Christopher Raphael and Joshua Stoddard
Department of Mathematics and Statistics
University of Massachusetts at Amherst
Amherst, MA 01003-4515 USA
{raphael, stoddard}@math.umass.edu

Functional Harmonic Analysis Using Probabilistic Models

A variety of musical analysis techniques, often collectively referred to as functional harmonic analysis, represents a musical passage as a sequence of chords. The chords are expressed in terms of their function (e.g., *dominant* or *tonic*), often written with corresponding Roman numerals like V or I. Each chord is analyzed in the context of a key, which might modulate over time. This article focuses on the study of algorithms for this type of analysis.

An obvious application of algorithmic harmonic analysis would be locating musical examples in a database matching a particular harmonic query, for example, ''What are the earliest examples of the use of German augmented sixth chords or Neapolitan chords?'' ''Which Beatles songs have deceptive cadences?'' ''Where can I buy the piece I heard on the radio with the harmonic progression I vi IV V I repeated many times?'' It is likely that such applications will be most useful to musicologists, because the mere formulation of such queries requires a more sophisticated understanding of harmony than would be expected of an average music enthusiast.

Another application of this type of harmonic analysis could be found in score-writing programs. When these programs produce musical notation from MIDI files, they often produce incorrect spellings of accidentals. Many of these errors could be avoided if the harmonic content were understood at a level beyond the nominal key signature, as in Chew and Chen (2003).

A more subtle, yet perhaps more important, application might be one of representation. Harmonic analysis reduces music to a one-dimensional sequence of symbols from a small alphabet. The one-dimensional nature of this representation lends it to the wealth of search techniques treating strings as the basic unit of study. Such string-matching algorithms can find the string in a database minimizing a variety of edit-like distances in linear time. Pickens et al. (2002) show an example of a technique somewhat like harmonic analysis for representation and retrieval.

More generally, the one-dimensional musical reduction afforded by harmonic analysis might form the basis for genre classification or the construction of various music similarity metrics. Perhaps such analysis might even serve as a useful compositional tool by making unexpected links between musical passages, as in Peter Schickele's comical musical pastiches.

While we are interested in these applications, we find the study of the cognitive, or ''artificial intelligence,'' aspect of harmonic analysis ample motivation by itself. Our basic approach is statistical; this orientation and methodology distinguishes our work from most other efforts we know. The most significant benefit of the probabilistic modeling we employ is the ability to learn aspects of our model in an unsupervised manner, for instance, using generic (unmarked) MIDI data. However, we also inherit computational machinery that identifies the best harmonic parse globally. In addition, we prefer the transparency and honesty of a clearly specified probabilistic model.

That said, we find common ground with several other previous efforts in harmonic analysis. Krumhansl (1990) identifies key by matching a histogram of pitches to a collection of possible key templates. Although our approach simultaneously identifies chord and key, the actual computation that measures the appropriateness of a particular key hypothesis is similar to that of Krumhansl. We share with Temperley and Sleator (1999) the recognition that rhythmic content is useful in harmonic analysis and the notion that harmonic analyses that fluctuate rapidly between keys are implausible and should be discouraged or penalized. Pardo (2002) builds upon this approach with a dynamic-programming algorithm optimized over the

Computer Music Journal, 28:3, pp. 45–52, Fall 2004

exponential number of segmentations in a computationally efficient manner. Similarly, dynamic programming is also fundamental to our work.

An overview of the approaches to algorithmic harmonic analysis is presented by Barthelemy and Bonardi (2001). Most approaches similar in scope to ours are rule-based: the music is reduced and recognized through a series of deterministic state transformations (merges, simplifications, intermediate labelings, etc.) moving systematically toward a final representation. In our view, there are two principal disadvantages of rule-based approaches. First, such schemes fail to articulate any measure of ''goodness'' of the possible ''answers'' and hence do not formulate the problem clearly. Second, rule-based schemes balance each decision or transformation precariously on the shoulders of previous decisions, hence irrevocably propagating errors forward. Our approach, like that of Pardo (2002), is decidedly not rule-based.

