Automatic Chord Recognition from Audio Using a Supervised HMM Trained with Audio-from-Symbolic Data*

Kyogu Lee
Center for Computer Research in Music and Acoustics
Music Department, Stanford University
kglee@ccrma.stanford.edu

Malcolm Slaney
Yahoo! Research
Sunnyvale, CA94089
malcolm@ieee.org

ABSTRACT

A novel approach for obtaining labeled training data is presented to directly estimate the model parameters in a supervised learning algorithm for automatic chord recognition from the raw audio. To this end, harmonic analysis is first performed on symbolic data to generate label files. In parallel, we synthesize audio data from the same symbolic data, which are then provided to a machine learning algorithm along with label files to estimate model parameters. Experimental results show higher performance in frame-level chord recognition than the previous approaches.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms

Keywords

Chord recognition, hidden Markov model, supervised learning, MIDI

1. INTRODUCTION

A musical chord is a set of simultaneous tones. Succession of chords over time, or chord progression, form the core of harmony in a piece of music. Hence analyzing the overall harmonic structure of a musical piece often starts with labeling every chord. Automatic chord labeling is very useful for those who want to do harmonic analysis of music. Once the harmonic content of a piece is known, a sequence of chords can be used for further higher-level structural analysis where phrases or forms can be defined. Chord sequences with timing information of chord boundaries are also a very compact and robust mid-level representation of musical signals for such applications as music identification, music segmentation, music similarity finding, and audio thumbnailing. For these reasons and others, automatic chord recognition has recently attracted a number of researchers in the Music Information Retrieval field.

Hidden Markov models (HMMs) are very successful for speech recognition, and gigantic databases with labels accumulated over decades play an important role in estimating the model parameters appropriately. However, there is no such database available for music. Furthermore, the acoustical variance in a piece of music is far greater than that in speech in terms of its frequency range, instrumentation, dynamics, and/or duration, and thus a lot more data is needed to train the models for generalization.

Hand-labeling the chord boundaries in a number of recordings is not only an extremely time consuming and tedious task but also is subject to errors made by humans. In this paper, we propose a method of automating this daunting task to provide the machine learning models with labeled training data. To this end, we use symbolic data such as MIDI data to generate chord names and their boundaries as well as to create audio. Audio and chord boundary information generated this way are in perfect alignment, and we can use them to directly estimate the model parameters.

There are several advantages to this approach. First, we do not need to manually annotate chord boundaries with chord names to obtain training data. Second, we can generate as much data as needed with the same symbolic files but different musical attributes by just changing instrumentation, tempo, or dynamics when synthesizing audio. This helps avoid overfitting the models to a specific type of music. Third, sufficient training data enable us to include more chord types such as 7th, augmented, or diminished.

This paper continues with a review of related work in Section 2; in Section 3, we describe how we extract the feature vectors, and explain the model and the method of obtaining the labeled training data; in Section 4, we present empirical results with discussions, and draw conclusions followed by directions for future work in Section 5.

2. RELATED WORK

Sheh and Ellis proposed a statistical learning method for chord segmentation and recognition [11]. They used the hidden Markov models (HMMs) trained by the Expectation Maximization (EM) algorithm, and treated the chord labels

*This paper is based on the previous work by the same authors [8].

AMCMM'06, October 27, 2006, Santa Barbara, California, USA.

as hidden values within the EM framework. In training the models, they used only the chord sequence as an input to the models, and applied the forward-backward or Baum-Welch algorithm to estimate the model parameters. The frame accuracy in percent they obtained was about 76% for segmentation and about 22% for recognition, respectively. The poor performance for recognition may be due to insufficient training data compared with a large set of classes (20 songs for 147 chord types). It is also possible that the flat-start initialization of training data yields incorrect chord boundaries resulting in poor parameter estimates.

Bello and Pickens also used HMMs with the EM algorithm [1]. What was novel in their approach was that they incorporated musical knowledge into the models by defining a state transition matrix based on the key distance in a circle of fifths, and avoided random initialization of a mean vector and a covariance matrix of observation distribution. In addition, in training the model's parameter, they selectively updated the parameters of interest on the assumption that a chord template or distribution is almost universal, thus disallowing adjustment of distribution parameters. The accuracy thus obtained was about 75% using beat-synchronous segmentation with a smaller set of chord types (24 major/minor triads only). In particular, they argued that the accuracy increased by as much as 32% when the adjustment of the observation distribution parameters is prohibited.

The present paper expands our previous work on chord recognition [8]. It is based on the work of Sheh and Ellis or Bello and Pickens in that the states in the HMM represent chord types, and we try to find the optimal path, *i.e.*, chord sequence in a maximum-likelihood sense. The most prominent difference in our approach is, however, that we use labeled training data by which model parameters can be directly estimated. Furthermore, we propose a method to automatically obtain the labeled training data, removing the problematic and time consuming task of manual annotation of precise chord boundaries. In this paper, we build two separate HMMs from two training data sets of different instrumentation, and investigate how each model performs when various types of input are given.

