Automatic Chord Recognition from Audio

Alex R. Emmons
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
emmon046@morris.umn.edu

ABSTRACT

Automatic chord recognition from audio is used in the area of *Music Information Retrieval* (MIR) to document and categorize music. In addition to providing harmony to music, chords also provide a way to describe the harmony of a piece. In almost every chord recognition system the audio signal is represented by a *Pitch Class Profile* (PCP), which measures the intensity of energy in each of the frequency regions where musical notes occur [4]. Some systems perform what is known as preprocessing before generating a PCP, to get rid of unwanted frequencies in the audio file. The next step, known as pattern matching, is to assign chord labels by matching the harmonic features to a set of chord models. In this paper we will discuss these processes in greater detail and compare the results of three research cases, each of which uses a different chord recognition system.

Keywords

Automatic chord recognition, hidden Markov models, pitch class profile, signal processing

1. INTRODUCTION

A chord is a set of tones played simultaneously. A chord progression is a sequence of chords over time and is what describes the harmony of a piece [2]. Automatic chord recognition is the process of extracting a chord progression from an audio file. These chord sequences are used by musicians as lead sheets (summaries containing chords, melody, and lyrics) as well as by researchers for tasks such as key detection, genre classification, and lyric interpretation. Performing chord analysis by hand is time consuming, prone to human error, and requires two or more trained experts. This is what makes automatic chord recognition an important area of research [3].

The two main steps of automatic chord recognition are feature extraction and pattern matching. Feature extraction is the process of extracting useful information from audio files, and pattern matching is how chord labels are applied to that data.

There are many challenges encountered by systems that process audio signals. There are background noises, percussion instruments, and other unwanted tones in audio recordings. It is also difficult to distinguish when chords change and to line these points up exactly with the beat. Preprocessing helps eliminate unwanted information from the audio files before or during the feature extraction step, depending on the system. An overview of one of the chord recognition systems looked at in this paper can be seen in Figure 1.

Chord recognition systems have been improving and becoming more usable in recent years. This paper will compare three different systems that use a variety of techniques in each step of the process. By looking at the components of the highest performing systems, we will determine the most effective methods used for each step of the process.

2. BACKGROUND

In order to explain the process of automatic chord recognition, some general information about feature extraction and pattern matching is needed.

2.1 Feature Extraction

The first step of generating a chord progression from audio data is processing the signal to extract harmonic features. Feature extraction is a fairly simple process, but can be made more complex by introducing optimization steps to increase accuracy [3]. Preprocessing is one of these steps, performed before calculating *Pitch Class Profile* (PCP), which is a representation what notes are present over time.

2.1.1 Preprocessing

In Figure 2 the light areas show where frequencies have been detected, and the dark areas show empty space. It is clear that frequencies other than just the chord tones have been detected because the light areas are not solid white and the dark areas not solid black. The goal of preprocessing is to reduce as much of this background noise as possible from the audio file, in an effort to provide a smooth and clear PCP. Depending on the audio resolution, very low pitches can be hard to distinguish and tend to blur across multiple pitches. Percussion sounds also need to be addressed, especially those that create a pitch such as a bass drum.

Another issue is that musical instruments produce a series of harmonics at higher and lower frequencies than the tone that is played. These tones, called overtones, can confuse feature extraction techniques and need to be removed.

UMM CSci Senior Seminar Conference, December 2014 Morris, MN.

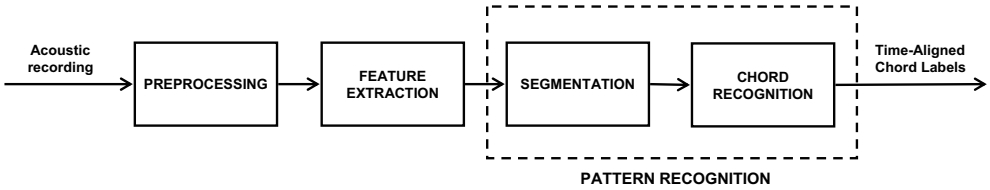

Figure 1: Overview of the chord recognition system used in [4].

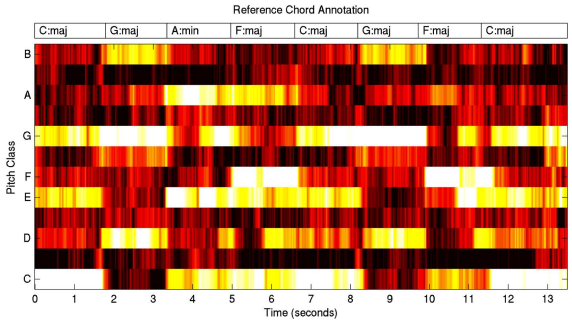

Figure 2: A typical chromagram, or PCP, generated from the opening to *Let It Be* (Lennon/McCartney). Pitch class (chroma) at time t is shown by the intensity or lightness at that point. The true chord progression (simplified) is shown above for comparison.