Another significant difference between our approach and all others we know is our simultaneous recognition of chord and key. The hope here is that the more structured sequence of chord functions (e.g., tonic, dominant) will help guide the analysis when the choice of key is ambiguous. For instance, repeated alternation of C-major and F-major triads argues strongly for the key of F when the chords are labeled as dominant and tonic, respectively. In contrast, a random rearrangement of the same pitches is more harmonically ambiguous. In this case, and many others, the identification of the intermediate functional labeling is important for understanding the harmonic structure.

With the exception of the Extending the Model section of this article, which discusses future directions, we treat a model analogous to the ''bag of words'' model from text information retrieval: given a key and chord, the observed notes are a random sample from a distribution characterized by the key and chord. The main limitation of such a ''bag of notes'' model is its failure to represent any ''linear'' characteristics that emerge when the music is viewed in terms of voices, such as suspensions, appoggiaturas, etc. In this way, our analysis ends with the segmentation of the music into regions, labeled with key and functional chord. Although we do not attempt any deeper harmonic analysis, we acknowledge that a full harmonic understanding would require further study. However, the Extending the Model section suggests a more powerful model that might serve as the backbone for a more sophisticated analysis.

Finally, we acknowledge that the notion of functional harmony is by no means universal. When music does not follow the assumptions of our model, the analysis will of course not be especially meaningful.

The Model

Our harmonic analysis is based on pitch and rhythm. From the outset, we acknowledge that there are likely no two elements of music that do not interact in some musical situation; therefore, limiting our treatment to these two elements undoubtedly loses some relevant information. However, in most musical contexts, the human listener can easily base a plausible harmonic analysis solely on pitch and rhythm. This and the rather obvious virtue of beginning with simplicity lead us to start here.

Our harmonic analysis is performed on a fixed musical period, q, say a measure ($q = 1$) or half measure, ($q = 1/2$). To this end, we partition the pitches in our musical composition into a sequence of subsets $y_1, \ldots, y_N$, where y_n is the collection of pitches whose onset time, in measures, lies in the interval $[nq, (n+1)q)$. We notate this collection explicitly as $y_n = \mathrm{y}_n^1, \ldots, \mathrm{y}_n^K$, where the number of pitches, K, depends on n. Our analysis is based only on pitch class, so we regard the pitches as elements of {0, 1, . . ., 11}, where C and B-sharp = 0, C-sharp and D-flat = 1, and so on. While the extension of our approach to include enharmonic spellings is obvious, MIDI data clearly form the lion's share of available test cases at present; because MIDI does not use enharmonic spellings, we do not model them.

Our goal is to associate a key and chord describing the harmonic function of each period, y_n. Thus each y_n will be labeled with an element of

$$L = T \times M \times C = \{0, \ldots, 11\} \times \{\text{major, minor}\} \times \{\text{I, II, } \ldots \text{, VII}\} \quad (1)$$

where T, M, and C stand for tonic, mode, and chord, respectively. For instance, $(t, m, c) = (2,$ major, II) would represent the triad in the key of 2 = D major built on the II = second scale degree, which contains pitches E, G, and B. (For simplicity, we will ignore the usual convention of using lower- and upper-case Roman numerals for minor and major triads, respectively.) Although it is possible to use a broader range of possible chords (in fact, we do in our experiments), nothing significant is lost by limiting ourselves to the seven basic triads in this discussion. Similarly, it would be possible to include more modes than the basic major and harmonic minor that we treat here. We do not currently model chord inversion in this work, because representing only the pitch class of a note makes it impossible to recover inversion information.

