Journal of New Music Research
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/nnmr20

Using Online Chord Databases to Enhance Chord Recognition
Matt McVicar a, Yizhao Ni a, Raul Santos-Rodriguez b & Tijl De Bie a
a University of Bristol, UK
b Universidad Carlos III de Madrid, Spain
Available online: 27 Jun 2011

To cite this article: Matt McVicar, Yizhao Ni, Raul Santos-Rodriguez & Tijl De Bie (2011): Using Online Chord Databases to Enhance Chord Recognition, Journal of New Music Research, 40:2, 139-152
To link to this article: http://dx.doi.org/10.1080/09298215.2011.573564

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: http://www.tandfonline.com/page/terms-and-conditions

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae, and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand, or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.
Using Online Chord Databases to Enhance Chord Recognition

Matt McVicar¹, Yizhao Ni¹, Raul Santos-Rodriguez², and Tijl De Bie¹
¹University of Bristol, UK; ²Universidad Carlos III de Madrid, Spain

Abstract
Advances in chord recognition research using machine learning are hampered by two factors: the scarcity of annotated training data, and the limited complexity of the features and models used. Both problems are intertwined, as with few training examples, increasing the complexity of the model would inevitably lead to overfitting. In this paper we develop a way to address the first problem by exploiting chord annotations from online chord databases. We show how such chord annotations, despite being noisy and lacking exact chord onset times, can be put to use both during the recognition and training stage. We note that the ability to exploit this large untapped resource may enable researchers to also address the second problem: with more training data, one may be able to use more complex models without running the same high risk of overfitting.

1. Introduction
Musical chords are mid-level features that capture a large amount of information about the harmonic content of a piece. In fact, a chord sequence for a song can be sufficient for a musician to play it in an unpracticed situation. Evidence for this can be found in the popularity of The real book (Various, 2004), or countless chord annotation books. Furthermore, the importance of chords is underscored by their successful application as features in various high-level music information retrieval tasks. For example, chord structures have in the past been used for genre classification (Anglade, Ramirez, & Dixon, 2009) and cover song identification (Bello, 2007). For these reasons, the task of chord recognition from musical audio has attracted significant attention in the past few years.

1.1 Chord recognition
An example of a chord annotation, in Chris Harte’s suggested format (Harte, Sandler, Abdallah, & Gomez, 2005), is shown in Figure 1. The first column indicates the start time of the chord, the second is the chord end time, and the third is the chord label. Chord recognition as we define in this paper is the task of automatically reconstructing such a chord annotation from the audio of a musical piece. The structure of most approaches to this task is to segment a song into a high time resolution sequence of windows (known as frames), after which pattern recognition techniques are used to assign a chord label to each frame, based on the features extracted and the local context.

The annual MIREX (Music Information Retrieval Evaluation eXchange¹) competition has a task dedicated to chord recognition, where participants attempt to extract chord labels and boundaries for a collection of songs. In the most recent competition, results have reached 80.22% accuracy on a collection of Beatles, Queen and Zweieck songs for which the ground truth annotations are available. The chords in this case have been simplified to an alphabet of 25 classes: 12 major chords, 12 minor chords and a no-chord symbol for periods of silence, speaking or other times when no chord label can be assigned. In all our experiments complex chords (‘C aug’, ‘Cmin7’ etc.) were mapped to one of these 25 classes by third tone. Chords without a third were considered to be of major tonality. We index the alphabet

¹www.music-ir.org/mirex/wiki/MIREX_HOME

Correspondence: Matt McVicar, Intelligent Systems Laboratory, MVB, Woodland Road, Bristol, BS8 1UB, UK.
E-mail: matt.mcvicar@bristol.ac.uk
DOI: 10.1080/09298215.2011.573564 © 2011 Taylor & Francis
0.000000 8.730703 N
8.730703 10.216780 D:min
10.216780 12.074376 A:min
12.074376 13.746213 F:maj
13.746213 15.232290 G:maj
...

Fig. 1. Example chord annotation featuring start times, end times and chord labels.

set and refer to it as $A = \{a_i\}_{i=1}^{25}$, where $a_i$ indicates the $i$th chord in the alphabet.

1.2 The state of the art

Most current approaches to chord recognition can be divided into two categories: those based on expert knowledge where parameters are set based on music theoretic knowledge of the developers, and those based on machine learning where the parameters are learned from a fully annotated training data set. Examples of expert systems can be found in Yoshioka, Kithara, Komatani, Ogata, and Okuno (2004), Paiement, Eck, and Bengio (2005), Oudre, Grenier, and Frevotte (2009), Mauch and Dixon (2008, 2010a, 2010b), Mauch (2010), and Papadopoulos and Peeters (2011), whereas a concise list of machine learning based methods is given by Sheh and Ellis (2003), Lee and Slaney (2006), Lee and Slaney (2008), Khadkevich and Omologo (2009), Ellis and Weller (2010), and McVicar and Bie (2010). At least two reviews of machine learning chord recognition algorithms have also been conducted (Burgoyne, Kere-liuk, & Fujinaga, 2007; Cho, Weiss, & Bello, 2010).

At present, both approaches (those learned from data and those based on expert knowledge) are very competitive with each other, with expert models slightly ahead of machine learning approaches due to more complex features and knowledge they apply (Mauch & Dixon, 2010a). Indeed, machine learning based methods are dominated by Hidden Markov Models (HMMs) (Cho et al., 2010) with chromagrams being used as features. The state-of-the-art methods based on expert knowledge, on the other hand, use more complex models such as Dynamic Bayesian Networks and richer features such as bass as well as treble chromagrams that are corrected for the presence of harmonics (Mauch & Dixon, 2010a).

