CHORDR: Hidden-Markov-Perceptron for Chord Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

Analysis of harmonic structure in music often starts with labelling every chord in a musical piece. A system for performing chord recognition is very useful for harmonic analysis and for applications such as music search, music similarity identification and music composition. In this report, we present CHORDR, an improved model for automatic chord recognition based on BREVEs HMPerceptron model [5]. We compare our results validated on a corpus of chorales from JS Bach and various other classical compositions.

1 Introduction

Our system takes a MIDI input re-encoded as a list of events. Each event contains information of the frequency of the twelve pitch classes played, the bass note, and its meter (relative accent). We aim to assign each event a chord label - a set of simultaneously played notes.

We also take into consideration chord progressions and accents as contextual information. We consider an event as vertical information and the events that occur around it as horizontal information. Together, they form the framework for evaluating our features functions.

We approach the chord recognition problem as a sequential supervised learning (SSL) problem. For each training piece m, we denote the form (X_m, Y_m), where $X_m = (x_{m,1}, \dots, x_{m,Tm})$ is a sequence of T_m events and $Y_m = (y_{m,1}, \dots, y_{m,Tm})$, are the corresponding chord labels. Using the HMPerceptron model, we obtain a classifier H that given a new sequence of pieces X, predicts the corresponding chord labels $Y = H(X)$ as follows [5]:

$$H(X) = \arg \max_{Y' = y'_1 \cdots y'_T} \sum_t \sum_s \mathbf{w}_s \phi_s(X, y'_t, y'_{t-1}) \tag{1}$$

where ϕ_s are boolean feature functions of the sequence of events X and the previous and current labels.

Finding the classifier H is essentially a path-finding problem over a layered-graph. Here, the vertices represent chord labels which are weighted using vertical basis functions, while transitional weights are calculated using horizontal basis functions. With T layers, and K labels, we have a $T \times K$ graph. Finding the optimal path can be solved using Viterbi algorithm, yielding a complexity of $\Theta(TK^2)$. However, our system uses CarpeDiem algorithm, solving the problem in $\Theta(TKlog(k))$.

2 Model Formulation

BREVEs CarpeDiem path-finding algorithm [6] is already highly efficient and we dont think it needs further improvement. However, we found the original 43 feature functions lacking in functionality.

We first describe the problems of BREVEs feature functions and then present our improvements through CHORDRs feature functions. We denote the current event with x_t, and the currently predicted label with y_t.

2.1 A critique of BREVE's Feature Functions

In BREVE, five consecutive entries in the feature vector are used to indicate exactly how many notes of yt are present in xt. Unfortunately, some chord structures contain 3 distinct notes, while others, like the dominant seventh chord, contain 4 distinct notes. Of the supported chord types, approximately half of the chord structures are comprised of 4 distinct notes, however, these chords are used less frequently. As a result, the learned weights prioritize having three notes of y_t present in x_t, which diminishes the discriminative power of this feature. Consider an event x_t, which contains the pitch classes C, E, G, and $B\flat$. Here, x_t is a fully stated dominant seventh chord, yet it contains 3 distinct notes of A_{m7}, E_{m6}, C_M, C_{M4}, C_{M6}, C_{M7}, G_d, while containing 4 distinct notes of C_V. In general this feature aims to gauge the similarity between a collection of notes and a chord label, however this approach is not optimal, as it reports C_V to be less likely than a number of other labels.

Alternatively, CHORDR uses five consecutive entries in the feature vector to indicate the percentage of notes in xt which are contained in yt. Notably, the percentage is quantized into partitions each with a width of 20%. Using this approach, the learnt weights reflect the fact that having $(80\% - 100\%]$ is the most desirable condition for a predicted label. As a result, both the training error and the testing error decreased using this metric.

Furthermore, the model adopted in BREVE makes the assumption that the presence of each note within a given chord structure is equally significant, when in fact this is not the case. For example, given a C_{M7} chord, or any seventh chord for that matter, the presence or absence of a fifth is a very weak indicator, as it is quite common to voice seventh chords without the fifth. In contrast, the presence or absence of a seventh is extremely important, as this chord type specifically dictates its presence. With this in mind, it is somewhat surprising that Radicioni et al. have two features which indicate if the root and added note of a chord y_t are present in x_t, but neglect to observe the presence of a third or fifth. Additionally, it seems counterintuitive that BREVE only observes if the root of yt is present in x_{t+1}, neglecting to consider the content of x_{t-1}. In light of this reality, CHORDR reports the presence of a third and fifth.