Consider the common situation, occurring in a clearly established C-major context, in which we encounter the chord progression C major, D major, G major. In this case, it appears that D acts as the ''dominant'' of the dominant chord G major—a so-called secondary dominant, often notated ''V/V.'' Our basic vocabulary of chords does not include any kind of secondary chord, but such interpretations can still be represented using key modulation. For instance, the above example can be represented as (c = 0, major, I), (g = 7, major, V), (g = 7, major, I). Clearly our representation allows for a rich variety of secondary functionality while avoiding murky distinctions between secondary function and actual modulation.

Let $X_1, X_2, \ldots, X_N$ be the sequence of harmonic labels where X_n is a member of the label set L. We model this sequence probabilistically as a homogeneous Markov chain

$$p(x_{n+1}|x_1, \ldots, x_n) = p(x_{n+1}|x_n) \quad (2)$$

The Markov assumption is of course only an approximation; however, we believe it captures quite a bit of musical structure, especially when one considers the simplicity of the model. For instance, the fact that keys tend to remain constant for relatively long periods of time is easily expressed in terms of a Markov model: the key at each time period is, with high probability, the same as it was on the previous time period. Furthermore, when the key is constant, music is often composed of familiar chord progressions, such as the stabilizing I-V-I-V-I or the ubiquitous 12-bar blues progression I-IV-I-V-IV-I. Although chord progressions often have longer memory, as in the Rock example, a significant component of functional harmonic behavior is captured by transition tendencies: V tends to go to I, ii often goes to V, etc. Similarly, many of the tendencies for chord transitions are mirrored by analogous key modulations tendencies. For instance, as the chords I and V frequently appear side-by-side, modulations to neighboring keys in the circle of fifths are also common. (Although treated separately in our model, these phenomena are really not completely distinct.) It is these transition tendencies that can be represented by a Markov chain.

We of course do not observe the sequence of labels $X_1, \ldots, X_N$ directly, but rather our note data $y_1, \ldots, y_N$. The second assumption of the hidden Markov model (HMM) is that each data vector, y_n, is an observation of a random variable, Y_n, whose distribution depends only on the current label

$$p(y_n|x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) = p(y_n|x_n) \quad (3)$$

Essentially, this assumption says that every time we visit a state (harmonic label), the data are obtained by ''spitting out'' a collection of pitches from a distribution characteristic of the state. Again, this is certainly not ''correct,'' but the pitch data clearly does depend heavily on the harmonic label. Although including more structure (dependencies) may make the model more realistic, it only helps our particular cause when it improves the model's ability to discriminate between harmonic labels.

Hand-Tying of States

Our model is parameterized by the transition probabilities $p(x_{n+1}|x_n)$ and output probabilities $p(y_n|x_n)$. One of the greatest advantages of the HMM is that these parameters can be learned automatically from unlabeled data, e.g., generic MIDI files. However, at present the transition probabilities consist of (12 ×

$2 \times 7)^2 = 28224$ parameters—more than we can expect to train reliably with a modest data set. A similar problem exists with the output probabilities $p(y_n|x_n)$. We introduce some hand-crafted simplifying assumptions that lead to feasible training. As in the preceding, our research bias is for making our assumptions explicit, even when they seem imperfect.

Recall that each harmonic label, x, is a triple consisting of a tonic, mode, and chord, $x = (t, m, c)$. We model the transition probabilities $p(x'|x)$ as

$$\begin{aligned} p(x'|x) &= p(t', m', c'|t, m, c) \qquad (4a) \\ &= p(t', m'|t, m)p(c'|t', m', t, m, c) \\ &= p(t' - t, m'|m) \begin{cases} p(c'|c) & t' = t,\ m' = m \\ p(c') & \text{otherwise} \end{cases} \\ &\stackrel{\text{def}}{=} q_t(t' - t, m'|m) \begin{cases} q_c^2(c'|c) & t' = t,\ m' = m \\ q_c^1(c') & \text{otherwise} \end{cases} \qquad (4b) \end{aligned}$$

In Equation 4, we have assumed that the probability of the new key, (t', m'), given the current state (t, m, c), $p(t', m' \mid t, m, c)$, does not in fact depend on the current chord c. The left factor of Equation 4, $p(t' - t, m'|m)$, represents a translation invariance assumption about key modulations—the probability of modulating by some particular interval does not depend on the current tonic. As the authors do not have absolute pitch and hear only relative pitch movement, this and other pitch translation invariance assumptions seem unassailable. (The difference $t' - t$ is taken modulo 12.)