The fact that expert knowledge based systems are slightly ahead of machine learning based systems seems to contrast with the tendency in artificial intelligence research to move away from systems based on expert knowledge to machine learning systems (e.g. in machine translation research there is a trend from approaches based on linguistic rules to statistical approaches (Hutchins, 1997)). The cause of this paradox is arguably that machine learning methods are heavily constrained by the amount of training data, which for chord recognition is very limited (Harte et al., 2005). We believe that because of this, they have to make use of fairly simple features and models (e.g. 12-dimensional chromagrams and HMMs), since using more complex models with such limited data would, without strong regularization, lead to overfitting and poor generalization.

1.3 Overcoming the data scarcity problem

To resolve these problems, various solutions may be adopted. The simplest, but most costly and least scalable solution would be to obtain more fully annotated training data, paying trained musicians to annotate new songs.

In this paper, we try to remedy this problem by a methodologically more challenging but cheaper and scalable approach: we suggest making use of large and freely available chord databases that are maintained by guitar enthusiasts, such as e-chords. Such databases are less directly usable than full annotations, as exact timings of chords are absent (only an ordering is given), and they are affected by various types of errors and omissions, to which we will refer to as noise. In this paper, we show how these noise issues can be overcome, and how such large untapped resources can be used as additional training data.

1.4 On the use of chord websites

E-chords.com is a website where registered users are able to upload the chords, lyrics, keys, and structural information for popular songs (we used e-chords due to the ease of data extraction, although there are many such websites). Although the lyrics and key may provide useful information, we discard them in the current analysis. An annotation in the style typically found on chord databases is shown in Figure 2. Notice that the duration of the chords is not explicitly stated, although an indication of the chord boundaries is given by their position on the page (we will exploit this information in Section 3). Since timings are absent in the e-chords annotations we refer to each chord sequence as an Untimed Chord Sequence or UCS, and denote it as $e \in A^{|e|}$, where $A$ is the chord alphabet used. For instance, the UCS corresponding to the song in Figure 2 is $e = [C G F/E (newline) Am7 (newline) ... G7]$. Note that we cannot infer periods of silence from a UCS. To counteract this we added a no-chord symbol at the start and end of each UCS.

It is worth noting that of some songs multiple versions exist. A variation may be a different interpretation of the

http://www.e-chords.com/
1.5 Contributions in this paper

The remainder of this paper is organized as follows. We start by detailing a simple but close to state-of-the-art chord recognition system that will be used as a baseline and starting point for the contributions in this paper (Section 2). Then we introduce the two main contributions of this paper.

The first contribution is an approach to allow the exploitation of UCSs (such as those from e-chords) at the testing stage for improved chord recognition (Section 3). Although this task is simpler than the MIREX chord recognition task since we use external information, using noisy UCSs to recognize the ideally noise-free chord sequence with exact timings is a nontrivial task.

The second contribution investigates how such UCSs can also be used as additional training data (Section 4). The approach we have developed for this purpose is based on a form of Expectation Maximization (EM). Crucially, the second contribution relies on the first contribution.

Both contributions are evaluated empirically in Section 5.

2. Baseline chord recognition system

To illustrate the benefits of our approach to exploit noisy UCSs for chord recognition, we need to compare it to a baseline method. In this section we will describe this baseline method, including the preprocessing and feature extraction, and the Hidden Markov Model (HMM) on which it is based.

2.1 Preprocessing and feature extraction

We first converted our signals to mono 11,025 Hz, and separated the harmonic and percussive elements with our own implementation of the Harmonic/Percussive Signal Separation algorithm (HPSS) (Ono, Miyamoto, Roux, Kameoka, & Sagayama, 2008). After the tuning of the pieces was estimated by selecting the peak of a histogram of sinusoids found, we used the mirtoolbox (Lartillot & Toiviainen, 2007) to calculate a Fourier transform-based chromagram for each song. The frequency range of the chromagram was 27.5 to 1661.22 Hz, covering six octaves. We conducted a simple cross-validation and found that the optimal Fourier-transform parameters were a window length of 2048 samples (190 ms) and hop length of 256 samples (23 ms). We estimated beat positions using the beat tracker presented in Ellis and Poliner (2007) and took the mean chromagram feature between consecutive beats, and also beat synchronized our chord annotations by taking the most prevalent chord label between beats. Finally, each mean feature vector with the corresponding beat-synchronized chord annotation is regarded as one frame.

song (we assume the annotations on e-chords are uploaded by people without formal musical training), different recordings of the same song, or be in a transposed key. The last of these is common because some keys on the guitar are easier to play in than others. We refer to the multiple files as song redundancies, and to be exhaustive we consider each of the redundancies in every key transposition. We will discuss a way of choosing the best key and redundancy in Section 3.5.
2.2 Ground truth to audio alignment

Similar to the labROSA group (Ellis & Weller, 2010) we discovered that our ground truth annotations were not well aligned to the audio. In fact they had been shifted in the time domain by a constant and also stretched, due to us using remastered versions of the original CDs as well as our CD-ripping software adding small periods of silence to the beginning of the songs. To counteract this we did a simple chord estimate, and used a local search to shift the ground truth annotations to match the prediction optimally. We then retrained and repeated the process until no further shifts were required. We discovered that the maximum shift required was a constant shift of 0.4 s and a linear stretch of 0.5% among the data collection.

2.3 Hidden Markov Model (HMM)

Suppose we have a collection of \( N \) songs and have calculated a chromagram obtained from the procedure described in Section 2.1 for each song, as well as a ground truth annotation as explained in Section 2.2. We will denote the collection of chromagrams for all songs as \( Y \). We calculated a chromagram obtained from the procedure 2.3 Hidden Markov Model (HMM) until no further shifts were required. We discovered that optimally. We then retrained and repeated the process ground truth annotations to match the prediction.

