Following a musical performance from a partially specified score.

Bryan Pardo and William P. Birmingham

Artificial Intelligence Laboratory
Electrical Engineering and Computer Science Dept. and School of Music
The University of Michigan
110 ATL building, 1001 Beal Drive
Ann Arbor, MI 48109
bryanp@umich.edu, wpb@eecs.umich.edu

Abstract

This paper describes a system that maps a musical performance, recorded as MIDI, onto a partially specified score (lead sheet). Our system breaks the performance into appropriate segments for hypothesized chords by representing it as a directed acyclic graph (DAG) whose edges represent time-intervals of the music. The highest-reward path through the DAG corresponds to the best segmentation of the performance. During segmentation, the name of the best matching chord for each segment label is given to the segment. The sequence of chord names generated by the performance analysis is then aligned against the sequence of chord names in the lead sheet using an alignment technique drawn from gene-sequence analysis. Scores for the alignment are determined by estimating the probability of a random match between chord sequences compared to the probability of the match. Probabilities are calculated through the analysis of chord frequencies on a corpus of tonal music drawn from a standard music theory text.

1 Introduction

Automated musical accompaniment that reacts naturally to the human performer is a long-standing goal of computer-music research. Algorithms that match a written score to a human performance are essential for an automated accompanist that reacts appropriately during performance. Systems that perform this function are called score matchers.

Score-matching research (Dannenberg 1984; Dannenberg 1988; Puckette and Lippe 1992; Vantomme 1995; Heijink, Desain et al. 2000) has mostly concentrated on matching a score that very closely specifies the pitch and ordering of every note in a piece of music to the performance of a musician attempting to recreate the score as exactly as possible. This results in approaches whose goal is producing one-to-one mapping between score events and performance events.

Many genres of folk and popular music have partially specified scores (lead sheets) that notate only main melodic notes, basic harmonies and overall structure, supplying far less musical information than a classical score. Musicians necessarily improvise melodic embellishments, the figuration of the harmonic accompaniment and even vary the overall form. This presents a problem for the traditional score-matching approach: the score events will be far fewer, and perhaps significantly different, than the performance events. Therefore, a one-to-one mapping is not possible.

This paper describes a system that maps a MIDI performance of a piece onto a lead sheet. We induce a metric reduction from the performance, removing structurally unimportant notes, leaving a rhythmically simplified musical surface that emphasizes chord tones. This structure is automatically annotated with the associated chord names. These chord labels are aligned (matched) to the chords in the lead sheet using a method borrowed from research in biological-sequence alignment.

2 Finding and Using the Metric Reduction

Our approach to metric reduction is based on automated chordal analysis. Our metric-reduction system has a vocabulary of six chord qualities (major, minor, dominant 7, diminished, half-diminished, and fully diminished). Given 12 possible roots for each quality, this gives a vocabulary of 72 chord *templates*.

During processing, the system constructs a directed acyclic graph (DAG) whose edges represent time-intervals of the music. The reward of an edge in the DAG corresponds to how well the notes in the corresponding segment match those in the closest chord template available to the system. The highest-reward path in the DAG corresponds to the best segmentation of the performance, which can be computed in linear time with respect to the number of chords (Pardo and Birmingham 2001).

Once a performance is segmented, the best matching chord template for each segment label is used to filter the performance for that segment. Notes matching a chord element are considered harmonic. Harmonic notes, when extended to the length of each segment, approximate a metric reduction of the music. Figure 1 contains a performance, a metric reduction of the performance generated by our system, and the lead sheet upon which the performance was based.

Figure 1 shows that the metric reduction, while closer to the lead sheet than the performance, has much detail not found in the lead sheet, leaving open the question of what should be matched between lead sheet and metric reduction. In this paper, we focus on the matching the sequence of chord labels in the metric reduction to the sequence of chords in the lead sheet.