The right factor of Equation 4 is composed of two assumptions. The first (top) is that when the key is constant, the chord transitions do not depend on the current key. This is essentially another translation invariance assumption and seems to us undeniable provided we restrict our attention to either major or minor modes. The assumption goes a bit further and asserts that, for example, the probability of moving from I to V is the same in both major and minor modes. The second (bottom) assumption is that when we do move from one key to another, we choose the new chord at random without regard for the new or old keys. We doubt this particular assumption would hold up under empirical investigation but also doubt that a more nuanced modeling of this case will achieve significant improvements in recognition accuracy. These assumptions reduce the number of parameters necessary to represent $p(x'|x)$ to those involved in the distributions q_t, q_c^1, q_c^2: $12 \times 2 \times 2 + 7 \times 7 + 7 = 104$ parameters, further reduced by the constraint that each probability distribution must add to 1.

In modeling the output distributions $p(y|x)$, we have observed that, whereas expressive dissonances are a mainstay of musical surprise, surprise is almost by definition an exception to the norm; in particular, we anticipate that chord tones are more likely to occur on rhythmically strong beats than weak ones. Rather than trying to quantify such a notion directly, we simply allow the output distributions to depend on the known measure positions in a manner we will learn from data. Thus, we treat the measure positions as covariates in our model and condition on them as well as the chord label.

In particular, suppose that the pitch set $y = y^1, \ldots, y^K$ has an associated vector $r = r^1, \ldots, r^K$, where r^k labels the measure position occupied by y^k. In our experiments with music in 4/4 time, we took r^k to be 0 if y^k occupies the beginning of a measure, 1 if y^k begins on the second half note of the measure, 2 if y^k lies on the second or fourth quarter-note positions, etc., with a final category of 3 for "other." We then proceed to model

$$\begin{aligned} p(y|x, r) &= p(y^1, \ldots, y^K|x, r^1, \ldots, r^K) \qquad (5a) \\ &= \prod_{k=1}^{K} p(y^k|x, r^k) \\ &= \prod_{k=1}^{K} \frac{p(d(y^k, x)|r^k)}{V(d(y^k, x))} \qquad (5b) \\ &\stackrel{\text{def}}{=} \prod_{k=1}^{K} \frac{q_o(d(y^k, x)|r^k)}{V(d(y^k, x))} \end{aligned}$$

where

$$\begin{aligned} d(y^k, x) &= d(y^k, t, m, c) \\ &= \begin{cases} 1 & \gamma \text{ is root of chord } t, m, c \\ 2 & \gamma \text{ is third of chord } t, m, c \\ 3 & \gamma \text{ is fifth of chord } t, m, c \\ 4 & \gamma \text{ is scale } t, m \\ & \text{but not triad } t, m, c \\ 5 & \text{otherwise} \end{cases} \qquad (6) \end{aligned}$$

and $V(d)$ is the number of chromatic pitches falling into the dth category: $V(1) = V(2) = V(3) = 1$; $V(4) = 4$; and $V(5) = 5$.

Equation 5 states that given the harmonic label, x, the pitches $y^1, \ldots, y^K$ are random samples from their respective rhythm-conditional (r^k) distributions. Although we expect that this assumption will seem familiar to many, we believe it is among the most problematic: the order in which pitches appear clearly affects one's harmonic perception. We will discuss a possible variation on our model that does not make this assumption in a later section. The assumption does, however, lead to a significant reduction in model complexity. Equation 6 states that the probabilities of observing the categories chord root, chord third, chord fifth, non-triad scale tone, or non-scale tone are fixed and do not depend on the harmonic label. There are several chromatic pitches in the latter two categories, and our assumption is that within a category, the pitches will be equally likely. So, for instance, for notes at a given measure position, the probability of observing D in the IV chord of C major is the same as that of B-flat in the I chord of D minor. Additionally, the pitches C-sharp, D-sharp, F-sharp, A-flat, and B-flat all have the same probability in the key of C major, regardless of the particular chord; however, this probability depends on the measure position.