Furthermore, given a chord, the 12-dimensional chromagram vector \( x \) is modeled as a first-order Markovian process. Mathematically, the Markov and conditional independence assumptions allow factorizing the joint probability of the feature vectors and chords \( (X, y) \) of a song as follows:

\[
P(X, y|\Theta) = P_{\text{ini}}(y_1|\Theta) \cdot P_{\text{obs}}(x_1|y_1, \Theta) \cdot \prod_{i=2}^{N} P_{\text{tr}}(y_i|y_{i-1}, \Theta) \cdot P_{\text{obs}}(x_i|y_i, \Theta).
\]

Here, \( P_{\text{ini}}(y_1|\Theta) \) is the probability that the first chord of a song is equal to \( y_1 \), \( P_{\text{tr}}(y_i|y_{i-1}, \Theta) \) is the probability that a chord \( y_{i-1} \) is followed by chord \( y_i \) in the subsequent frame, and \( P_{\text{obs}}(x_i|y_i, \Theta) \) is the probability density for chromagram vector \( x_i \) given that the chord of the \( t \)th frame is \( y_t \).

It is common to assume that the HMM is stationary, which means that \( P_{\text{tr}}(y_i|y_{i-1}, \Theta) \) and \( P_{\text{obs}}(x_i|y_i, \Theta) \) are independent of \( t \). Then the parameter set \( \Theta \) of an HMM for chord recognition is commonly given by

\[
\Theta = \{ P_{\text{tr}}, P_{\text{ini}}, \{ \mu_i \}_{i=1}^{|A|}, \{ \Sigma_i \}_{i=1}^{|A|} \},
\]

where \( P_{\text{tr}} \in \mathbb{R}^{|A| \times |A|} \), \( P_{\text{ini}} \in \mathbb{R}^{|A|} \), \( \mu_i \in \mathbb{R}^{|A|} \), and \( \Sigma_i \in \mathbb{R}^{12 \times 12} \). With this parameterization, the initial distribution \( P_{\text{ini}} \) and transition distribution \( P_{\text{tr}} \) are defined as multinomial distributions with \( P_{\text{ini}}(y_1|\Theta) = P_{\text{ini}}(y_1) \) and \( P_{\text{tr}}(y_i|y_{i-1}, \Theta) = P_{\text{tr}}(y_i|y_{i-1}, \Theta) \). The observation probability distribution for chord \( a_i \), \( P_{\text{obs}}(x_i|a_i, \Theta) \) is modeled as a 12-dimensional Gaussian with mean vector \( \mu_i \) and covariance matrix \( \Sigma_i \).

In the machine learning setting, \( \Theta \) is estimated as \( \Theta^* \) on a set of labelled training data \( \{X, y\} \), usually using Maximum Likelihood (ML) estimation, and our baseline system follows this approach. Mathematically,

\[
\Theta^* = \arg \max_{\Theta} P(X, y|\Theta),
\]

where \( P(X, y|\Theta) = \prod_{i=1}^{N} P(X_i, y_i|\Theta) \).

Finally, given the HMM with parameters \( \Theta \), the chord recognition task can be formalized as the computation of the chord sequence \( y^* \) that maximizes the joint probability with the chromagram feature vectors \( X \) of the given song

\[
y^* = \arg \max_{y} P(X, y|\Theta).
\]
It is well known that this task can be solved efficiently using the Viterbi algorithm (Rabiner, 1989).

3. Improving chord recognition using the e-chords website

Our first contribution is to make use of the partially labelled chord sequence annotations such as those on e-chords to help with chord recognition. The principle is to label chord sequence annotations such as those on e-chords to help with chord recognition. The principle is to use the UCSs to constrain the set of possible chord transitions in a certain way. Mathematically, this is done by modelling the joint probability of chords and chromograms of a song \((X, y)\) by

\[
P'(X, y|\Theta, e) = P_{\text{ini}}(y_1|\Theta) \cdot P_{\text{obs}}(x_1|y_1, \Theta) \cdot \prod_{i=2}^{y} P'_{\text{tr}}(y_i|y_{i-1}, \Theta, e) \cdot P_{\text{obs}}(x_i|y_i, \Theta).
\]

(3)

This distribution is the same as in (1), except that the transition distribution \(P'_{\text{tr}}\) now also depends on the e-chord UCS \(e\) for this song, essentially by constraining the transitions that are allowed as we will detail below.

An important benefit of this approach is that the chord recognition task can still be solved by the Viterbi algorithm, albeit applied to an altered model with an augmented transition probability distribution. That is, chord recognition using the extra information from the UCS amounts to solving

\[
y^* = \arg \max_y P'(X, y|\Theta, e).
\]

(4)

The more stringent the constraints imposed on \(P'_{\text{tr}}\) are, the more information from the UCS is used, but also the more detrimental the effect of noise will be. On the other hand, if the information used from the UCS is less detailed, noise will have a smaller effect. The challenge is to find the right balance and to understand which information from the UCSs can be trusted for most of the songs. In the following subsections we will explore various ways in which e-chord UCSs can be used to constrain chord transitions, in search for the optimal trade-off. The empirical results will be demonstrated in Section 5.1.

3.1 Alphabet Constrained Viterbi (ACV)

Given the e-chord UCS \(e \in \mathcal{A}^6\) for a test song, the most obvious constraint that can be placed on the original state diagram is to restrict the output to only those chords appearing in \(e\). This is implemented simply by setting the new transition distribution \(P'_{\text{tr}}\) as

\[
P'_{\text{tr}}(a_j|a_i, \Theta, e) = \begin{cases} \frac{1}{Z_a} P_{\text{tr}}(a_i, a_j) & \text{if } a_i \in e \land a_j \in e, \\ 0 & \text{otherwise}, \end{cases}
\]

(5)

with \(Z_a\) as a normalization factor.\(^4\) An example of this constraint for a segment of the annotation from Figure 2 is illustrated in Figure 4(a), where the hidden states (chords) with 0 transition probabilities are removed. We call this method Alphabet Constrained Viterbi, or ACV.