Explain two types of preprocessing, background spectrum and overtone removal. Explain types and how they work and give more info. Explain more about overtones and how they are detected, give more examples of 'other tones'. the sum of sinusoidal tones of integer multiples of the fundamental frequency. Add a figure.

2.1.2 *Pitch Class Profile*

The pitch of a note is measured in two dimensions - height and chroma (pitch class). Height tells which octave a note belongs to, and chroma tells where a note stands within the octave (the name of the note). Height is not a factor in determining chord type because two notes that are an octave apart have the same chroma value. A chromagram, or PCP, is a 12-dimensional vector representation of chroma, representing the intensity of each of the twelve semitones in the chromatic scale, over a period of time. An example of a common PCP, along with the actual progression, can be seen in Figure 2 [3]. For over a decade PCP has been the most popular way to represent harmonic features for chord recognition. Most new approaches are variations or refinements of this approach [1]. Modern systems can have many more steps in converting audio to PCP including tuning correction, which compensates for music that is not tuned to standard pitch $A4 = 440\ Hz$, and beat-synchronization, which calculates the average pitch between beats to get rid of changes caused by noise and other transients. [3].

Describe them or some of them. Decide between using pitch class and chroma. More detail in paper 1 under feature extraction

2.2 Pattern Matching

Decide between pattern recognition and pattern matching.

Almost all chord recognition systems use PCP or some other chroma-based feature extraction technique. What differentiates these systems is the mechanism used to label chords.

Define chord model.

Generating the chord model against which the PCP will be matched can be done in one of two ways: manually using musical knowledge, or stochastically, by deriving it from real-world music. In a manual system the chroma values being sounded are compared against pre-defined chord templates.

Define these better.

These two methods are compared by Cho and Bello in [1]. Stochastic chord models are more sophisticated and complex. HMMs used to be the method of choice, but many recent systems prefer Gaussian mixture models [1].

Add sentance describing stocastic chord model system after manual system. Get rid of more sophisticated and complex.

2.2.1 *Hidden Markov Models*

A *hidden Markov model* (HMM) is a statistical model which describes a finite set of states, in this case chords, each with a probability distribution. Transitions between these states are governed by a set of transition probabilities that describe the likelihood of transitioning from one to another [1]. HMMs are used in a wide variety of pattern recognition environments such as speech, handwriting, and gesture recognition, as well as in bioinformatics.

Need to re-write this section from draft. Include figures for models. find first instance of HMM.

2.2.2 *Gaussian Mixture Models*

Gaussian mixture models (GMMs) are used in many modern chord recognition systems. A GMM consists of a distribution of weighted Gaussian components that represent descriptions of each chord in the training data.

Need a better definition

To train these models the PCP from the training data is transposed to all C-based chords (C-major and C-minor). These normalized chords are used to train the C-major and C-minor models, and then re-transposed to the remaining 11 keys [1].

Need more here. REWRITE

2.2.3 Support Vector Models

3. RESEARCH CASES

This paper looks at three research cases that involve automatic chord recognition. All of the cases use three common steps in the process: feature extraction, preprocessing, and pattern recognition. Here we will give an overview of each system and the datasets that were used.

Decide between study and case.

3.1 Case 1: Effects of Proper Signal Processing

The first study [4] uses a chord recognition system that begins with a preprocessing block, followed by feature extraction, where PCP is calculated. After this, the signal is segmented along predicted chord boundaries, and then chord labels are assigned to each segment. An overview of this system can been seen in Figure 1 [4].

There are two stages in the preprocessing block, to address background noise and overtones. The first step, Homomorphic Liftering, is a method of separating out the frequencies of musical tones from background and system noise. This is done by finding strong frequency peaks in areas corresponding to the pitch range of the notes. Frequencies above and below a specified range can also be removed to reduce noise. The second step, known as the *Harmonic Product Spectrum* (HPS), is a method which emphasizes frequencies when their overtones are present. HPS is calculated by compressing the spectrum by factors of 1 to R and multiplying the resulting compressed spectra. The resulting output energy is then summed according to pitch class and a PCP is created.