We denote these ''output'' probabilities by q_o where $q_o(d \mid r)$ is the probability of observing a pitch of category $d \in \{1, \ldots, 5\}$ for a note beginning at rhythmic position $r \in \{0, \ldots, 5\}$. These assumptions reduce the number of parameters in the representation of $p(y \mid x, r)$ to $5 \times 6 = 30$.

Training the Model

Our preference for the HMM, and, more generally, probabilistic graphical models, is partly due to the way the model parameters—the transition probabilities parameters, q_t, q_c^1, q_c^2, and the output distributions, q_o, can be trained from unlabeled examples. Because our model is based on rhythm as well as pitch, essentially any collection of MIDI files that explicitly represent both rhythm and pitch can be used for training. This is the case for most MIDI files that do not come from actual performances.

The essential idea of the training is as follows. If we had a collection of labeled data giving not just the pitch and rhythm information but the harmonic labels as well, $x_1, \ldots, x_N$, then the training process would be simple. For instance, $q_t(c' \mid c)$ could be estimated as the ratio of the number of times we observed the chords c, c' in sequence (with common key) to the total number of times we observed c (with the next chord in the same key). Similarly, $q_o(d \mid r)$ would be estimated as the ratio of the of number of times we observed a note of rhythm category r and pitch category d divided by the number of times we observed rhythm category r. (Note that the harmonic label must be known to compute $d = d(y^k, x)$.) In practice, we do not know the harmonic labels, but given a configuration of model parameters, we can estimate them. The idea of the forward-backward approach, or Baum-Welch algorithm, is to iteratively estimate the hidden labels and re-estimate the model parameters. It is well known that this is an example of the more general Expectation Maximization (EM) algorithm for maximum likelihood estimation of parameters in a mixture model (e.g., Rabiner 1993).

The harmonic labels $x_1, \ldots, x_N$ are estimated through the forward-backward iterations. We recursively define the forward probabilities, $\alpha_n(x_n)$, for $n = 1, \ldots, N$, $x_n \in L$ by

$$\begin{aligned} \alpha_1(x_1) &= p(x_1)p(y_1 \mid x_1) \\ \alpha_{n+1}(x_{n+1}) &= \sum_{x_n \in L} \alpha(x_n)p(x_{n+1} \mid x_n)p(y_{n+1} \mid x_{n+1}) \end{aligned} \tag{7}$$

and the backward probabilities, $\beta_n(x_n)$, for $n = 1, \ldots, N$, $x_n \in L$ by

$$\begin{aligned} \beta_N(x_N) &= 1 \\ \beta_{n-1}(x_{n-1}) &= \sum_{x_n \in L} \beta_n(x_n)p(x_n \mid x_{n-1})p(y_n \mid x_n) \end{aligned} \tag{8}$$

A standard argument shows that $\alpha_n(x_n) = p(x_n, y_1, \ldots, y_n)$, where the latter is viewed as a function of x_n with the y's held fixed. Similarly, $\beta_n(x_n) = p(x_n, y_{n+1}, \ldots, y_N)$. The α and β probabilities lead to label probabilities through

$$p(x_n|y_1, \ldots, y_N) = \frac{p(x_n, y_1, \ldots, y_n)p(y_{n+1}, \ldots, y_N|x_n)}{\sum_{x'_n} p(x'_n, y_1, \ldots, y_n)p(y_{n+1}, \ldots, y_N|x'_n)} = \frac{\alpha_n(x_n)\beta_n(x_n)}{\sum_{x'_n} \alpha_n(x'_n)\beta_n(x'_n)} \quad (9)$$