3.2 Alphabet and Transition Constrained Viterbi (ATCV)

We can also directly restrict the transitions that are allowed to occur by setting all \(P_{\text{tr}}(a_i, a_j) = 0\) unless we observe a transition from chord \(a_i\) to chord \(a_j\) in the e-chords file (e.g. Figure 4(b)). This is equivalent to constraining \(P'_{\text{tr}}\) such that

\[
P'_{\text{tr}}(a_j|a_i, \Theta, e) = \begin{cases} \frac{1}{Z_a} P_{\text{tr}}(a_i, a_j) & \text{if } a_i a_j \in e, \\ 0 & \text{otherwise}, \end{cases}
\]

(6)

where \(a_i a_j\) denotes a pair and \(Z_a\) is the normalization factor. We call this method Alphabet and Transition Constrained Viterbi, ATCV.

3.3 Untimed Chord Sequence Alignment (UCSA)

Yet a more stringent constraint on the chord sequence \(y\) for a test song would be to require it to respect the exact ordering of chords as seen in the UCS \(e\). Doing this corresponds to finding an alignment of \(e\) to the audio, since all that remains for the chord recognizer to do is ascertain the duration of each chord. In fact, symbolic to audio sequence alignment has previously been exploited as a chord recognition scheme and was shown to achieve promising results on a small set of Beatles and classical music (Sheh & Ellis, 2003), albeit in an ideal noise-free setting.

Interestingly, sequence alignment can be formalized as Viterbi inference in an HMM with a special set of states and state transitions (see e.g. the pair-HMM discussed in Durbin, Eddy, Krogh, and Mitchison (1998)). In our case, this new hidden state set \(\mathcal{A}' = \{1, \ldots, |e|\}\) corresponds to the ordered indices of the chords in the UCS \(e\) (see Figure 4(c)). The state transitions are then constrained by designing \(P'_{\text{tr}}\) such that

\[
P'_{\text{tr}}(j|i, \Theta, e) = \begin{cases} \frac{1}{Z_a} P_{\text{tr}}(e_i, e_j) & \text{if } j \in \{i, i+1\}, \\ 0 & \text{otherwise}, \end{cases}
\]

(7)

where \(Z_a\) denotes the normalization factor for the new hidden state \(e_i\).

Briefly speaking, each state (i.e. each circle in Figure 4(c)) can only undergo a self-transition or move to the next state, constraining the chord prediction to follow the same
order as appears in the e-chord UCS. This method is named Untimed Chord Sequence Alignment (UCSA).

3.4 Jump Alignment (JA)

A prominent and highly disruptive type of noise in e-chords is that the chord sequence is not always complete or in the correct order. As we will show in Section 5.1, exact alignment of chords to audio showed a decrease in performance accuracy. This was due to repetition cues (e.g. ‘Play verse chords twice’) not being understood by our scraper. Here we suggest a way to overcome this by means of a more flexible form of alignment to which we refer to as Jump Alignment.
Beyond it, not only to UCSs from the large e-chords database but also to databases contain this, such that the JA method is applicable. However, most online chord databases lack the availability of line information. Therefore, methods such as the Jump Alignment (JA) method presented in Fremerey, Müller, and Clausen (2010), which makes use of the line information of the UCSs, can be applied. In the UCSA setting, the only options were to remain on the current line or jump to other parts of the annotation. The salient feature of JA is that instead of moving from chord to chord in the e-chords sequence, at the end of an annotation line we allow jumps to the beginning of the line, as well as previous and subsequent lines. This means that it is possible to repeat sections which may correspond to repeating verse chords etc.

An example of a potential JA is shown in Figure 5. In the strict alignment method (UCSA), the decoder would be forced to go from the G7 above ‘blue’ to the C above ‘Ooh’ to start the chorus (see Figure 4(c)). We now have the option of ‘jumping back’ from the G7 to the beginning of the first line (or any other line). We can therefore take the solid line path, then jump back (dashed path 1), repeat the solid line path, and then jump to the chorus (solid path 2). This gives us a path through the chord sequence that is better aligned to the global structure of the audio.

This flexibility is implemented by allowing transitions corresponding to jumps backward (dot-and-dashed lines in Figure 4(d)) and jumps forward (dashed lines in Figure 4(d)). The transition probability distribution $P_{tr}'$ (still on the new augmented state space $A' = \{1, \ldots, |e|\}$ introduced in Section 3.3) is then expressed as

$$P_{tr}'(j|i, \Theta, e) = \begin{cases} \frac{1}{|e|} P_{tr}(e_i, e_j) & \text{if } j \in \{i, i+1\}, \\ \frac{2}{|e|} P_{tr}(e_i, e_j) & \text{if } i < j \land i \text{ is the end and } j \text{ the beginning of a line}, \\ \frac{2}{|e|} P_{tr}(e_i, e_j) & \text{if } i > j \land i \text{ is the end and } j \text{ the beginning of a line}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Although Jump Alignment is similar to the jump dynamic time warping (jumpDTW) method presented in Fremerey, Müller, and Clausen (2010), it is worth pointing out that the situation we encountered is more difficult than that faced by music score-performance synchronization, where the music sections to be aligned are noise-free, and where clear cues are available in the score as to where jumps may occur. Furthermore, since the applications of JA and jumpDTW are in different areas, the optimization functions and topologies are different.

We should point out that our method depends on the availability of line information. However, most online chord databases contain this, such that the JA method is applicable not only to UCSs from the large e-chords database but also beyond it. Hence, if the current chord to be aligned is not the end of an annotation line, the only transitions allowed are to itself or the next chord, which executes the same operations as in UCSA. At the end of a line, an additional choice to jump backward or forward to the beginning of any line is permitted with a certain probability. In effect, Jump Alignment can be regarded as a constrained Viterbi alignment, in which the length of the Viterbi path is fixed to be $|X|$.