3. SYSTEM

Our system starts with extracting suitable feature vectors from the raw audio. Like most chord recognition systems, a chroma vector or a PCP vector is used as the feature vector.

3.1 Chroma Features

A chromagram or a Pitch Class Profile (PCP) is the choice of the feature set in automatic chord recognition or key extraction since introduced by Fujishima [3]. Perception of musical pitch involves two dimensions – *height* and *chroma*. Pitch height moves vertically in octaves telling which octave a note belongs to. On the other hand, chroma tells where it stands in relation to others within an octave. A chromagram or a pitch class profile is a 12-dimensional vector representation of a chroma, which represents the relative intensity in each of twelve semitones in a chromatic scale. Since a chord is composed of a set of tones, and its label is only determined by the position of those tones in a chroma, regardless of their heights, chroma vectors appear to be an ideal feature to represent a musical chord or a musical key.

Fujishima developed a realtime chord recognition system, where he derived a 12-dimensional pitch class profile from the DFT of the audio signal, and performed pattern matching using the binary chord type templates [3]. Gomez and Herrera proposed a system that automatically extracts from audio recordings tonal metadata such as chord, key, scale and cadence information [4]. They used as the feature vector, a Harmonic Pitch Class Profile (HPCP), which is based on Fujishima's PCP, and correlated it with a chord or key model adapted from Krumhansl's cognitive study [7]. Similarly, Pauws used the maximum-key profile correlation algorithm to extract key from the raw audio data, where he averaged the chromagram features over variable-length fragments at various locations, and correlate them with the 24 major/minor key profile vectors derived by Krumhansl and Kessler [9]. Harte and Sandler used a 36-bin chromagram to find the tuning value of the input audio using the distribution of peak positions, and then derived a 12-bin, semitone-quantized chromagram to be correlated with the binary chord templates [5].

There are some variations when computing a 12-bin chromagram, but it usually follows the following steps. First, the DFT of the input signal $X(k)$ is computed, and the constant-Q transform X_{CQ} is calculated from $X(k)$, using a logarithmically spaced frequencies to reflect the frequency resolution of the human ear [2]. The frequency resolution of the constant-Q transform follows that of the equal-tempered scale, which is also logarithmically based, and the kth spectral component is defined as

$$f_k = (2^{1/B})^k f_{min}, \tag{1}$$

where f_k varies from f_{min} to an upper frequency, both of which are set by the user, and B is the number of bins in an octave in the constant Q transform. Once $X_{CQ}(k)$ is computed, a chromagram vector CH can be easily obtained as:

$$CH(b) = \sum_{m=0}^{M-1} \left| X_{CQ}(b + mB) \right|, \tag{2}$$

where $b = 1, 2, \cdots, B$ is the chromagram bin index, and M is the number of octaves spanned in the constant Q spectrum. For chord recognition, only $B = 12$ is needed, but $B = 24$ or $B = 36$ is also used for fine tuning.

In our system, we used $f_{min} = 48$ Hz, and $f_{max} = 5,250$ Hz. For the number of bins in an octave, we used $B = 12$ for MIDI-synthesized audio, and $B = 36$ for real recordings to obtain 12-bin Quantized chromagram proposed by Harte and Sanlder [5], which compensates a possible mistuning present in the recordings by reallocating the peaks based on the peak distribution.

3.2 Hidden Markov Model

A hidden Markov model [10] is an extension of a discrete Markov model, in which the states are *hidden* in the sense that an underlying stochastic process is not directly observable, but can only be observed through another set of stochastic processes.

We recognize chords using a 36-state HMM. Each state represents a single chord, and the observation distribution is modeled by a single multivariate Gaussian in 12 dimensions defined by its mean vector μ_i and covariance matrix Σ_i, where i denotes ith state. We assume the features are

uncorrelated with each other, and thus use diagonal covariance matrix. State transitions obey the first-order Markov property; *i.e.*, the future is independent of the past given the present state. In addition, we use an ergodic model since we allow every possible transition from chord to chord, and yet the transition probabilities are learned.

Once the model parameters – initial state probabilities, state transition probabilities, and mean vector and covariance matrix for each state – are learned, the Viterbi algorithm is applied to the model to find the optimal path, *i.e.*, chord sequence, in a maximum likelihood sense given an input signal.

In our model, we have defined 36 classes or chord types according to their sonorities only – major, minor, and diminished chords for each pitch class. We grouped triads and seventh chords with the same root and sonority into the same category. For instance, we treated E minor triad and E minor seventh chord as just E minor chord without differentiating the triad and the seventh. Augmented chords were not considered because they scarcely appear in Western tonal music. We found this class size appropriate in a sense that it lies between overfitting and oversimplification.