The probabilities $p(x_n \mid y_1, \ldots, y_N)$ function as surrogates for the true class labels. For instance, in estimating the output distributions if $p(x_n \mid y_1, \ldots, y_N) = 1/2$ for some particular state x_n, then y_n counts as 1/2 a sample from the output distribution for x_n. More precisely, we re-estimate q_o by

$$p(x_n, x_{n+1}|y_1, \ldots, y_N) = \frac{\alpha_n(x_n)\beta_{n+1}(x_{n+1})p(x_{n+1}|x_n)p(y_{n+1}|x_{n+1})}{\sum_{x'_n, x'_{n+1}} \alpha_n(x'_n)\beta_{n+1}(x'_{n+1})p(x'_{n+1}|x'_n)p(y_{n+1}|x'_{n+1})} \quad (10)$$

A similar argument leads to a re-estimate of the transition probabilities. For instance, we can compute the probabilities

$$q_c^2(c'|c) = \frac{\sum_{(t_n, m_n)=(t_{n+1}, m_{n+1}), c_n=c, c_{n+1}=c'} p(x_n, x_{n+1}|y_1 \ldots, y_N)}{\sum_{(t_n, m_n)=(t_{n+1}, m_{n+1}), c_n=c} p(x_n, x_{n+1}|y_1 \ldots, y_N)} \quad (11)$$

and re-estimate $q_c^2(c'|c)$ by

$$\hat{x} = \arg\max_x p(x|y) = \arg\max_x p(x, y) \quad (12)$$

where $x_n = (t_n, m_n, c_n)$ and $x_{n+1} = (t_{n+1}, m_{n+1}, c_{n+1})$.

We can also estimate q_c^1 and q_t in an analogous manner.

Experiments

We have performed a variety of training experiments, as discussed in the Training section of this article, involving approximately five short movements and requiring several minutes of computing on a 1 GHz Linux computer. We generally perform five iterations of the training algorithm and learn the output probabilities q_o, as well as the chord-transition probability matrix q_c^2. The remaining parameters of the transition probabilities, q_t and q_c^1, seem to require larger training sets to be reliably estimated. We have set these parameters by hand in the experiments here, but we plan on automatically learning them in the future. We bias the training toward a reasonable outcome by initializing our output probabilities to reflect our knowledge; if a particular chord and key are sounding, then the members of that chord should be the most likely pitches, the other notes in the key should be somewhat less likely, while the pitches not in the key are the least likely. No dependence on the rhythmic position is included in the initialization. The initialization of the functional chord transition probabilities does not seem to affect the result much, if at all.

Once the model is in place, we compute our harmonic parse, $\hat{x}$, as the most likely labeling given the data

$$\begin{aligned} P_n(x_n) &\stackrel{\text{def}}{=} \max_{x_1, \ldots, x_{n-1}} p(x_1, \ldots, x_n, y_1, \ldots, y_n) \\ &= \max_{x_{n-1}} \max_{x_1, \ldots, x_{n-2}} p(x_1, \ldots, x_{n-1}, y_1, \ldots, y_{n-1}) \\ &\quad p(x_n|x_{n-1})p(y_n|x_n) \\ &= \max_{x_{n-1}} P_{n-1}(x_{n-1})p(x_n|x_{n-1})p(y_n|x_n) \end{aligned} \quad (13)$$

where $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_N)$, $x = (x_1, \ldots, x_N)$, and $y = (y_1, \ldots, y_N)$. It is well known (e.g., Rabiner 1993) that the global maximum, $\hat{x}$, can be constructed with dynamic programming by letting $P_1(x_1) = p(x_1, y_1) = p(x_1)p(y_1 \mid x_1)$ and recursively computing