This extra flexibility comes at a cost: we must specify a jump backward probability $p_b$ and a jump forward probability $p_f$ to constrain the jumps. To tune these parameters, we used maximum likelihood estimation, which exhaustively searches a pre-defined $(p_b, p_f)$ matrix and picks up the pair that generates the most probable chord labelling for an input $X$.

Note that UCSA is a special case of JA which is obtained by setting both jump probabilities $(p_b, p_f)$ to 0. The pseudo-code of the JA algorithm is presented in Table 1, where two additional matrices $P_{obs} = \{P_{obs}(x_i|\theta, \Theta)\}_{i=1, \ldots, |X|}$ and $i=1, \ldots, |A|}$ and $P_{tr}' = \{P_{tr}'(j|i, \Theta, e)\}_{i,j=1, \ldots, |e|}$, are introduced for notational convenience.

### 3.5 Choosing the best key and redundancy

In all the above methods we needed a way of predicting which key transposition and redundancy was the best to use, since there were multiple versions and key transpositions in the database. Similar to Lee and Slaney (2008), we suggest to use the log-likelihood as a measure of the quality of the prediction (this scheme is referred to as...
The Viterbi path
Derive the Viterbi path (3)

(1) Restructure the transition probabilities
Initialize a new transition matrix $P'_u \in \mathbb{R}^{k \times k}$
for $i = 1, \ldots, [e]$
for $j = 1, \ldots, [e]$
(a) if $i = j$ then $P'_u(i, j) = P_{tr}(e_i, e_j)$
(b) if $i = j - 1$ then $P'_u(i, j) = P_{tr}(e_i, e_j)$
(c) if $i$ is the end of a line and $j$ is the beginning of a line
   if $i > j$ then $P'_u(i, j) = p_h \times P_{tr}(e_i, e_j)$
   if $i < j$ then $P'_u(i, j) = p_l \times P_{tr}(e_i, e_j)$
(d) else $P'_u(i, j) = 0$
Re-normalize $P'_u$ such that each row sums to 1

(2) Fill in the travel grid
Initialize a travel grid $G \in \mathbb{R}^{[X] \times [e]}$
Initialize a path tracing grid $TR \in \mathbb{R}^{[X] \times [e]}$
for $j = 1, \ldots, [e]$
$G(1, j) = P_{obs}(x_1, e_j) \times P_{ini}(e_j)$
for $t = 2, \ldots, [X]$
for $j = 1, \ldots, [e]$
$G(t, j) = P_{obs}(X_t, e_j) \times \max_{i = 1}^{[e]} \left(G(t - 1, i) \times P'_u(i, j)\right)$
$TR(t, j) = \arg \max_{i = 1}^{[e]} \left(G(t - 1, i) \times P'_u(i, j)\right)$

(3) Derive the Viterbi path
The path probability $P = G([X], [e])$
The Viterbi path $VP = \{[e]\}$
for $t = [X], \ldots, 2$
$VP = \{TR(t, VP(t)), VP\}$
$VP = e(VP)$
Output: The Viterbi path $VP$ and the path likelihood $P$

‘Best-Guess’ in Table 2). In the experiments in Section 5.1 we investigate the performance of this approach to estimate the correct transposition, showing that it is almost as accurate as using the key and transposition which maximized the performance (it is referred to as ‘Best-Accuracy’ in Table 2).

4. Using online chord databases for training chord recognition systems
We have outlined an approach to using e-chords-like data to improve the prediction process. Obviously, this can only be applied to songs of which the UCSs are available from the online chord databases. Although this is the case for many popular songs, it would be more desirable if we could use the UCS annotations to inform the learning process itself, rather than just using it to improve recognition.

Table 1. Pseudo-code of the Jump Alignment algorithm.

| Input: A chromagram X and its UCS e, the observation probability matrix |
| $P_{obs}$, the transition probability matrix $P_{tr}$, the initial distribution vector $P_{ini}$ and the jump probabilities $p_h$ and $p_l$ |
| (1) Restructure the transition probabilities |
| Initialize a new transition matrix $P'_u \in \mathbb{R}^{k \times k}$ |
| for $i = 1, \ldots, [e]$ |
| for $j = 1, \ldots, [e]$ |
| (a) if $i = j$ then $P'_u(i, j) = P_{tr}(e_i, e_j)$ |
| (b) if $i = j - 1$ then $P'_u(i, j) = P_{tr}(e_i, e_j)$ |
| (c) if $i$ is the end of a line and $j$ is the beginning of a line |
| if $i > j$ then $P'_u(i, j) = p_h \times P_{tr}(e_i, e_j)$ |
| if $i < j$ then $P'_u(i, j) = p_l \times P_{tr}(e_i, e_j)$ |
| (d) else $P'_u(i, j) = 0$ |
| Re-normalize $P'_u$ such that each row sums to 1 |
| (2) Fill in the travel grid |
| Initialize a travel grid $G \in \mathbb{R}^{[X] \times [e]}$ |
| Initialize a path tracing grid $TR \in \mathbb{R}^{[X] \times [e]}$ |
| for $j = 1, \ldots, [e]$ |
| $G(1, j) = P_{obs}(x_1, e_j) \times P_{ini}(e_j)$ |
| for $t = 2, \ldots, [X]$ |
| for $j = 1, \ldots, [e]$ |
| $G(t, j) = P_{obs}(X_t, e_j) \times \max_{i = 1}^{[e]} \left(G(t - 1, i) \times P'_u(i, j)\right)$ |
| $TR(t, j) = \arg \max_{i = 1}^{[e]} \left(G(t - 1, i) \times P'_u(i, j)\right)$ |
| (3) Derive the Viterbi path |
| The path probability $P = G([X], [e])$ |
| The Viterbi path $VP = \{[e]\}$ |
| for $t = [X], \ldots, 2$ |
| $VP = \{TR(t, VP(t)), VP\}$ |
| $VP = e(VP)$ |
| Output: The Viterbi path $VP$ and the path likelihood $P$ |

Table 2. Results for the baseline, constraining the alphabet and/or transitions from the Viterbi algorithm, as well as the UCS alignment and the JA methods. We used either the aligned ground truth annotations (first row), best accuracy (second row) or best guess UCSs (third row). Bold numbers refer to the best results. $P$-values of T-test for statistical significance in the difference between the baseline and our constrained methods are shown in the lower table.