The first step of pattern matching for this system is chord segmentation, where the audio signal is segmented at the boundaries where chords change. This can be difficult because some chords are not played all at once, the notes are played in sequence and held. To find the points where change has occurred the PCP is analyzed frame-by-frame to find significant change in pitch-class content.

The next and final step is assigning chord labels to the segments. Given an instance of the PCP for a region of the audio, the most probable chord label is picked from a set of training PCPs. In this study the use of GMMs and SVMs is compared.

3.1.1 Datasets

In this experiment *MIDI* (Musical Instrument Digital Interface) recordings were used to create time aligned labelled audio. Two datasets were used: one of isolated chords synthesized on piano and strings, and one of continuous single-instrument audio synthesized on piano.

The isolated chord dataset consisted of 7790 chords and inversions, where the notes are stacked in a different order. Three chord complexity levels were tested, labelled DS1, DS2, and DS3. The first involved the common triad types (major, minor, augmented, and diminished), the second included variations of the 7th chord (major 7th, minor 7th, dominant 7th, fully diminished 7th, and half diminished 7th), and in the third all 11 different chord types that the system could recognize were used.

Need to somehow name these and reference them as DS1,2,3 in table.

	Label used in:		
Chord Label	DS1A	DS1B	DS1C
Major	Major	Major	Major
Minor	Minor	Minor	Minor
Major 7	-	Major	Major 7
Minor 7	-	Minor	Minor 7
Dom. 7	-	Major	Dom. 7
Dim.	Dim.	Dim.	Dim.
Full Dim.	-	Dim.	Full Dim.
Half Dim.	-	Dim.	Half Dim.
Augmented	Aug.	Aug.	Augmented
Sus. 4	-	-	Sus. 4
7 Sus. 4	-	-	7 Sus. 4

Table 1: Labels used for isolated chord data.

	Type	Sampling Rate	FFT Ln.	Lifter	HPS R
FV1	FB	44100	32768	yes	5
FV2	PCP	11025	4096	no	1
FV3	PCP	44100	32768	no	1
FV4	PCP	44100	32768	yes	5

Table 2: Feature Vectors used in isolated chord recognition.

This can be seen in Table 1. Four feature vectors (FV), or models were used for comparison on each of these datasets, which can be seen in Table 2. The first model started with preprocessing using homomorphic liftering and harmonic product spectrum. FV2 has a low sampling rate and no preprocessing, FV3 increases the sample rate and FFT length, and FV4 first preprocessed the signal and then used the same homomorphic processing and HPS as FV1.

Need more info here. Decide between using FV and model. Define liftering HPS and alot of other things. Explain table 2 and use the same terminology here.

The continuous single-instrument audio dataset consisted of 50 hymn verses selected from the Trinity Hymnal, with 40 used for training and 10 used for testing. FV4 was used from the isolated chord experiment [4].

Explain SVM in a separate section or leave out.

3.1.2 Results

For the isolated chords dataset, this system was trained using chords synthesized on a piano, and was tested on chords synthesized on both piano and strings. The dataset was randomly divided with 80% of the chords used for training and 20% for testing. Five of these training and testing sets were created and the recognition rates from these were averaged. The overall chord recognition accuracies for piano can be seen in table 3 and for strings in table 4. FV4 performed the best with the chords played on strings, and showed the least difference between recognizing piano and string chords [4].

For the continuous single-instrument dataset, only FV4 was used.

Need to add results of the continuous single-instrument dataset.

Feature Vector	DS1A	DS1B	DS1C
FV1	83.68	61.85	57.24
FV2	90.33	82.44	82.26
FV3	91.76	84.20	84.09
FV4	85.64	79.40	78.93

Table 3: Isolated chord recognition accuracy using GMM, training set: piano, testing set: piano.

Feature Vector	DS1A	DS1B	DS1C
FV1	68.06	42.00	33.62
FV2	42.72	18.60	16.30
FV3	43.49	22.00	18.31
FV4	86.94	80.23	80.18

Table 4: Isolated chord recognition accuracy using GMM, training set: piano, testing set: strings.

3.2 Case 2: HMM with Audio From Symbolic Data

The next study [2] uses a supervised HMM trained with audio from symbolic data.

Supervised and audio from symbolic data is not defined.

The system used here begins with feature extraction, again using PCP. At the same time, chord label data is generated from the MIDI data that was used to generate the audio. This data is used to train a 36-state HMM [2]. These 36 states represent 36 chords - major, minor, and diminished for each 12 notes. The system is trained on a dataset, and then fed a separate dataset for testing and analysis.

Need transition here.