$$Q_n(x_n) \stackrel{\text{def}}{=} \arg\max_{x_{n-1}} P_{n-1}(x_{n-1})p(x_n|x_{n-1})p(y_n|x_n) \quad (14)$$

for $n = 2, \ldots, N$. From the definition of P_n, it follows that $\max_x p(x, y) = \max_{x_N} P_N(x_N)$. The actual sequence $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_N)$ attaining this maximum can be identified by performing a calculation parallel to P_n: we define

$$q_o(d|r) = \frac{\sum_{d(y_n^k, x_n)=d, r_n^k=r} p(x_n|y_1, \ldots, y_N)}{\sum_{r_n^k=r} p(x_n|y_1, \ldots, y_N)} \quad (15)$$

$n = 2, \ldots, N$. The Q_n functions lead to the optimal state sequence by $\hat{x}_n = Q_{n+1}(\hat{x}_{n+1})$ where $\hat{x}_N = \arg\max P_N(x_N)$. Although in many applications, the dynamic programming recursion is approximated

with a ''beam search,'' the size of our state space allows for full-fledged dynamic programming.

We have made several examples of our experiments available on the Web at fafner.math.umass.edu/ismir03. These include the first movement of Haydn's Piano Sonata No. 6; Chopin's ''Raindrop'' Prelude in D-Flat, Op. 28, no. 15; and the Prelude from Debussy's *Suite Bergamasque*. Our analysis is represented as a MIDI file of a mechanical piano performance with the series of chords produced by our algorithm superimposed as sustained harmonica chords. In addition, an obliging MIDI player will write out text messages giving the harmonic label as a (Roman numeral, tonic, mode) triple, aligned to highlight key changes. The messages are written as chord changes occur, essentially annotating the MIDI performance in real-time. All three examples mentioned above are in 4/4 time to facilitate a uniform definition of the rhythm variables expressing the ''strength'' of measure positions, r_n. In the Haydn and Chopin examples, we only allowed harmonic transitions to occur on two-beat boundaries. This was relaxed to one-beat boundaries in the Debussy example, which results in a somewhat over-analyzed labeling. As mentioned in the The Model section of this article, the choice of possible chords is somewhat arbitrary. In these experiments we have added the dominant 7th chord to the seven basic triads. Some basic extensions, such as fully diminished 7th chords and chords in minor mode built on the flat seventh scale degree, are needed in these experiments; however, more exotic additions such as augmented sixth chords and Neapolitan chords are possible.

These examples are representative of the more successful applications of our program. However, they are comparable to the majority of cases we have examined in which our program produces a plausible interpretation. (The less-successful results seem primarily to be compositions with very sparse textures.) We believe that these results are quite promising, especially taking into account the simplicity of our approach. In addition, we feel there is significant potential for improvement with more careful attention to modeling subtleties and larger-scale training experiments. One of our future goals is to have a program on the Web that will automatically analyze a visitor's submission and produce an annotated MIDI file as described above.

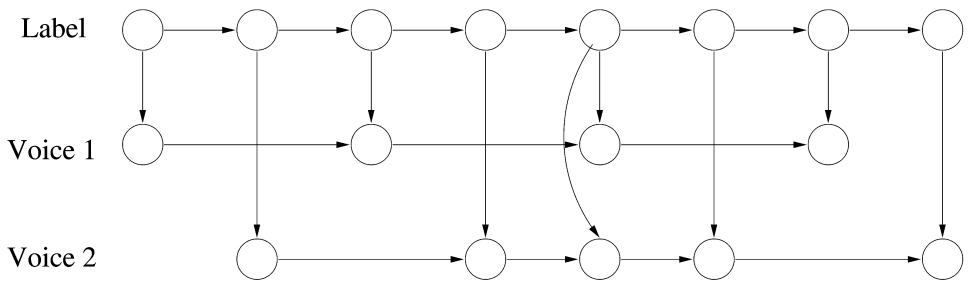
At this point, we do not offer any objective measure of success, such as error rates or comparisons of our results. This is due in part to the difficulty in defining and obtaining ''correct'' harmonic analyses. However, we also believe that rather straightforward continued efforts may lead to significant improvements and that such evaluation will be more appropriate at a later stage.