3.3 Harmonic Analysis on Symbolic Data

In order to train a supervised model, we need label files which must contain annotated chord boundaries as well as chord names. To automate this laborious process, we use symbolic data to generate label files as well as audio data. To this end, we first convert a symbolic file to a format which can be used as an input to a chord analysis tool. Chord analyzer then performs harmonic analysis and outputs a file with root information and note names from which complete chord information (*i.e.*, root and its sonority – major, minor, or diminished triad/seventh) is extracted. Sequence of chords are used as ground-truth or labels when training the HMM. In parallel, we use the same symbolic files to generate audio files using a sample-based synthesizer. Audio data generated this way are in perfect sync with chord label files obtained above, and are enharmonically rich as in real acoustic recordings because audio samples in a synthesis engine contain the upper harmonics as well. Figure 1 illustrates the overview of the system.

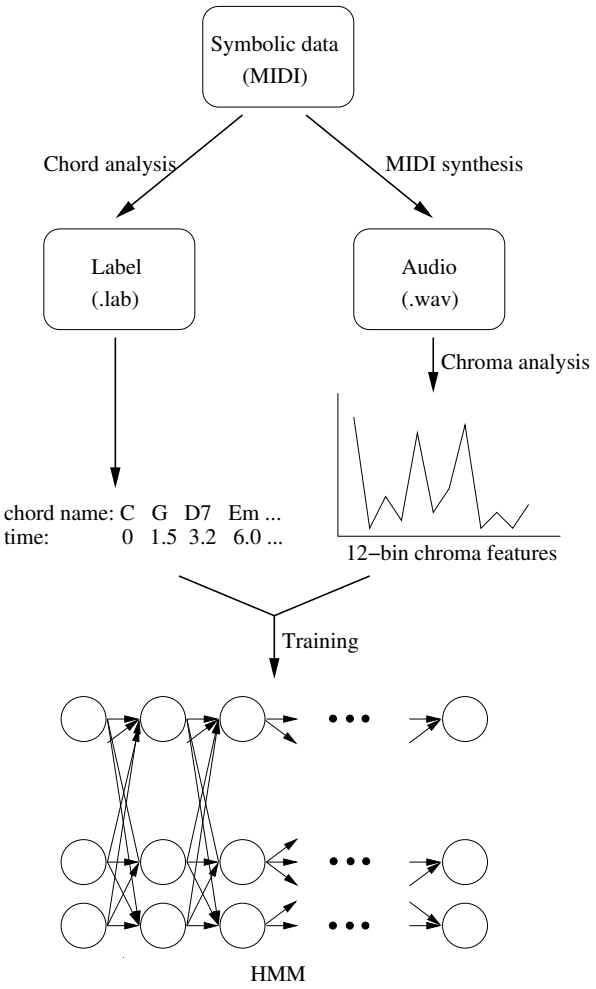

Figure 1: Overview of the system.

4. IMPLEMENTATION AND EXPERIMENTS

As shown in Figure 1, our system for generating labeled training data has two main blocks running in parallel. First, harmonic analysis is performed on symbolic data. We used symbolic files in Humdrum data format. Humdrum is a general-purpose software system intended to help music researchers encode, manipulate, and output a wide variety of musically-pertinent representations.[1] For harmonic analysis, we used the Melisma Music Analyzer developed by Sleator and Temperley by the authors.[2] The Melisma Music Analyzer takes a piece of music represented by an event list, and extracts musical information from it such as meter, phrase structure, harmony, pitch-spelling, and key. By combining harmony and key information extracted by the analysis program, a complete Roman-numeral analysis is performed, from which we can generate label files with sequence of chord names.

The analysis program was tested on a corpus of excerpts and the 48 fugue subjects from the *Well-Tempered Clavier*, and the harmony analysis and the key extraction yield the accuracy of 83.7% and 87.4%, respectively [12].

In the feature extraction block in our system, MIDI files are synthesized using Timidity++. Timidity++ is a free software synthesizer, and converts MIDI files into audio files in a WAVE format.[3] It uses a sample-based synthesis technique to generate enharmonically rich audio as in real recordings. The raw audio is downsampled to 11025 Hz, and 12-bin chroma features are extracted from it with the frame size of 8192 samples and the hop size of 2048 samples. The chroma vectors are then used as input to the HMM along with the label files obtained above.