<table>
<thead>
<tr>
<th>Viterbi</th>
<th>ACV</th>
<th>ATCV</th>
<th>UCSA</th>
<th>JA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth 75.78% 79.08% 82.26% 86.65% –</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best-Accuracy 75.78% 77.86% 79.73% 73.08% 82.39%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best-Guess 75.78% 77.03% 79.36% 72.84% 80.51%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$P$-value in paired T-test

<table>
<thead>
<tr>
<th>ACV</th>
<th>ATCV</th>
<th>UCSA</th>
<th>JA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Truth 0 3.18e − 30 1.34e − 40 –</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best-Accuracy 6.49e − 9 7.77e − 16 9.97e − 1 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best-Guess 2.70e − 3 7.98e − 13 9.98e − 1 5.55e − 17</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this section, we will explain how UCSs can be used as additional data for training, despite the various types of noise they are affected by. Note that the UCSs can be regarded as incomplete annotations of the songs they correspond to, such that the problem we face is reminiscent of semi-supervised learning, where the aim is to make use of both labelled and unlabelled data for training (Zhu & Goldberg, 2009).

Training an HMM from a set of fully annotated songs as well as a set of songs for which only an UCS is available thus amounts to inference of the model parameters $\Theta$ given partial observations. This is known to be a computationally hard problem, although in practice it can be adequately addressed using iterative procedures such as Expectation-Maximization (EM). In this particular study, we chose to use Hard-EM, which is also known as Viterbi EM or Sparse EM (Neal & Hinton, 1998; Spitkovsky, Alshawi, Jurafsky, & Manning, 2010). Using Hard-EM is particularly convenient here, as it allows us to directly use the methods presented in Section 3.

Note that the use of EM for training an HMM-based chord recognition system has been investigated before in Sheh and Ellis (2003). However, their method assumed the availability of noise-free annotations in the correct key, such that it would be unable to exploit annotations from noisy but large online chord databases. In contrast, our method explained below is able to deal with the various types of noise these databases are affected by, as we demonstrate experimentally in Section 5.2. Furthermore, the method of Sheh and Ellis (2003) was evaluated on a very limited set of songs only.
4.1 Additional notation

In the following, suppose there is a collection of \( N \) songs, where \( N = N_c + N_e \). Let \( N_c \) be the number of songs corresponding to the fully annotated ground truth data, and let \( N_e \) be the number of songs for which only UCSs are available. Then, the whole chromagram collection \( X \) will consist of both subsets \( X = X_c \cup X_e \), where \( X_c = \{X^n|X^n \in \mathbb{R}^{12 \times T^n}\}_{i=1}^{N_c} \) and \( X_e = \{X^n|X^n \in \mathbb{R}^{12 \times T^n}\}_{i=N_c+1}^{N} \). We will refer to the pair \( \{X_c,Y_c\} \) of training chromagrams and corresponding ground truth annotations as the core training set. The set of \( N_e \) songs with only UCSs is referred to as the expansion set \( \{X_e\} \). Since there are no ground truth annotations for these songs, any annotation set \( Y_e = \{y^n|y^n \in A^{T^n}\}_{n=N_c+1}^{N} \) will be a chord prediction collection estimated based on its UCS collection \( e = \{c^n|c^n \in A^{(T^n)}\}_{n=1}^{N_c} \) using a method presented in Section 3. Together, the core training and the expansion sets form the training data for our algorithm.

4.2 Hard-EM formulation

Given the entire training data, the probability of the chord sequences for the songs in the training set conditioned on the parameters is given by the likelihood function:

\[
L_{X_c,Y_c,X_e} (\Theta, Y_e) = \prod_{n=1}^{N_c} P(X^n, y^n | \Theta) \prod_{n=N_c+1}^{N} P(X^n, \hat{y}^n | \Theta, e^n),
\]

(9)

where \( P(X^n, y^n | \Theta) \) is as in Equation 1 and \( P(X^n, \hat{y}^n | \Theta, e^n) \) is as in Equation 3.

A common approach to infer the parameters \( \Theta \) is then to optimize the likelihood function jointly with respect to the estimated chord sequences \( \hat{Y}_e \) as well as the parameters \( \Theta \):

\[
\max_{\Theta} \max_{\hat{Y}_e} L_{X_c,Y_c,X_e} (\Theta, \hat{Y}_e).
\]

(10)

This optimization is a hard problem in general. However, it can typically be solved using Hard-EM, a process of alternatively maximizing over these two sets of parameters. The algorithm includes the following steps:

1. Train a set of HMM parameters \( \Theta^* \) from the core training set.
2. Repeat the following until convergence:
   a. Fix the HMM parameters \( \Theta^* \) and do chord prediction on the expansion set

   \[
   \hat{Y}_e^* = \arg \max_{\hat{Y}_e} L_{X_c,Y_c,X_e} (\Theta^*, \hat{Y}_e).
   \]

   This is equivalent to predicting the chord sequence \( \hat{Y}_e^* \) for the \( n \)th song using Equation 4 and letting \( \hat{Y}_e^* = \{\hat{y}_e^n\}_{n=1}^{N_e} \).

b. Merge the core training and the expansion sets and retrain the HMM parameters \( \Theta \)

\[
\Theta^* = \arg \max_{\Theta} L_{X_c,Y_c,X_e} (\Theta, \hat{Y}_e^*).
\]

Note that step 2(a) relies on the methods presented in Section 3. Since the JA method worked best as shown in Section 5.1, it was chosen for 2(a) in the corresponding experiments.