Extending the Model

In our view, the most troubling modeling assumption we make is the conditional independence of pitches: in essence, the collection of pitches associated with a chord is a random sample from some distribution. This assumption disregards the way music is usually composed of independent parts or voices that obey an internal logic, such as a preference for scales and arpeggios. Given the often-unvoiced nature of MIDI data and our current focus on piano music, we have begun with a simple model that does not require voicing information. However, we now propose a model that regards the data as a collection of voices where the evolution of each voice is conditionally independent of the others given the harmonic state.

Automatically partitioning MIDI data into voices is no doubt a challenging problem if one requires the voicing to be identical to the true voicing, if one exists, or to the ''ground truth'' supplied by a musician. But it is rather simple to create an algorithm that performs reasonably well. We use a simple dynamic programming algorithm maximizing a function measuring the plausibility of a voice partition. Other possibilities exist, such as techniques described by Kilian and Hoos (2002). We assume here that we begin with voiced data, either from an ''official'' or algorithmically composed source. In particular, we begin with a collection of monophonic overlapping voices with no assumption about the number of voices that overlap at any particular time or the range of pitches associated with a particular voice.

Figure 1. The graphical representation of a model containing two conditionally (on X*) independent voices.*



Suppose that $y_1, \ldots, y_N$ is a sequence of pitch classes, represented as numbers in 0,. . .,11, corresponding to a single voice. Let $X_1,\ldots,X_N$ be the sequence of (key, chord) variables, assumed to be a Markov chain as before. We continue to assume that y_n is the observation of a random variable Y_n, but unlike before, we now assume that the distribution of Y_n depends on X_n and Y_{n-1}, rather than just X_n. Figure 1 shows a graphical representation of such a model containing two conditionally (on X) independent voices.

Such a model, suitably trained, would understand a voice's preference for scales within the key, arpeggios within the (key, chord) pair, and tendencies regarding the resolution of non-chord tones. These preferences should assist in distinguishing between various chord hypotheses given input data.

Although this model is not an HMM, it has a linear structure amenable to an analogous training algorithm as well as the identification of the most-likely state sequence. Thus, the model is computationally tractable with cost comparable to that of the HMM.

Acknowledgments

This work is supported by NSF ITR grant IIS-0113496.

References

Barthelemy, J., and A. Bonardi. 2001. ''Figured Bass and Tonality Recognition.'' In *Proceedings of the Second International Conference on Music Information Retrieval*. Bloomington, Indiana: Indiana University, pp. 129–135.

Chew, E., and Y. Chen. 2003. ''Determining Context-Defining Windows: Pitch Spelling Using the Spiral Array.'' In *Proceedings of the Fourth International Conference on Music Information Retrieval*. Baltimore, Maryland: Johns Hopkins University, pp. 223–224.

Kilian, J., and H. Hoos. 2002. ''Voice Separation: A Local Optimisation Approach.'' In *Proceedings of the Third International Conference on Music Information Retrieval*. Paris: IRCAM, pp. 39–46.

Krumhansl, C. 1990. *Cognitive Foundations of Musical Pitch*. Oxford: Oxford University Press.

Pickens, J., et al. 2002. ''Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach.'' In *Proceedings of the Third International Conference on Music Information Retrieval*. Paris: IRCAM, pp. 140–149.

Pardo, B. 2002. ''Algorithms for Chordal Analysis.'' *Computer Music Journal* 26(2):27–49.

Rabiner, L. 1993. ''A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition.'' *Proceedings of the IEEE* 77:257–286.

Temperley, D., and D. Sleator. 1999. ''Modeling Meter and Harmony: A Preference Rule Approach.'' *Computer Music Journal* 15(1):10–27.