Using an ergodic model, which allows every possible transition from chord to chord, the model parameters are learned, and then the Viterbi algorithm is applied. The Viterbi algorithm finds the most likely path, or chord sequence, by restricting unlikely chord transitions [1]. Two elements are needed to train this model: chord label files, and audio data. In this case they are both being generated from the same symbolic data. See figure 3 for a representation of this. The first step is to use a chord analysis tool to generate a file with complete chord information for a piece of music. Using the same symbolic data, the audio files are generated using a sample-based synthesizer. This audio data is in perfect sync with the chord label file, and simulates a real recording because it contains the overtones that would be generated from real instruments [2].

Need to re-write from section draft! More on Viterbi.

3.2.1 Datasets

The training data used for the first model in this case consisted of 81 solo piano pieces by J.S. Bach, Beethoven, and Mozart. For the second case 196 string quartet pieces by Beethoven, Haydn, and Mozart were used. The models were then tested on excerpts from the Kostka and Payne's book, which includes analysis and audio recordings done by the composers. 10 excerpts - 5 piano solos and 5 string quartets were selected, with no overlap of the training data. The output was compared to the hand-marked data for frame rate accuracy [2].

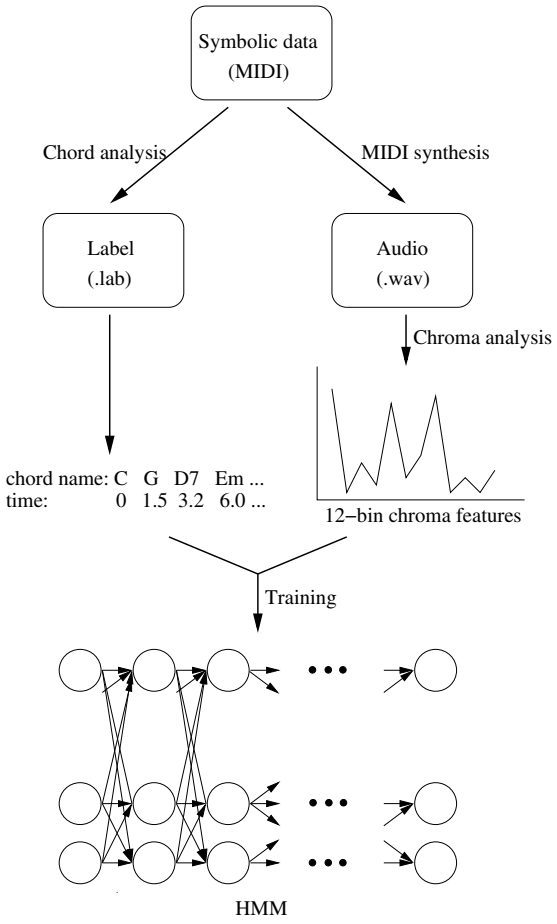

Figure 3: Overview of the HMM trained with audio from symbolic data in [2].

Training Data	Test Data	Recognition Rate
Piano	Piano	68.69
String Quartet	Piano	73.40
All	Piano	74.41
Piano	String Quartet	79.35
String Quartet	String Quartet	79.76
All	String Quartet	80.16

Table 5: Recognition results for HMM trained with audio from symbolic data.

3.2.2 *Results*

In this case there are two datasets, piano and strings, and each can be tested on a system that is trained with either piano or strings. So each combination was tested, in addition to testing on a system trained with both piano and strings. The results of these experiments can be seen in table 5.

3.3 Case 3: Importance of Individual Components

The final study [1] compares the most common methods of each step: feature extraction, pattern recognition, and pre/post filtering. This study describes the overall system a little differently than the other two, with preprocessing included in the feature extraction stage, and HMMs included in the post-filtering stage. The pre-filtering stage is where attempts are made to smooth out the PCP.

Four experiments were conducted in this study: Feature extraction and pattern matching, effect of pre-filtering, effect of post-filtering, and combined pre- and post-filtering. Each experiment was run on 495 chord annotated pop songs. An overview of the system can been seen in figure 4.

decide between chord-annotated and hand marked or define difference.

3.3.1 *Datasets*

The dataset consisted of 180 Beatles songs, 20 Queen songs, 20 songs from the RWC (Real World Computing) pop dataset, and 195 songs from the US-Pop dataset. For training, 5-fold cross validation was used with each group having 99 songs selected randomly. For each fold one group is selected and the other four are used for training. Accuracy is represented by the total duration of correct chords out of the total duration of the dataset [1].