The most prominent difficulty in discerning the correct chord label is in cases where xt is contains a small number of distinct pitches. Without contextual information, it is reasonable to assert that an event containing the pitch classes C and G may represent a C_m, C_M, G_{M4} or $A\flat_{M7}$ chord. Quite frequently, a fully voiced chord is not sounded simultaneously, as the remaining chord tones have been stated previously or are to be stated on the following beat. In order to mitigate these uncertain circumstances, contextual analysis is necessary.

2.2 Determining the Relevance of CHORDR

Unfortunately, it is not admissible to take adjacent events into consideration for every input xt, as the nature of adjacent events is only illuminating in specific circumstances. As a result, one experimented with several differing methods for determining a relevant set of events, before determining that the joint criterion of event similarity and accent trajectory was sufficient.

In order to evaluate the similarity between two events xA and xB, the number of common tones, or equivalently the number of elements in xA xB, are counted. When this value surpasses a threshold, the events are considered to be similar. In most musical genres, chord changes are accented. As a result, a set of unaccented events following an accented event frequently constitutes a single chord structure. Since accents are relative to the surrounding context, a pair of events xA and xA+1 satisfy the accent condition when xA is less accented than xA+1. With regards to the music of J.S. Bach and his Baroque contemporaries, accented beats are directly derived from the metric structure. Accordingly, chord changes often occur on strong beats. Although this musical convention has decreased in prevalence since the Baroque era, chord changes are consistently accented despite the fact that these accents may be syncopated. Additionally, MIDI files contain velocity information which may be used to extrapolate the required accent information. Consequently, this analysis is not limited to the music of J.S. Bach, as it is dependent on accent trajectory not metric structure.

Let xR be a set of consecutive events, necessarily including xt, which collectively satisfy the similarity condition and accent condition. Then, xR is constructed as follows. To begin, xt is the only element contained in xR. Here, xt.accent refers to the relative accent of event t, an integer in [1, 5]. Given L ¡ t, and xL+1 xR, xL satisfies the accent condition if xL.accent ¿ xL+1.accent and the similarity condition if —xL xt— ¿ threshold. Given L ¿ t, and xL-1 xR, xL satisfies the accent condition if xL.accent ¡ xL-1.accent and the similarity condition if —xL xt— ¿ threshold. To reiterate, xL is an eligible addition to xR if it directly precedes or follows one of the events in xR and satisfies the similarity condition and the accent condition.

Events	x_{28}	x_{29}	x_{30}	x_{31}	x_{32}	x_{33}	x_{34}
Label	D_{m7}	D_{m7}	D_{m7}	G_M	C_M	F_M	F_M
Metric Accent	5	2	1	3	4	3	2

For added clarity, an excerpt from the J.S. Bach dataset will be analyzed, as shown in Table 1. Without considering similarity, $x_R = x28, x29, x30$ with $28 \leq t \leq 30, x_R = \{x_31\}$ with $t = 31$, and $x_R = \{x_{32}, x_{33}, x_{34}\}$ with $32 \leq t \leq 34$. Evidenced by the correct chord labels provided above, this segmentation of events is quite reasonable, however, it is evident that the similarity condition is needed to disassociate x_{32} from $\{x_{33}, x_{34}\}$.

2.3 An Overview of Feature Functions in CHORDR

Upon determining xR, several feature functions make observations on general subsets of xR. Notably, in some cases these subsets may be completely empty, and in that case such features report 0. Let xC be the set difference of xR - xt, such that xC contains all contextually relevant events with the exclusion of xt. Let xS be the set intersection of all events in xR, such that xS only contains pitch classes common to all contextually relevant events.

Here, one will briefly rationalize the importance each aforementioned subset of xR. Clearly, it is important to exclusively consider the contents of xt as it is the event we are interested in labelling. Similarly, it is essential to consider the nature of xC, in accordance with the reasons for contextual analysis discussed above. With regards to xS, quite frequently this set contains very few pitch-classes, however, the presence of even a single pitch class in xS, is significant. Often this pitch class is the root or the fifth of the correct chord label. As a result, it was not beneficial to report if xS is completely stated, as this case is extremely rare and would likely only occur with xR = xt = xS. However, in that case this attribute is already captured by reporting if xt is completely stated.