The learning procedure is visualized in the schematic Figure 6. It aims to explore the comprehensive information from a partially annotated data so as to provide a more robust parameter set \( \Theta \) for predicting chord sequences. Generally speaking, this semi-supervised learning scheme deals with a new situation in chord recognition research, where there is a small collection of high-quality chord annotations (e.g. C. Harte’s annotated data), a large set of partially annotated chord sequences (e.g. e-chords UCSs), and a vast set of unknown songs as a test set.

5. Experimental evaluation

Here we summarize the results of the main experiments we conducted, which consists of our constrained Viterbi and alignment methods compared to the baseline system, and using them in Hard-EM to train a chord recognition model in a semi-supervised scheme. For direct comparison, we restrict ourselves to an alphabet of 25 chords (12 major, 12 minor, and no-chord), and we use the same evaluation metric used in the MIREX competitions to report performance (for a description of this evaluation, see below).
Our experiments make use of the fully annotated songs from The Beatles and the Greatest Hits of Queen, 180 and 20 respectively (obtained with thanks from Mauch et al. (2009)). Some experiments below also use 52 songs from Oasis for which at least one e-chords UCS is available, although no ground truth annotations are available for these songs. For the experiments requiring an e-chords UCS for songs for The Beatles, we had to restrict ourselves to the 175 songs for which an e-chords version is available, and we developed a scraper for obtaining the UCSs for these 175 songs. The final Beatles song corpus was then the intersection of those songs which are used in MIREX, and those which appeared on e-chords, which numbered 171 in total. More specific information on the experiments is given below.

5.1 Effectiveness of using e-chords to improve chord recognition

The aim of these experiments was to discover which of the methods detailed in Sections 2 and 3 was the most effective. We tested using either the aligned ground truth annotations, or (when redundancies were available) either the transposition and key which maximized the performance or the one with the largest likelihood of alignment. We used regular Viterbi as the baseline system, then ran Alphabet Constrained Viterbi, Alphabet and Transition Constrained Viterbi, Untimed Chord Sequence Alignment and Jump Alignment methods. Three-fold cross-validation was used and the evaluation was the same as is used in MIREX (i.e. number of correctly identified frames divided by total number of frames, with each fold containing approximately \( \frac{3}{5} \) of each album in the training set).

The results are shown in Table 2, including p-values demonstrating significance of the results. A p-value of less than 0.05 indicates an improvement that is significant at the 5% level. The first row shows the maximum potential of our method. These results would be achieved if e-chords was completely noise-free, i.e. the annotations were in fact just the ground truth chord annotations without timing information. There is no score for using JA on the ground truth annotations since there is no line information for these data.

The next row shows what would happen if we somehow knew which key transposition and redundancy of an e-chords file is best to use, since in this scenario we use the key and redundancy which has the best performance when compared to the ground truth. We see improvements for constraining the alphabet and transitions, a decrease in performance for using normal alignment, and finally a significant improvement for using JA. This last result shows that the JA method overcomes the segmentation issues presented in e-chords.

The last row shows when the only measure to help decide which key and redundancy is best to use is the largest likelihood score, and we notice a similar pattern to the second row, albeit slightly lower in all cases. This result shows that the likelihood is a good proxy for the performance accuracy, i.e. we can automatically choose the best key and redundancy of an e-chords UCS. As this approach is the more realistic one in practice, in all further experiments we make use of the ‘best-guess’ approach, selecting the transposition and version of e-chords using the log-likelihood.

5.2 Experiments using e-chords for training

This subsection details the experiments for using UCSs as additional training data. There are three scenarios, ranging from the hypothetical cases to the realistic case.

- **Using Ground Truth Annotations as an Expansion Set.**
  In this scenario, we used increasing amounts of ground truth annotations as an expansion set. This models the hypothetical situation that a large amount of training data similar to the test set has been produced.

- **Using Aligned Ground Truth Annotations as an Expansion Set** (i.e. UCSs are the ground truth annotations without timing information, re-aligned using UCSA). This scenario models a noise-free e-chords dataset, since the chords will be correct and only the timing information remains to be inferred. These results then show the upper bound on our approach.

- **Using e-chords UCSs as the expansion set.** This is the realistic case of using online chord databases for training. The aim of this scenario was to see if we could use aligned e-chords sequences as additional training data to enhance chord recognition.

We now specify the experiments set up for exploring the effectiveness of the Hard-EM learning scheme, where the core training set will be fixed as the Queen data, the test set will be a random sample of \( \frac{3}{5} \) of the Beatles songs, and the expansion set will choose between a random sample of roughly \( \frac{1}{4} \) of the Beatles and the

---

7The following five were missing: Love You To, Within You Without You, Wild Honey Pie, Revolution 1, I Want You (She’s So Heavy).

8The aligned ground truth annotations are generated by stripping the timing information away from ground truth annotations and re-aligning them using UCSA.

9To make the results in Section 5.2.2 comparable to that in Section 5.2.3, we only used 52 Beatles songs as the expansion set, which is roughly \( \frac{1}{4} \) of the total data size.
whole Oasis data. All experiments are repeated 40 times to assess variance.

5.2.1 The number of iterations for Hard-EM

To find out how the performance improves over the Hard-EM iterations, we used $\frac{1}{3}$ of the Beatles as the expansion set and completed five iterations of hard EM. The results are shown in Figure 7.