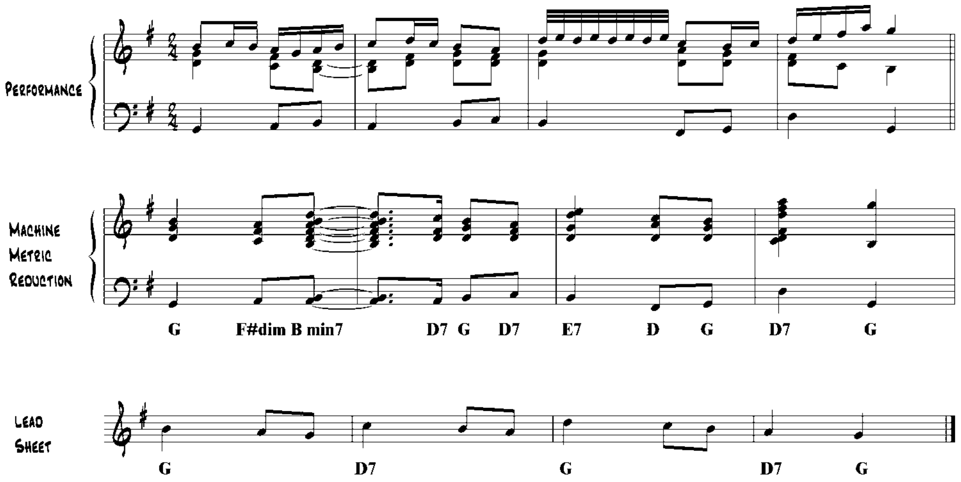

Figure 1: Lead sheet, performance, and metric reduction

2.1 Chord Sequence Alignment

Once the problem has been cast as finding the best alignment between two sequences drawn from the same alphabet of symbols (chord names), we can draw upon the extensive work done in the field of biological sequence analysis. Dynamic-programming-based approaches have been used with success for over 30 years (Needleman and Wunsch 1970) to align gene sequences based on a common ancestor. We have based our chord-aligner on a dynamic-programming algorithm described by Gotoh (Gotoh 1982), which is designed to find the globally best alignment among sequences, given some match scoring model.

2.2 Pair Match Scores

In comparing sequences, we look for evidence that each element in chord sequence *A* is related to some element in sequence *B* and then estimate the probability that the sequences are related rather than a chance pair.

Here is a summary of our notation. Let there be a pair of chord sequences, *A* and *B*, of length |*A*| and |*B*|, respectively. Let a_i be the *i*th element of *A* and b_j be the *j*th element of *B.* Let the elements of each sequence be drawn from the same alphabet. In our case, the alphabet contains the 72 chord names used for metric reduction.

Assume each element, *e*, of the alphabet occurs independently with some known prior probability, *P*(*e*). Then, the probability of a random joint occurrence (*R*) of two sequences *A* and *B* is the product of the prior probabilities of the sequence elements.

$$P(A,B\,|\,R)=\prod_i P(a_i)\prod_j P(b_j) \tag{2.2.1}$$

If, indeed, the two sequences are a match (*M*) of some kind, then aligned pairs of chords occur with a joint probability that indicates how likely it is that the elements are related.

$$P(A,B\,|\,M)=\prod_i P(a_i,b_i\,|\,M) \tag{2.2.2}$$

The *odds ratio* is the ratio between these two equations.

$$\frac{P(A,B\,|\,M)}{P(A,B\,|\,R)}=\frac{\prod_i P(a_i,b_i\,|\,M)}{\prod_i P(a_i)\prod_j P(b_j)} \quad \text{if i = j, then} \quad =\frac{\prod_i P(a_i,b_i\,|\,M)}{\prod_i P(a_i)P(b_j)} \tag{2.2.3}$$

In order to have a simple additive model, we take the log of this ratio,

$$S=\sum_i s(a_i,b_i) \tag{2.2.4}$$

where

$$s(a,b)=\log\left(\frac{P(a,b\,|\,M)}{P(a)P(b)}\right) \tag{2.2.5}$$

is the log of the likelihood of the pair (*a, b*) occurring because they are related as opposed to occurring randomly. One nice feature of using the log ratio is that when the probability of a match is below that of a random co-occurrence, the value is negative. Similarly, the value is positive when a match is more likely than random chance.

The values for $s(a,b)$ can be placed in a *score matrix*, where the element i,j represents the score for the log likelihood match of the ith chord in the alphabet to the jth chord in the alphabet.

In order to fill the score matrix, one must find the prior probability for each of the six chord qualities in all 12 keys. A natural way to find the prior probability of each chord is to take the observational frequencies of chord occurrence from a representative corpus of music.

The *Kostka-Payne Corpus* is a set of 46 excerpts of tonal music from various periods. These excerpts have been annotated with chord labels. The occurrence count for each of the six chord qualities used in our system was tabulated from these annotations. The prior probability for each of the 12 chords of a particular quality was then taken to be the probability of the chord quality divided by 12. For example, 46.3% of the chord annotations in the corpus were major triads. Thus, P("C major") = 0.436/12 = 0.03633.

Each $s(a,b)$ also requires an estimate of $P(a,b \mid M)$. To fill in the full score matrix, this must be done for all chord pairs (a,b). This probability can be thought of as the probability that a segment of music derived from chord a is assigned a label b by our metric reduction system. To estimate this probability, we again turned to the Kostka-Payne corpus. Each excerpt in the corpus was labeled with chord names by our metric reduction system and the resulting labels were compared to those given in the annotated corpus. From this, a table was compiled where the element i,j represents the number of times our system labeled a segment with the ith chord in the alphabet and the corpus labeled it with the jth chord in the alphabet. The final score matrix was then derived from the prior probabilities and this table, in accordance with Equation 2.2.5.

2.3 The Alignment Algorithm

We use a dynamic-programming algorithm introduced by Gotoh (1982), as described in (Durbin, Eddy, et al. 1998) to find an optimal global alignment between two sequences A and B, allowing gaps between the sequences. This is done by constructing a matrix F, where $F(i,j)$ is the score of the best alignment between the initial segment a_1 through a_i of A and the initial segment b_1 through b_j of B.