To examine the model's dependence on the training data, we chose two different training data sets and obtained two model parameters on each data set. For the first model, we used as a training data set 81 files of solo piano music by J. S. Bach, Beethoven, and Mozart in a Humdrum data format at the Center for Computer Assisted Research in

[1] http://dactyl.som.ohio-state.edu/Humdrum/
[2] http://www.link.cs.cmu.edu/music-analysis/
[3] http://timidity.sourceforge.net/

the Humanities at Stanford University.[4] We used 196 files of string quartet music by Beethoven, Haydn, and Mozart to estimate the second set of parameters. These files were converted to a format which can be used in the Melisma Music Analyzer as well as to a MIDI format using the tools developed by Craig Sapp.[5] The audio data synthesized from these MIDI files for the first and the second model is about 4 hours long or 76,500 frames, and 12 hours long or 233,500 frames, respectively (16 hours or 310,000 frames total).

Figure 2 shows a transition probabilities matrix and transition probabilities for C major chord estimated from the training data set. It can be observed that the transition matrix is strongly diagonal since chord duration is usually longer than the frame length, and thus the state does not change for several frames, which makes a transition probability to itself highest.

As illustrated in Figure 3, however, chord progression based on music theory can also be found in transition probabilities, for example, in the case of C major chord. As mentioned, it has the largest probability of staying within the same state, *i.e.*, within C major chord, because of faster frame rate than the rate of chord changes, but has comparably higher probabilities for making a transition to specific chords like F major, G major, or F minor chord than to others. F major and G major have subdominant-tonic and dominant-tonic relationships with C major, respectively, and transitions between them happen very often in Western tonal music. C major chord is also a dominant chord of F minor, and therefore a C major to F minor transition is frequent as well. This tonic-dominant relationship can also be observed in Figure 2 as off-diagonal lines with 4 and 5 semitone offsets with respect to their tonics.

Figure 4 exemplifies the observation distribution parameters estimated from each training data set for C major chord. On the left is the mean chroma vector for C major chord for each model. It is obvious that they both have three largest peaks at chord tones or at C, E, and G, as expected. In addition, we can also see relatively large peaks at D, A#, and B, which come from the third harmonics of chord tones and/or from the seventh chords. One noticeable thing is a relatively high peak at the root note or at C in the bottom figure for HMM B, which was obtained from the training data composed of 196 string quartets. This can be explained by the fact that there are four instruments – violin I/II, viola, and cello – in a string quartet, and there is a very high chance of more than one instrument playing the root note simultaneously, which is the most important note in a theory of harmony, resulting in stronger spectral energy at the root note. Diagonal vectors of covariance matrices for C major chord is also consistent with what is expected from the music theoretical knowledge. Chord tones or C, E, and G are strongly correlated with themselves whereas very low correlation was found with D#, F#, or G#.

4.1 Empirical Results

We tested our models on the selected corpus of excerpts from the Kostka and Payne's book [6]. The book not only includes harmonic analyses of the excerpts done by the authors, but also is accompanied by corresponding audio files which were recorded using real acoustic instruments. The test set consists of 10 short excerpts – 5 piano solos and 5 string quartets. None of the test set was included in a training data set. Table 1 describes information on test material in more detail. Test data first goes through the chroma analysis which outputs 12-bin quantized chroma feature vectors. These feature vectors are then fed into the trained HMMs. Recognition is accomplished as the Viterbi algorithm finds the optimal path given the model parameters and the input observation vectors. We compared the output of the model, which is a sequence of frame-level chord names, with the hand-marked ground-truth to make scores for frame rate accuracy.

In computing scores, we only counted exact matches as correct recognition. We tolerated the errors at the chord boundaries by having some time margins of a few frames around the boundaries. This assumption is fair since the ground-truth was generated by human by listening to a piece, which can't be razor sharp.

Since we have two separate parameter sets trained on two different training data sets, we tested each test data set for each parameter set. In addition, we estimated another set of parameters using all the training data, *i.e.*, by combining both piano solo and string quartets together, which amount to 277 audio files, 16 hours of audio, or 310,000 frames. Therefore, we have six categories of test results for two test data sets and three parameter sets.

The recognition results from one example is shown in Figure 5. The test material was the fourth piano example of Beethoven's Piano Sonata Op. 14, No. 2, II. 12-bin chromagram is shown at the top, and the recognition for each model is displayed below it.

As can be seen in Figure 5, all three parameter sets successfully identify chord types as well as their boundaries. It is interesting to note that the performance is worst for piano parameters even though the test material was of the same kind. This may suggest that the model performance is more dependent on the amount of training data than its specificity. This observation is generally true for all test data except for a few exceptions.

Table 2 shows frame-rate recognition results for all six possible test data – parameter pairs. The total recognition rate was highest for the combined parameters, followed by the string quartet parameters and the piano parameters for both test data sets although the differences were not significant.