Figure 7 shows two constant lines which are independent of the iteration number: the upper bound is when we use the ground truth annotations as the expansion set, the lower bound is when there is no expansion set (i.e. only using the core training set). From a baseline performance of 70.83% we gain an initial improvement to 73.62%/74.24% using Beatles UCSs/aligned ground truth annotations. These results are after one iteration in Hard-EM. After this both performances increase only slightly until convergence: 0.47%/0.26% for the UCSs/aligned ground truth annotations compared to an initial improvement of 2.80%/3.42%. Thus, we can save a significant amount of computation time at a small performance cost by just doing one iteration of Hard-EM, and we will do this in all future experiments (i.e. no realignment process in Figure 6). We also show error bars of width one standard deviation in Figure 6; the fact that they overlap confirms that the increase in performance by using more than one iteration of Hard-EM may not result in a significant improvement in recognition performance.

5.2.2 Using The Beatles as the expansion set

This experiment is a truly realistic setting where we are aligning an artist for which no songs are present in the training set. The aim was to see if we could use aligned e-chords sequences as additional training data for testing on the same artist, based on fully labelled training data from a different artist. To investigate how the size of the expansion set affected performance, we increased it from 0 to 52 in multiples of 10 (the last set had two songs).

The flow chart of the experiment is illustrated in Figure 8 and the results are indicated by the solid, dashed and dotted lines in Figure 9. It is encouraging to see that using e-chords UCSs follows the same pattern as the other two hypothetical cases. These are important and

![Flow chart for Section 5.2.2.](image)

![Results for Section 5.2.](image)

Fig. 7. The results of five iterations of Hard-EM. The horizontal axis shows iterations of Equation 10, the vertical axis performance on a held-out Beatles test set, with error bars of 1 standard deviation.

Fig. 8. Flow chart for Section 5.2.2.

Fig. 9. Results for Section 5.2. The expansion set is taken either as The Beatles ground truth (solid line), The Beatles aligned ground truth (dashed line), The Beatles best guess paths (dotted line), Oasis best guess paths (solid line, triangle markers), or Oasis best guess paths, expanding on the hidden chain only (dot-and-dashed line). Error bars of width three standard deviations are included, which correspond to a confidence interval of 0.3%.
encouraging results, since they mean that we are able to use the e-chords database to enhance chord recognition on artists for which the ground truth chord annotation does not yet exist.

5.2.3 Using Oasis as the expansion set

The goal of this experiment was to investigate the benefit of complementing a small set of hand annotations with UCSs for an artist different from the artist of the test songs. The experimental setup is identical to Section 5.2.2 (see Figure 10). The evaluation is again done on songs of The Beatles.

The first results are denoted by the solid line with triangle markers in Figure 9, where we observed that using aligned Oasis songs as an expansion set does not help chord recognition on The Beatles. We postulate that this is because the observation probabilities are different due to different instrumentation, modern recording techniques etc. In fact, the relatively simple emission probability model and 12-dimensional chroma representation may be too simple to allow a good fit for songs with instrumentations and sound as different as Oasis is from The Beatles.

To investigate this hypothesis, we updated the parameters $P_{tr}$ and $P_{ini}$ only during the parameter refinement, but kept $\mu_r$ and $\Sigma_r$ as learned from the Queen labels. The results are shown in the dot-and-dashed lines and display an improvement in recognition accuracy. This confirms that the hidden chain of the HMM is similar for The Beatles and Oasis, but that the emissions differ. The musical intuition behind these results is that Oasis have exploited similar chord progressions as The Beatles, but used with different instrumentation and/or studio techniques. Note that while the performance on The Beatles decreases, we believe it may actually go up for Oasis-sounding bands.

To verify this, we manually annotated a small collection of five Oasis songs from our dataset to test on and used the remaining 47 Oasis songs as an expansion set, with the core training set remaining Queen. The results showed an increase in recognition from 43.88% to 48.15% with a $P$-value (paired T-test) of 0.0835. Note that the power of this test is limited by the small number of annotated songs, such that this $P$-value is as small as one could expect it to be.

6. Conclusions

We have demonstrated that using online chord databases can improve chord recognition in a number of ways. Firstly, we showed how to constrain the Viterbi decoding of an HMM-style chord recognition system to bias the chord sequence output. We explored various approaches to achieve this goal, most of which achieved significant improvements, the best one of which is the Jump Alignment method with an absolute improvement of 4.73%. These results are based on choosing the best key and redundancy by only consulting the likelihood of the alignment, meaning that knowledge of the correct key is not needed for the method to be applicable.

Furthermore, we showed how e-chord style UCSs can be used as additional data for training, and these new datasets universally improve chord recognition when the expansion set contains songs similar to those to which the chord recognition system needs to be applied. When we tried to expand using a different artist, we experienced a small decrease in recognition accuracy, which we postulated was because the emission probabilities were different, as corroborated by a small experiment with newly annotated songs.

In future work, we plan to use the presented methods on a large scale, exploiting as much of e-chords (or similar websites) as possible. To make this happen, it may be necessary to choose specific songs for the expansion set when the goal is to recognize the chords for a certain artist. Also, we intend to develop automatic ways to assess the quality of an e-chords annotation, so that the more noisy ones can be discarded.

Perhaps the most promising avenue for further research is investigating if the newfound ability to exploit these vast amounts of data will allow us to increase the complexity of models and features used, without the risk of overfitting. Such a model could be more versatile in terms of genres, such that the problem of decreasing performance when the expansion set is too different from

Fig. 10. Flow chart for Section 5.2.3.

10The songs were Bring it on Down, Cigarettes and Alcohol, Don’t Look Back in Anger, Morning Glory, and My Big Mouth. We plan to make the annotations available online shortly.
the test songs might even disappear. Indeed, we believe that the current simple emission probability model for chords, in combination with the limited chromagram feature representation, is too simplistic to model various genres simultaneously.

Acknowledgements

This work is supported by the EPSRC project EP/G056447/1 and by the PASCAL2 European Network of Excellence and also partially by EPSRC grant number EP/E501214/1. We are grateful to Matthias Mauch, Emilio Parrado-Hernandez, and Sandor Szemek for useful feedback on an earlier draft of this paper and for interesting discussions.

References