In the first experiment, different combinations of feature extraction and pattern matching are compared to show whether or not model complexity affects performance of the system. They also test the effect of different types of chroma features. The second experiment looks at the pre-filtering techniques of moving filters and beat-synchronization.

Need to define these and provide more here.

The effect of post-filtering experiment compares the use of different stochastic models. The final experiment uses pre- and post-filtering and again looks at moving filters and beat-synchronization.

Needs more work

3.3.2 *Results*

In the first experiment of this case no filtering is applied during the process. This experiment is testing the difference between preprocessing techniques in the feature extraction stage. Results can be seen in table 6. The binary template is the hand-made chord model, and the rest are GMMs with a number indicating the number of Gaussian components used. We would expect to see higher accuracy with more processing and more Gaussian components used, but this is not the case. The highest error was in distinguishing major and minor chords with the same root, because two of the notes are shared and the third is only a half-step different.

Need to add the results for pre and post filtering experiments.

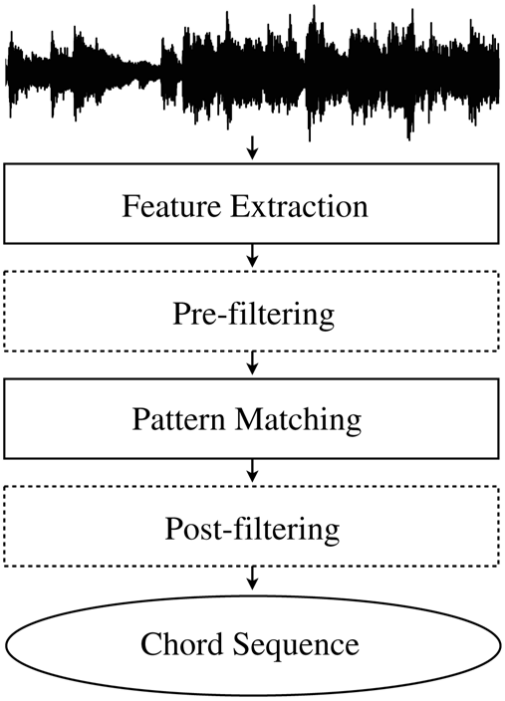

Figure 4: General layout of the chord recognition system used in [1].

4. CONCLUSIONS

Look at the components of the highest performing systems and determine the best techniques used for feature extraction and pattern matching. Discuss the effects of preprocessing during feature extraction and using HMMs to eliminate unlikely sequences. All of the research cases use PCP and preprocessing of some kind. Case 1 and 3 use GMMs. Case 2 and 3 use HMMs.

Talk about some of the issues and pitfalls with these and all chord recognition systems. Including dense recordings, extremely fast chord changes, and other types of chords that are not recognized. There are also chords that can only be determined by their context and chords that can be interpreted in more than one way.

4.1 Future Work

Mention study using online chord database. Genre-specific models.

	Binary Template	GMM-1	GMM-5	GMM-10	GMM-15	GMM-20	GMM-25
Base	46.95	46.46	46.26	48.10	48.39	48.74	48.77
Overtone removal 1	52.12	49.40	47.38	50.24	51.04	51.42	51.71
Overtone removal 2	54.38	54.51	50.49	50.90	51.42	52.14	51.97
Timbre Homogenization 1	45.18	48.24	50.44	49.34	49.36	49.06	49.35
Timbre Homogenization 2	44.37	40.12	39.80	39.61	40.49	40.87	40.86
OR1 & TH1	55.51	**58.30**	57.58	57.73	57.72	57.70	57.69
OR1 & TH2	53.24	53.83	54.66	54.67	54.00	54.03	53.55
OR2 & TH1	55.00	56.29	53.09	53.24	53.28	53.33	53.43

Table 6: Average accuracy without filtering (research case 3, experiment 1).

5. ACKNOWLEDGEMENTS

6. REFERENCES

[1] T. Cho and J. Bello. On the relative importance of individual components of chord recognition systems. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(2):477–492, Feb 2014.

[2] K. Lee and M. Slaney. Automatic chord recognition from audio using a supervised hmm trained with audio-from-symbolic data. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, pages 11–20, New York, NY, USA, 2006. ACM.

[3] M. McVicar, R. Santos-Rodriguez, Y. Ni, and T. D. Bie. Automatic chord estimation from audio: A review of the state of the art. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(2):556–575, Feb 2014.

[4] J. Morman and L. Rabiner. A system for the automatic segmentation and classification of chord sequences. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, pages 1–10, New York, NY, USA, 2006. ACM.