# Chord Recognition*

Quang Hoang Nguyen Vo
Friedrich-Alexander Universityät Erlangen-Nürnberg
quang.nguyen.vo@FAU.de

12.01.2025

## 1 Introduction

Harmony plays a vital role in music. It is the concept of combining different sound together in order to create a cohesive sound to the listener. [2] The most fundamental building block of harmony, at least in Western music, is the chord. A chord is a set of notes played simultaneously, usually consisting of three or more notes. The most basic type of chord is the triad, which consists of three notes played together.

Harmony analysis studies the relationship and progression of chords in a piece of music. It is a crucial part of music theory and is essential for understanding the structure of a piece of music.

## 2 Basic Theory of Harmony

### 2.1 Intervals

An interval is the difference in frequency between two pitches or notes. [4] One of the most basic intervals is the octave, which is the interval between two pitches where one pitch has twice or half the frequency of the other. With this basic interval, we can build other intervals by taking frequency, geometric or note relations into account. For Western music, which uses twelve-tone equal temperament, there are 12 basic intervals, each is one semitone part from the other.

### 2.2 Chord and Triads

Two or more notes plays simultaneously are called a chord. [1] A triad consists of three note played together. The first note is called the root note, which gives the chord its name. The second and third note are called the third and the fifth, respectively. The major triads, consisting of a root note a major third and a perfect fifth, are often associated with a happy or bright sound. The minor triads, on the other hand, swap a major third for a minor third. They are often associated with a sad or melancholic sound. Aside from these, there are diminished triad and the augmented triad. Both of these triads have a more dissonant sound compared to the major and minor triads, and are often used to create tension in music.

### 2.3 Major and Minor Chords

A chord takes the name of its root note, a root note of A will have a chord **A**. The major chord is denoted by the chord name, while the minor chord is denoted by the chord name followed by a lowercase **m**. For example, the major chord with the root note of A is denoted as **A**, while the minor chord with the root note of A is denoted as **Am**.

There are several various way to realising a chord, such as root position, first inversion and second inversion. The root position is when the root note is the lowest note in the chord. The first inversion is when the third note is the lowest note in the chord. The second inversion is when the fifth note is the lowest note in the chord. Besides those exists the broken chord or arpeggio, where the notes are played one after the other instead of simultaneously.

## 3 Template-based Chord Recognition

One of the straighforward method for chord recognition is using pattern matching. Given the chroma sequence $X = (x_1, x_2, \ldots, x_N)$, and a set $\Lambda$ of possible chord labels, the task is to assign each chroma vector $x_n \in \mathbb{R}^{12}$ to a chord label $\lambda \in \Lambda$. Accounting for 24 major and minor chords, the set $\Lambda$ would be as follows

$$\Lambda = \{\mathbf{C}, \mathbf{C}^{\#}, \ldots, \mathbf{B}, \mathbf{Cm}, \mathbf{C}^{\#}\mathbf{m}, \ldots, \mathbf{Bm}\}. \quad (1)$$

For pattern matching,we precompute a set of templates $t_\lambda \in \mathcal{T}, \lambda \in \Lambda$, with each template can be regarded as a chroma vetor that represents a chord. The similarity between a chroma vector $x_n$ and a template $t_\lambda$ is computed using a similarity function $s$.

$$s : \mathcal{F} \times \mathcal{F} \to \mathbb{R} \quad (2)$$

Then, the template matching procedure is to find the chord label that maximises the similarity between the given chroma vector $x_n$ and the templates $t_\lambda$. Figure 3 illustrates the template matching process.

$$\lambda_n^* := \arg\max_{\lambda \in \Lambda} s(\mathbf{t}_\lambda, x_n). \quad (3)$$

*This is the summary for the reading assignment, which is part of the exercise of the lecture *Music Processing Analysis*, Winter Term 2023/24, Friedrich-Alexander Universität Erlangen-Nürnberg. Instructor: Prof. Dr. Meinard Müller, Tutor: Simon Schwär.

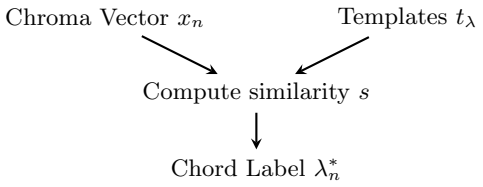


Figure 1: Template Matching Process

## 4 HMM-Based Chord Recognition

Chord progression in music are not arbitary, but rather follow a certain rules. Therefore, one may find that the chord that is being played at a given time is dependent on the chord that was played before. This dependency can be modelled using Hidden Markov Models (HMMs).