Analyses on the results show that most errors come from the non-chord tones such as passing tones especially in a piece with a fast tempo. Two worst performances in the whole test data sets are such cases (piano solo #3 and string quartet #2). Since the size of an analysis window is fixed, if a given input has a fast tempo, the window will span over more notes, and it is highly likely that more than one chord is contained in a single frame, causing a great confusion to the system. Beat-synchronous analysis done in [1] will help avoid this kind of problems because not only the rate of chord changes is usually slower than the beat, but also non-chord tones rarely occur on-beat.

We also tested our models on the whole recording of Bach's Prelude in C major performed by Glenn Gould. It is approximately 140 seconds long, and contains 753 frames. The ground truth came from harmonic analysis done by the authors. Figure 6 shows recognition results for the parameters trained on both piano music and string quartets.

As can be seen in Figure 6, estimated chord boundaries

[4] http://www.ccarh.org/

[5] http://extras.humdrum.net/

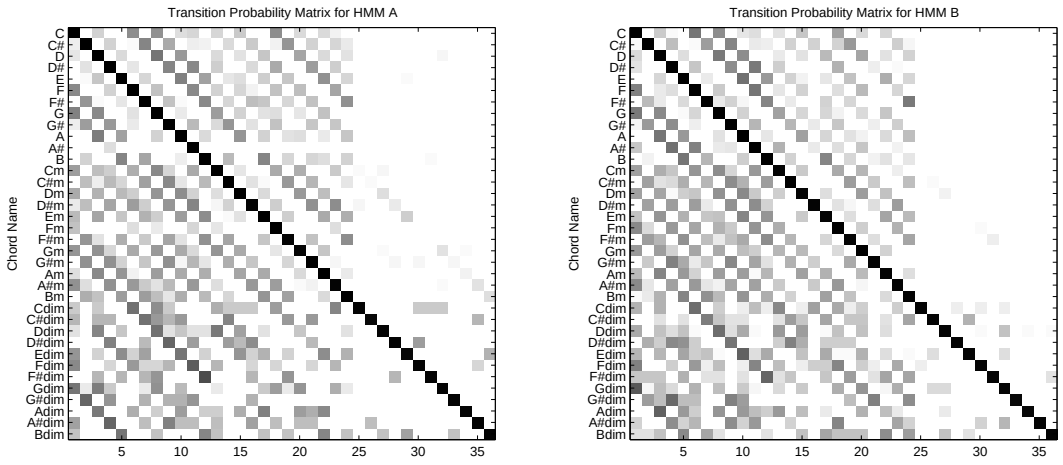

Figure 2: 36x36 transition probability matrices obtained from 81 solo piano music (HMM A) and from 196 string quartets (HMM B). For viewing purpose, logarithm was taken to the original matrices. Axes are numbered in the order of major, minor, and diminished chords.

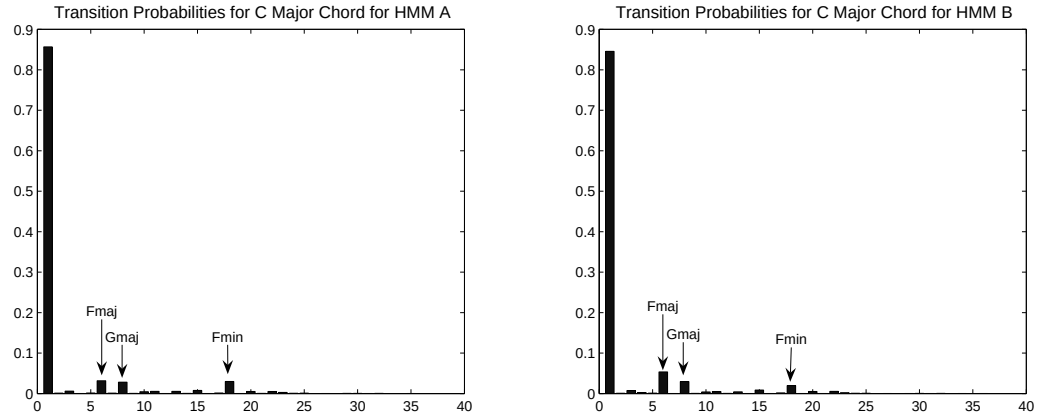

Figure 3: Transition probabilities for C major chord for each HMM.

