitly. One approach would be to use the original model to generate

[4] https://doi.org/10.5281/zenodo.2613646

block chords at the level of the harmonic rhythm, and then to post-process these block chords with an additional algorithm to add features such as passing notes and ornamentation.

While the model deserves further extension and validation, it seems ready to support ongoing research in music psychology, music theory, and computational creativity. Our R package, `voicer`, should be useful in this regard: It provides a convenient interface for analyzing voice leadings in musical corpora and for generating voice leadings for chord sequences. The ongoing development of this package may be tracked at its open-source repository (https://github.com/pmcharrison/voicer).

Author Note

Peter Harrison is now at the Computational Auditory Perception Group, Max Planck Institute for Empirical Aesthetics.

This paper first appeared as an unpublished preprint at https://doi.org/10.31234/osf.io/wrgj7. Peter Harrison was supported by a doctoral studentship from the EPSRC and AHRC Centre for Doctoral Training in Media and Arts Technology (EP/L01632X/1).

Correspondence concerning this article should be addressed to Peter M. C. Harrison, Max-Planck-Institut für empirische Ästhetik, Grüneburgweg 14, 60322 Frankfurt am Main. E-mail: peter.harrison@ae.mpg.de

References

Arthur, C., & Huron, D. (2016). The direct octaves rule: Testing a scene-analysis interpretation. *Musicae Scientiae*, *20*(4), 495–511. DOI: 10.1177/1029864915623093

Bregman, A. S. (1990). *Auditory scene analysis: The perceptual organization of sound*. Cambridge, MA: MIT Press.

Broze, Y., & Shanahan, D. (2013). Diachronic changes in jazz harmony: A cognitive perspective. *Music Perception*, *31*, 32–45. DOI: 10.1525/rep.2008.104.1.92

Burgoyne, J. A. (2011). *Stochastic processes and database-driven musicology* (PhD thesis). McGill University, Montréal, Canada.

Conklin, D. (2002). Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand, & A. Smaill (Eds.), *Music and artificial intelligence: Proceedings of ICMAI 2002* (pp. 32–42). Berlin, Germany: Springer-Verlag.

Ebcioğlu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal*, *12*(3), 43–51.

Elff, M. (2018). *mclogit: Mixed conditional logit models*. Retrieved from https://CRAN.R-project.org/package=mclogit

Emura, N., Miura, M., & Yanagida, M. (2008). A modular system generating Jazz-style arrangement for a given set of a melody and its chord name sequence. *Acoustical Science and Technology*, *29*, 51–57. DOI: 10.1250/ast.29.51

Fernández, J. D., & Vico, F. (2013). AI methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, *48*, 513–582. DOI: 10.1613/jair.3908

Fisher, A., Rudin, C., & Dominici, F. (2018). *Model class reliance: Variable importance measures for any machine learning model class, from the "Rashomon" perspective*. Retrieved from http://arxiv.org/abs/arXiv:1801.01489v2

Harrison, P. M. C., & Pearce, M. T. (2018a). Dissociating sensory and cognitive theories of harmony perception through computational modeling. In R. Parncutt & S. Sattmann (Eds.), *Proceedings of ICMPC15/ESCOM10*. Graz, Austria: ICMPC. DOI: 10.31234/osf.io/wgjyv

Harrison, P. M. C., & Pearce, M. T. (2018b). An energy-based generative sequence model for testing sensory theories of Western harmony. In X. H Gómez., E. Humphrey, & E. Benetos (Eds.), *Proceedings of the 19th International Society for Music Information Retrieval Conference* (pp. 160–167). Paris, France: ISMIR.

Harrison, P. M. C., & Pearce, M. T. (in press). Simultaneous consonance in music perception and composition. *Psychological Review*.

Hild, H., Feulner, J., & Menzel, W. (1984). HARMONET: A neural net for harmonizing chorales in the style of J. S. Bach. In R. P. Lippmann, J. E. Moody, & D. S. Touretzky (Eds.), *Advances in neural information processing systems* (pp. 267–274). Los Altos, CA: Morgan Kaufmann.