Notably, accuracy was slightly improved when reporting the asserted-degree of yt on pitch classes shared by xt and xt-1 or xt+1 dependent on xR membership, in contrast to reporting the asserted-degree of yt with respect to each event in xR. This is likely due to the fact that outermost events in xR are more likely to contain notes related to a previous chord structure (i.e suspensions) when xR contains more than 3 events. Since these features are dependent on the presence or absence of a single pitch class, they are quite sensitive to these unrelated pitches. Consequently, it is still relevant to consider (xR yt) / —xR— as the effects of these suspended pitches are significantly lessened. As an added measure, it was found beneficial to record the likelihood of a particular chord type, reported in features [43, 43 + K - 1], where K is number of label types. In the table below, the vertical feature functions are summarized, where highlighted rows signify feature functions developed independently for CHORDR. Notably, the horizontal features in BREVE were not modified.

ϕ **class**	ϕ **number**	ϕ **description**
Penalize Added Note on Weak Beats	1	if $X.meter(t) < X.meter(t-1)$ and y_t is an added note chord type $F(1) = 0$
Asserted-notes	2	root of y_t in x_t
	3	third of y_t in x_t
	4	fifth of y_t in x_t
	5	added note of y_t in x_t
Contextually Relevant Asserted-notes for each Chord Degree { root, third, fifth, added note }	6	root of y_t in $x_t \cap x_{t-1} given x_{t-1} in x_R$
	...	...
	10	root of y_t in $x_t \cap x_{t+1} given x_{t+1} in x_R$
	...	...
	14	root of y_t in x_s
	...	...
Root in Context Regardless of Relevance	18	root of y_t in x_{t-1}
	19	root of y_t in x_{t-1}
Completely Stated Chords	20	all notes of y_t are in x_t
	21	all notes of y_t are in x_t
	22	all notes of y_t are in x_t
Percentage of Notes in Y_t (For every 20%)	23	$\frac{(x_t \cap y_t)}{\lvert x_t \rvert}$
	...	...
	28	$\frac{(x_R \cap y_t)}{\lvert x_R \rvert}$
	...	...
	33	$\frac{(x_C \cap y_t)}{\lvert x_C \rvert}$
	...	...
	38	$\frac{(x_S \cap y_t)}{\lvert x_S \rvert}$
Chord Type Likelihood (K label types)	43	y_t is chord type M
	44	y_t is chord type $M4$
	...	...
Bass-at-degree	43 + K	root of y_t is bass note of x_t
	44 + K	third of y_t is bass note of x_t
	45 + K	fifth of y_t is bass note of x_t
	46 + K	seventh of y_t is bass note of x_t
	47 + K	root of y_t is bass note of x_{t+1}
	48 + K	fifth of y_t is bass note of x_{t+1}
	49 + K	root of y_t is bass note of x_{t+1}
	50 + K	fifth of y_t is bass note of x_{t+1}

ϕ **class**		ϕ **description**	ϕ **class**		ϕ **description**
Penalize Added Note on Weak Beats	1	if X.meter(t) ¡ X.meter(t-1) and yt is an added note chord type F(1) = 0	Percentage of Notes in Yt (For every 20%)	23	$(x_t \cap y_t)/\lvert x_t \rvert$
Asserted-notes	2	root of y_t in x_t		28	$(x_R \cap y_t)/\lvert x_R \rvert$
	3	third of y_t in x_t		33	$(x_C \cap y_t)/\lvert x_C \rvert$
	4	fifth of y_t in x_t		38	$(x_S \cap y_t)/\lvert x_S \rvert$
	5	added note of y_t in x_t	Chord Type Likelihood (K label types)	43	y_t is chord type M
Contextually Relevant Asserted-notes for each Chord Degree { root, third, fifth, added note }	6	root of y_t in x_t x_{t-1} given x_{t-1} in x_R		44	y_t is chord type $M4$
	. . .	. . .		. . .	. . .
	10	root of y_t in $x_t \cap x_{t+1}$ given x_{t+1} in x_R	Bass-at-degree	$43+K$	root of y_t is bass note of x_t
	. . .	. . .		$44+K$	third of y_t is bass note of x_t
	14	root of y_t in x_S		$45+K$	fifth of y_t is bass note of x_t
	. . .	. . .		$46+K$	seventh of yt is bass note of xt
Root in Context Regardless of Relevance	18	root of y_t in x_{t-1}		$47+K$	root of y_t is bass note of x_{t+1}
	19	fifth of y_t is bass note of x_{t+1}		$48+K$	fifth of y_t is bass note of x_{t+1}
Completely Stated Chords	20	all notes in y_t are in x_t		$49+K$	root of y_t in x_t
	21	all notes in y_t are in x_R		$50+K$	
	22	all notes in y_t are in x_S		$51+K$	