Table 1: Test Material

Type	Title	Length (sec/# of frames)
Piano Solo	Mozart, Sonata K.309, III	5.96/32
	Beethoven, Sonata Op.13, II	8.26/44
	Mozart, Sonata K.545, II	27.56/148
	Beethoven, Sonata Op.14, No.2, I	7.52/40
	Haydn, Sonata No.33, III	6.18/33
String Quartet	Haydn, Quartet Op.76, No.1, III	3.32/17
	Schubert, Quartet Op.post., I	6.24/33
	Haydn, Quartet Op.20, No.4, I	9.96/53
	Haydn, Quartet Op.3, No.3, IV	10.67/57
	Mendelssohn, Quartet Op.80, IV	16.33/87

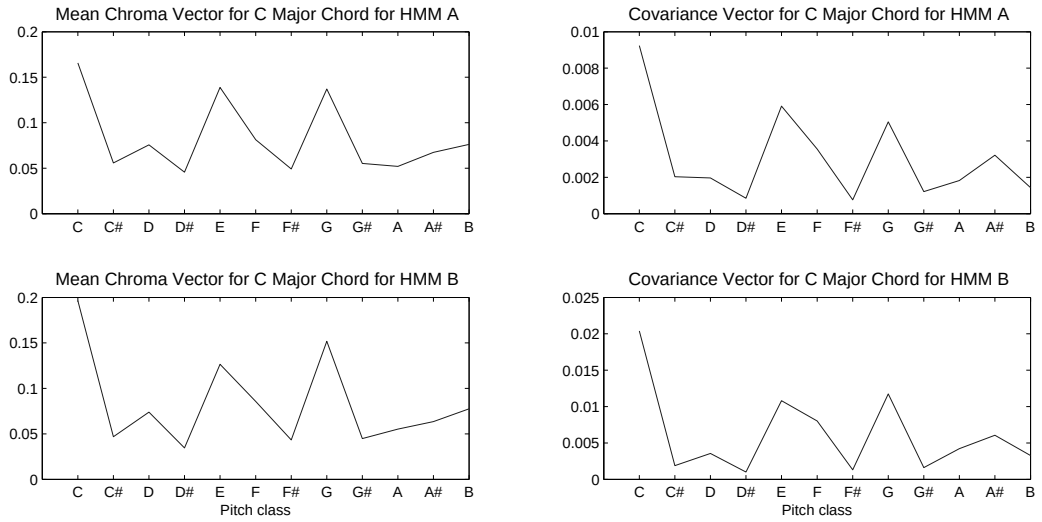

Figure 4: Estimated mean chroma vector and covariance matrix for C major chord for HMM A (top) and HMM B (bottom).

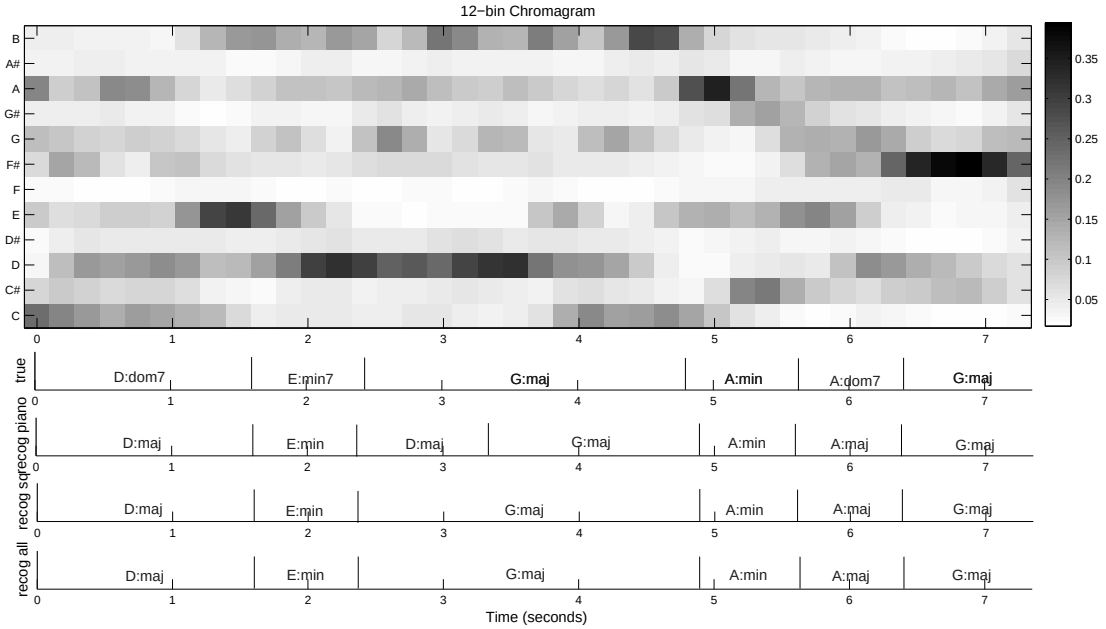

Figure 5: Recognition results for Beethoven's Piano Sonata Op. 14, No. 2, II. Below 12-bin chromagram are shown ground truth and three results for piano model, string quartet model, and combined model, respectively.