The process is initialized by setting $F(0,0) = 0$. Thereafter, the elements of the matrix are filled in using Equation 2.3.1, where d is the gap penalty assigned to skipping an element of either sequence to perform the alignment. The top line in Equation 2.3.1 gives the reward assigned in the score matrix for calling (a_i, b_j) a match. The middle line calculates the penalty for skipping b_j and the lowest finds the penalty for skipping a_i.

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(a_i, b_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases} \qquad (2.3.1)$$

Note that, in the case of score following, the lead sheet chord sequence is known in advance. Since each cell in the array is filled in using the values of the items above and to the left of it, each row may be filled in using only the knowledge of the prior row. This lets the system fill in this table as the metric reduction is being created, with a fixed number of steps per row. The time to fill in the table is linear with respect to the length of the performance, so this method presents no theoretical limit on the ability of a system to score-match in real-time.

As *F* is filled in, another table must be kept, keeping track of the parent used to fill in each cell *F*(*i*,*j*). In many cases, it may be that more than one of the parents from Equation 2.3.1 gives the maximum value. In this case, all parents are noted. Once this is done, the best-scoring alignment may be read by starting at the final cell in the matrix and tracing backwards through the series of parents used to generate the score.

Table 1 shows the alignment score matrix, *F*, for the chord sequence from the lead sheet (columns) and metric reduction (rows) in Figure 1, given a gap penalty, *d* = 1, and the probabilities calculated from the Kostka-Payne corpus. All maximal scoring alignments are shown by paths of arrows through the table. In this table, a vertical arrow indicates a skip of one chord in the metric reduction sequence, a horizontal arrow is a skip of a chord in the lead sheet, and a diagonal arrow indicates a match between the two.

Table 1: Alignment scores for sequences from Figure 1, given *d* = 1. Lead sheet chord names are on the horizontal axis. Metric reduction chord names are on the vertical axis.

		G major	**D dom7**	**G major**	**D dom7**	**G major**
	0	-1	-2	-3	-4	-5
G major	-1	6.24	5.24	4.24	3.24	2.24
F# dim	-2	5.24	12.13	11.13	10.13	9.13
B minor	-3	4.24	11.13	14.84	14.24	13.24
D dom7	-4	3.24	11.85	13.84	22.45	21.45
G major	-5	2.24	10.85	18.08	21.45	28.68
D dom7	-6	1.24	9.85	17.08	25.7	27.68
E dom7	-7	0.24	8.85	16.08	24.7	27.83
D major	-8	-0.76	7.85	15.08	23.7	26.83
G major	-9	-1.76	6.85	14.08	22.7	29.93
D dom7	-10	-2.76	5.85	13.08	21.7	28.93
G major	-11	-3.76	4.85	12.08	20.7	27.93

Note that, while not shown in this example, it is quite possible for the alignment matrix to show matches between chords of different name if they have a high probability of matching due to chord similarity and the gap penalty is large in comparison with the match scores.

For the purpose of score following, Table 1 is used to determine the most probable place(s) in the lead sheet given the current chord reported in the metric reduction. For example, the ninth chord in the metric reduction is "G major." The arrows in the table show that there are two best alignments for this chord. The first is to align it with the third chord in the lead sheet. The other is to align it with the final chord in the lead sheet.

3 Conclusions

We have described a system that maps a MIDI performance of a piece onto a partially specified score (lead sheet) for that piece. This is done by mapping the sequence of chords induced from an automatically generated metric reduction of the performance onto the sequence of chords in the lead sheet. Both the metric reduction and the alignment matrix can be generated in linear time with respect to the length of the performance. Thus, this approach is a good one for a real-time matcher to a partially specified score.

Future work in this area includes a more realistic gap penalty that reflects chord timing rather than simply order, a higher-level process to disambiguate between equal-scoring alignments and a real-time implementation of a score-matching system based on these techniques.

4 References

Dannenberg, R. (1984). "An On-Line Algorithm for Real-Time Accompaniment." International Computer Music Conference, International Computer Music Association.

Dannenberg, R. B. (1988). "New Techniques for Enhanced Quality of Computer Accompaniment." Proceedings of the International Computer Music Conference, Ann Arbor, MI.

Durbin, R., S. Eddy, A. Krogh, G. Mitchison (1998). Biological Sequence Analysis, Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge, U.K.

Gotoh, 0. (1982). “An improved algorithm for matching biological sequences” Journal of Molecular Biology 162: 705-708.

Heijink, H., P. Desain, et al. (2000). “Make Me a Match: An Evaluation of Different Approaches to Score-Performance Matching.” Computer Music Journal 24(1): 43-46.

Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins" Journal of Molecular Biology 48:443-453

Pardo, B. and W. Birmingham (2001). "The Chordal Analysis of Tonal Music." Technical report CSE-TR-439-01, The University of Michigan, Dept. of Electrical Engineering and Computer Science, Ann Arbor, MI

Puckette, M. and C. Lippe (1992). Score Following In Practice. International Computer Music Conference, International Computer Music Association.

Vantomme, J. (1995). “Score Following by Temporal Pattern.” Computer Music Journal 19(3): 50-59.