Hörnel, D. (2004). CHORDNET: Learning and producing voice leading with neural networks and dynamic programming. *Journal of New Music Research*, *33*, 387–397. DOI: 10.1080/0929821052000343859

Huron, D. (2001). Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, *19*, 1–64. DOI: 10.1525/mp.2001.19.1.1

Huron, D. (2016). *Voice leading: The science behind a musical art*. Cambridge, MA: MIT Press.

Huron, D., & Parncutt, R. (1993). An improved model of tonality perception incorporating pitch salience and echoic memory. *Psychomusicology*, *12*, 154–171.

Huron, D., & Sellmer, P. (1992). Critical bands and the spelling of vertical sonorities. *Music Perception*, *10*, 129–149.

Hutchinson, W., & Knopoff, L. (1978). The acoustic component of Western consonance. *Journal of New Music Research*, *7*, 1–29. DOI: 10.1080/09298217808570246

McFadden, D. (1974). Conditional logit analysis of qualitative choice behaviour. In P. Zarembka (Ed.), *Frontiers in econometrics* (pp. 105–142). New York: Academic Press.

Pardo, B., & Birmingham, W. P. (2002). Algorithms for chordal analysis. *Computer Music Journal*, *26*(2), 27–49. DOI: 10.1162/014892602760137167

Pearce, M. T. (2005). *The construction and evaluation of statistical models of melodic structure in music perception and composition* (PhD thesis). City University, London, London, UK.

R Core Team. (2018). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rohrmeier, M. A., & Cross, I. (2008). Statistical properties of tonal harmony in Bach's chorales. In K. Miyazaki, Y. Hiraga, M. Adachi, Y. Nakajima, & M. Tsuzaki (Eds.), *Proceedings of the 10th International Conference on Music Perception and Cognition*, 619–627. Sapporo, Japan: ICMPC.

Sapp, C. S. (2005). Online database of scores in the Humdrum file format. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR 2005)* (pp. 664–665). London, UK: ISMIR.

Sauvé, S. A. (2017). *Prediction in polyphony: modelling musical auditory scene analysis* (PhD thesis). Queen Mary University of London, London, UK.

Trainor, L. J., Marie, C., Bruce, I. C., & Bidelman, G. M. (2014). Explaining the high voice superiority effect in polyphonic music: Evidence from cortical evoked potentials and peripheral auditory models. *Hearing Research*, *308*, 60–70. DOI: 10.1016/j.heares.2013.07.014

Tymoczko, D. (2006). The geometry of musical chords. *Science*, *313*(5783), 72–74. DOI: 10.1126/science.1126287

Tymoczko, D. (2011). *A geometry of music*. New York: Oxford University Press.

Appendix A

A dynamic programming algorithm for maximizing the sum of the linear predictors over all chord transitions. Note that all vectors are 1-indexed.

input: `candidates`, a list of length `N`; `candidates[i]` lists the candidate voicings for chord `i`
output: `chosen`, a list of length `N`; `chosen[i]` identifies the chosen voicing for chord `i`

```
best_scores ← list(N) // total scores of best paths to each chord voicing
best_prev_states ← list(N) // best previous voicing for each chord voicing
best_scores[1] ← vector(length(candidates[1]))
for j ← 1 to length(candidates[1]) do
      best_scores[1][j] ← f(NULL, candidates[1][j])
end
for i ← 2 to N do
      best_scores[i] ← vector(length(candidates[i]))
      for j ← 1 to length(candidates[i]) do
            best_prev_states[i][j] ← 1
            best_scores[i][j] ← f(candidates[i – 1][1], candidates[i][j])
                  for k ← 2 to length(candidates[i – 1])} do
                  new_score ← f(candidates[i–1][k], candidates[i][j])
                  if new_score > best_scores[i][j] then
                        best_prev_states[i][j] ← k
                        best_scores[i][j] ← new_score
                  end
            end
      end
end
chosen ← vector(N)
chosen[N] ← which_max_j(best_scores[N][j])
for n ← N – 1 to 1 do
      chosen[n] ← best_prev_states[n + 1][chosen[n + 1]]
end
return chosen
```