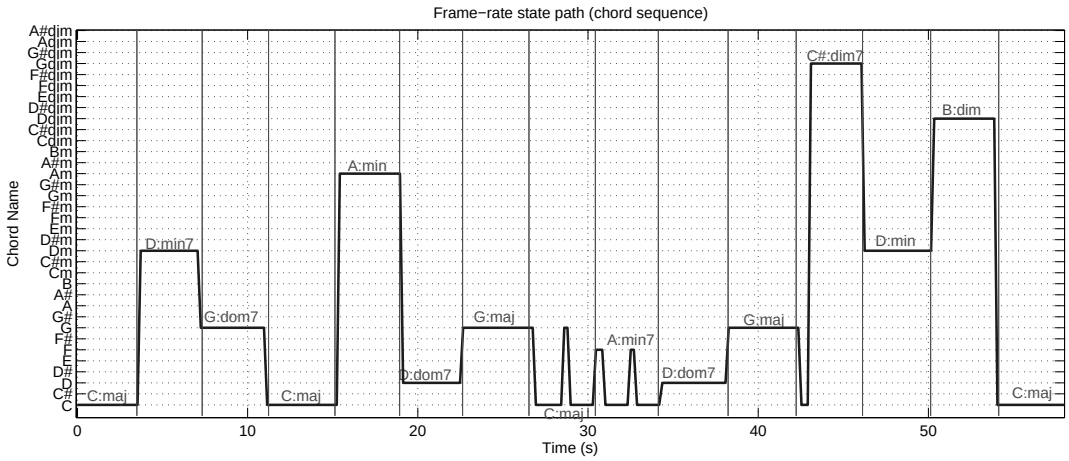

Figure 6: Frame-level state path (chord sequence) of the first 60 seconds from Bach's *Prelude in C Major* (BWV 846) performed by Glenn Gould. Solid vertical lines indicate actual chord boundaries.

Table 2: Recognition Results

Training Data	Test Data	Recog. Rate (%)
Piano	Piano 1	71.88
	Piano 2	93.18
	Piano 3	47.97
	Piano 4	90.00
	Piano 5	100.00
	Total	**68.69**
String Quartet	Piano 1	90.63
	Piano 2	72.73
	Piano 3	56.76
	Piano 4	100.00
	Piano 5	100.00
	Total	**73.40**
All	Piano 1	75.00
	Piano 2	90.91
	Piano 3	56.76
	Piano 4	100.00
	Piano 5	100.00
	Total	**74.41**
Piano	String Quartet 1	100.00
	String Quartet 2	51.52
	String Quartet 3	64.15
	String Quartet 4	94.74
	String Quartet 5	85.06
	Total	**79.35**
String Quartet	String Quartet 1	100.00
	String Quartet 2	45.45
	String Quartet 3	69.81
	String Quartet 4	92.98
	String Quartet 5	86.21
	Total	**79.76**
All	String Quartet 1	100.00
	String Quartet 2	45.45
	String Quartet 3	69.81
	String Quartet 4	94.74
	String Quartet 5	86.21
	Total	**80.16**

are very closely aligned with the ground-truth boundaries. Furthermore, almost all chord names are also correctly recognized. As mentioned in Section 3.2, dominant seventh chords were recognized as their root triads, which we treated as correct recognition. The overall frame-level accuracy was about 92.03%.

Except for some sporadic errors, most consistent errors in the test data came from the confusion between A minor seventh chord and C major chord as can be seen in the middle of Figure 6. A minor seventh is composed of four notes – A, C, E, and G – in which C, E, and G are also chord tones of C major triad. Since we treated A minor triad and A minor seventh as one class, it is highly likely that A minor seventh is misrecognized as C major triad in the presence of a G note, which was the case. We expect that the system will be less sensitive to this sort of confusion if we increase the class size to include seventh chords and train our model on more data.

To further investigate the validity of our model's generality on different musical styles, we performed another test using a totally different type of music in terms of its genre, instrumentation, era, etc. The test material was popular rock music by Michael Chapman. It has typical instrumentation that can be seen in any popular rock music, consisting of electric guitar, electric bass, voice, and drums. Figure 7 illustrates recognition results for the first 20 seconds using the parameters trained on all data.

Again, the model successfully identifies chord names and their boundaries almost perfectly except for a little earlier detection of A minor chord. This is very encouraging because the model was never trained on such type of music, and it still yields a very high performance. This in turn supports the idea of using one single model for all kinds of music.

It is hard to directly compare performance of our system with previous work since we are using different type of music for testing as well as for training. But we believe our high performance, when training on synthetic pieces and testing on live recordings, will only get better as we add more pieces to our training collection and add additional instrumentations.