### 4.1 Markov Chains and Transition Probabilities

In order to generalise the task [3], given a set of states of size $l \in N$

$$\mathcal{A} := \{\alpha_1, \alpha_2, \ldots, \alpha_l\}, \tag{4}$$

consisting of distinct elements $\alpha_i$ for $i \in [1:l]$, also known as **states**. A chord progression can be modelled as a system that can be described at any time instances $n \in N$ by a state $S_n \in \mathcal{A}$. For simplicity, the system is assumed to be memoryless, this property is expressed as a conditional probability

$$P(S_n|S_{n-1}, S_{n-2}, \ldots, S_1) = P(S_n|S_{n-1}). \tag{5}$$

Other assumptions includes the system being time-invariant. Hence the state transition probability $a_{ij}$ is independent of time $n$ and only depends on the states $S_i$ and $S_j$.

$$a_{ij} := P(S_n|S_{n-1}) = P(S_2|S_1). \tag{6}$$

Needless to say these coefficients follow standard probability constraint

$$\sum_{j=1}^{l} a_{ij} = 1. \tag{7}$$

A system that satisfies these properties are called a Markov chain.

### 4.2 Hidden Markov Models

Based on Markov chain, we extend the concept to Hidden Markov Models (HMMs). Unlike Markov chain, the state sequence is no longer directly observable, hence the name "hidden". Instead, each state can emit entities that are visible to the outside world. As an analogue, we can see the hidden states are the unidentified chord, while the observations are the chroma features of the audio at a given time.

Formally, an HMM is defined by a sequence of hidden states $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$, a probability matrix $A = [a_{ij\in[1:l]}]$ of state transition probabilities and vector $C = (c_1, c_2, \ldots, c_l)$ of initial state probabilities. In case of **discrete HMM**, the emission space is assumed to be a finite set

$$\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_K\}, \tag{8}$$

of size $K \in N$. The emission probability matrix $B = [b_{jk}]$ is defined as the probability of emitting observation $\beta_k$ in state $\alpha_j$, it also satisfies the probability constraint like the transition matrix. In general, a (discrete) HMM is defined by the tuple

$$\Theta = (\mathcal{A}, A, C, \mathcal{B}, B). \tag{9}$$

### 4.3 Uncovering

The task of chord recognition can be seen as an **uncovering problem**. Given an input sequence of observation $O = (o_1, o_2, \ldots, o_M)$ and a HMM $\Theta$, the goal is to find the most likely sequence of hidden states $S = (s_1, s_2, \ldots, s_M)$ that generated the observation sequence. We can use the Viterbi algorithm to find the most likely sequence of hidden states given the observed states. Viterbi algorithm is a dynamic programming algorithm that computes the most likely sequence of hidden states in a HMM. The idea is to compute the optimal path to each state at each time step, then use this information to compute the optimal path to the next state.

$$\delta(i, n) := \max_{(s_1,\ldots,s_n)} P[O(1:n), (s_1, s_2, \ldots, s_{n-1}, s_n = \alpha_i)|\Theta]. \tag{10}$$

$\delta(i, n)$ is highest probability along a single state sequnece $(s_1, \ldots, s_n)$ that ends in state $\alpha_i$ and generates the observation sequence $O(1:n)$. Then the solution is the state sequence or path that yields the maximum value of $\delta(i, n)$. The matrix $\delta$ can be computed recursively as follows

$$\delta(i, n) = \max_{j\in[1:l]}(\delta(j, n-1)a_{ji})b_{ik_n}. \tag{11}$$

After finding the state sequence, we need to perform **backtracking** to find the most likely sequence of hidden states. Let $S^*$ be the optimal state sequence, then the last element $\alpha_{iN}$ is determined by

$$i_N = \arg\max_{j\in[1:l]} \delta(j, N). \tag{12}$$

Finally, the optimal state sequence is found by backtracking from the last state to the first state.

## References


[1] O. Karolyi. *Introducing Music*. Pelican book A659. Penguin Books, 1965.

[2] J. D. Lomas and H. Xue. Harmony in design: A synthesis of literature from classical philosophy, the sciences, economics, and design. *She Ji: The Journal of Design, Economics, and Innovation*, 8(1):5–64, 2022.

[3] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[4] M. Müller. *Fundamentals of Music Processing – Using Python and Jupyter Notebooks*. Springer Verlag, 2nd edition, 2021.