Appendix B

Having installed the `voicer` package from its open-source repository (github.com/pmcharrison/voicer), the following code uses the package to voice a perfect (or authentic) cadence:

Code:

```
library(voicer)

library(hrep)

library(magrittr)

# Each chord is represented as a sequence of MIDI note numbers.

# The first number is the bass pitch class.

# The remaining numbers are the non-bass pitch classes.

list(pc_chord("0 4 7"), pc_chord("5 0 2 9"),

   pc_chord("7 2 5 11"), pc_chord("0 4 7")) %>% vec("pc_chord") %>%

 voice(opt = voice_opt(verbose = FALSE)) %>% print(detail = TRUE)
```

Output:

```
[[1]] Pitch chord: 48 64 67 72

[[2]] Pitch chord: 53 62 69 72

[[3]] Pitch chord: 55 62 65 71

[[4]] Pitch chord: 48 55 64 72
```

By default, `voicer` uses the same regression weights and voicing protocol as presented in the current paper. However, it is easy to modify this configuration, as demonstrated in the following example:

Code:

```
library(voicer)

library(hrep)

library(magrittr)

chords <- list(pc_chord("0 4 7"), pc_chord("5 0 9"),

        pc_chord("7 2 11"), pc_chord("0 4 7")) %>% vec("pc_chord")

# Modify the default weights to promote parallel fifths/octaves

weights <- voice_default_weights

weights["any_parallels"] <- 100

voice(chords, opt = voice_opt(verbose = FALSE,

                weights = weights,

                min_notes = 3,

                max_notes = 3)) %>% print(detail = TRUE)
```

Output:

```
[[1]] Pitch chord: 48 55 64

[[2]] Pitch chord: 53 60 69

[[3]] Pitch chord: 55 62 71

[[4]] Pitch chord: 60 67 76
```

The `voicer` package also exports functions for deriving regression weights from musical corpora, and using these new weights to parametrize the voicing algorithm. The following example derives regression weights from the first two pieces in the Bach chorale dataset, and uses these weights to voice a chord sequence.

Code:

```
if (!requireNamespace("hcorp"))
 devtools::install_github("pmcharrison/hcorp")
library(voicer)
library(hrep)
# Choose the features to model
features <- voice_features()[c("vl_dist", "dist_from_middle")]
# Compute the features
corpus <- hcorp::bach_chorales_1[1:2]
corpus_features <- voicer::get_corpus_features(
 corpus, min_octave = -2, max_octave = 1, features = features,
 revoice_from = "pc_chord", min_notes = 1, max_notes = 4,
 verbose = FALSE)
# Model the features
mod <- model_features(corpus_features, perm_int = FALSE, verbose = FALSE)
as.data.frame(mod$weights)
```

Output:

```
      feature  estimate  std_err     z      p
1     vl_dist -0.5320266 0.03441282 -15.460129 6.446884e-54
2 dist_from_middle -0.1910925 0.06838340 -2.794428 5.199166e-03
```

Code:

```
# Voice a chord sequence
chords <- list(pc_chord("0 4 7"), pc_chord("5 0 9"),
        pc_chord("7 2 11"), pc_chord("0 4 7")) %>% vec("pc_chord")
voice(chords, opt = voice_opt(weights = mod,
               features = features,
               verbose = FALSE)) %>% print(detail = TRUE)
```

Output:

```
[[1]] Pitch chord: 36 52 72 79

[[2]] Pitch chord: 41 53 72 81

[[3]] Pitch chord: 43 50 71 79

[[4]] Pitch chord: 48 52 72 79
```