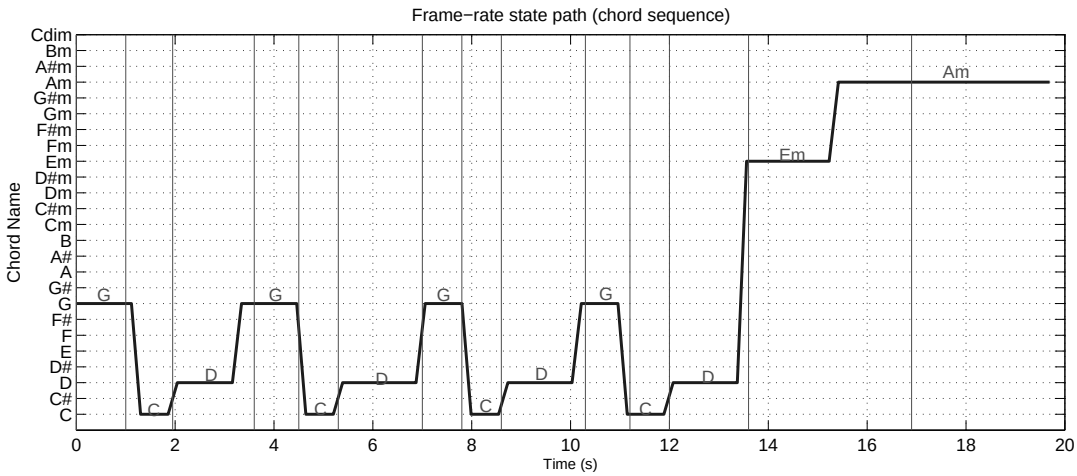

Figure 7: Frame-level state path (chord sequence) of the first 20 seconds from Michael Chapman's *Another Crossroads*. Solid vertical lines indicate actual chord boundaries.

5. CONCLUSION

The main contribution of this work is the automatic generation of labeled training data for a machine learning model for automatic chord recognition. By using the chord labels with explicit segmentation information, we directly estimate the model parameters in an HMM.

In order to accomplish this goal, we have used symbolic data to generate label files as well as to create audio files. The rationale behind this idea was that it is far easier and more robust to perform harmonic analysis on the symbolic data than on the raw audio data since symbolic files such as MIDI files contain noise-free pitch information. In addition, by using a sample-based synthesizer, we could create audio files which have enharmonically rich spectra as in real acoustic recordings.

As feature vectors, we used 12-bin tuned chroma vectors which have been successfully used by others for the chord recognition application. We have defined 36 classes or chord types in our model, which include for each pitch class three distinct sonorities – major, minor, and diminished. We treated seventh chords as their corresponding root triads, and disregarded augmented chords since they very rarely appear in tonal music.

In order to examine the generality of our approach, we obtained two different model parameters trained on two musically distinct data sets, and another set of parameters trained on all data sets. After the model parameters were estimated from the training data sets, various types of unseen test inputs from real recording were fed to the models, and the Viterbi algorithm was applied to find the best probable state path, *i.e.*, chord sequence, at the frame rate. Experiments showed very promising results in terms of model's generality as well as recognition performance.

In this paper, we trained our model only on classical music even if we had two data sets that differ in instrumentation. In the near future we plan to include more training data with different genres and styles to see if we can develop any genre-specific model. Particularly, we believe that a transition probability matrix can be used for musical genre identification. For instance, the transition probability matrix of a blues model will exhibit a very strong I-IV-I-V-I transition pattern since almost all blues music obey a rule of such harmonic progression.

In addition, we consider higher-order HMMs in the future because chord progressions based on Western tonal music theory show such higher-order characteristics. Therefore, knowing two or more preceding chords will help make a correct decision.

6. REFERENCES

[1] J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Symposium on Music Information Retrieval*, London, UK, 2005.

[2] J. C. Brown. Calculation of a constant-Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1990.

[3] T. Fujishima. Realtime chord recognition of musical sound: A system using Common Lisp Music. In *Proceedings of the International Computer Music Conference*, Beijing, 1999. International Computer Music Association.

[4] E. Gomez and P. Herrera. Automatic extraction of tonal metadata from polyphonic audio recordings. In *Proceedings of the Audio Engineering Society*, London, 2004. Audio Engineering Society.

[5] C. A. Harte and M. B. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the Audio Engineering Society*, Spain, 2005. Audio Engineering Society.

[6] S. Kostka and D. Payne. *Tonal harmony with an introduction to 20th-century music*. McGraw-Hill, 2004.

[7] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, 1990.

[8] K. Lee and M. Slaney. Automatic chord recognition from audio using an HMM with supervised learning (accpeted for publication). In *Proceedings of the International Symposium on Music Information Retrieval*, 2006.

[9] S. Pauws. Musical key extraction from audio. In

Proceedings of the International Symposium on Music Information Retrieval, Barcelona, Spain, 2004.

[10] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[11] A. Sheh and D. P. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the International Symposium on Music Information Retrieval*, Baltimore, MD, 2003.

[12] D. Temperley. *The cognition of basic musical structures*. The MIT Press, 2001.