Enhancing Chord Recognition Accuracy Using Web Resources

Matt McVicar
Intelligent Systems Laboratory
University of Bristol, UK
Matt.McVicar@bristol.ac.uk

Tijl De Bie
Intelligent Systems Laboratory
University of Bristol, UK
Tijl.DeBie@bristol.ac.uk

ABSTRACT

Machine learning methods for chord recognition have improved considerably in the past few years. However, further progress seems constrained by the scarcity of training data. In this paper, we show that this problem can be partially solved by exploiting noisy but freely and abundantly available online resources, in addition to fully labeled training data. We use these data to restrict the output of the Viterbi algorithm, resulting in significant improvements over the standard decoding process.

Categories and Subject Descriptors

H.5.5 [**Information Interfaces and Presentation**]: Sound and Music Computing—*Methodologies and techniques*

General Terms

Algorithms, Experimentation

Keywords

Chord recognition, Music information retrieval, Alignment

1. INTRODUCTION

The task of automatically identifying chords from music audio has attracted significant research efforts in the past decade. Contributions have focused on the design of better audio features, utilizing rhythmic information, and deploying more sophisticated machine learning methods.

Chromagram feature representations have been improved by attempts to suppress harmonics [6], by tuning songs [4], and by computing them using the constant-Q transform [1]. Furthermore, [8] have suggested to compute chromagram features only after splitting the audio into a percussive and a harmonic component, with excellent results. The fact that chord changes typically change in certain metric positions has been exploited as well [7]. Furthermore, many authors compute beat-matched chromagrams, the median of the chromagrams between each pair of consecutive detected beats in a song [2]. Lastly recently proposed methods for structured output prediction have been applied to the chord recognition task [9], as well as more advanced versions of the Hidden Markov Model known as Dynamic Bayesian Networks [7]. In this way, the performances of the best available chord recognition systems have now approached 80% when evaluated on an alphabet of the 12 major and minor chords and a no-chord symbol (see e.g. [6, 8]).

Despite this significant progress, there is still a wide gap between the performance of the best automatic method and the performance of trained musicians. Here, we explore an approach to help bridge the remaining gap by relying on auxiliary information that is noisy but freely and abundantly available on the web. In particular, we suggest to use chord annotations from websites such as `e-chords.com` maintained by guitar enthusiasts, where chord sequences are annotated above the lyrics, albeit without explicit information about the times of onset. For this reason, we will refer to such a chord sequence as an *Untimed Chord Sequence (UCS).*

We discuss three simple approaches exploiting this information to various degrees for enhanced chord recognition. Each of these approaches is ideally suited to be used in combination with a chord recognition system based on a Hidden Markov Model (HMM) and the Viterbi algorithm. In this paper, we improve the accuracy of such methods by constraining the result of the Viterbi algorithm using information contained in the UCS's such as those found on `e-chords.com`. The effectiveness of each of these three approaches is investigated by experiments on the Beatles data set provided by Chris Harte [3].

2. IMPROVING CHORD RECOGNITION USING E-CHORDS

2.1 UCS's and the e-chords website

We made a dedicated scraper to extract chords from `e-chords.com` and thus obtained UCS's for 75,863 songs. When investigating the scraped UCS's for Beatles songs (182 of 75,863), we discovered that several had two or three annotations, to which we shall refer as *redundant* UCS's. These might have been uploaded by a different user, or be set in a more guitar-friendly key. To counteract the fact that some UCS's may be in an incorrect key we will consider each UCS in all 12 transpositions.

2.2 Exploiting UCS's in chord detection

A standard approach for chord recognition is modeling

MML'10, October 25, 2010, Firenze, Italy.
Copyright 2010 ACM 978-1-4503-0161-9/10/10 ...$10.00.

the chromagram feature vectors as the observed chain of a Hidden Markov Model (HMM), with the hidden chain representing the chords. The HMM is traditionally trained on a set of fully annotated songs. The transition probabilities can be modeled by the multinomial distribution, and the emission probabilities using a Gaussian probability density.

Given an HMM, the likelihood of any chord labeling can be quantified efficiently. Furthermore, the Viterbi algorithm finds the most likely chord labeling efficiently, using Dynamic Programming (DP). We will compare our methods to this Viterbi baseline, to which we will refer as V.

The following three methods attempt to increase the accuracy of the decoding step, by employing the UCS's for each of the test songs in three different ways. However, note that training of the model parameters is still done on the (fully labelled) training set, as usual.

Method 1: Alphabet Constrained Viterbi (ACV).

As a first approach to exploit information from the UCS's, we restrict the output of the Viterbi algorithm to chords present in the UCS only. This method can be implemented simply by only filling up a part of the DP table used in the Viterbi algorithm.

Method 2: Alphabet & Transition Constrained Viterbi (ATCV).

A second approach extends the first by also constraining the allowed chord transitions to those seen in the UCS. This can be implemented conveniently e.g. by setting the probabilities of disallowed transitions equal to zero.

Method 3: UCS to Audio Alignment (UCSA).

A third approach is to force the chord labeling to respect the complete order in which the chords occur in the UCS, and infer the duration of each of the chords. Computing such a chord labeling compatible with the ordering in a UCS can be achieved by *aligning* the UCS with the audio.

This can be done efficiently using a DP algorithm. It builds a DP table of size $n_u \times n_s$, in which the rows correspond to the n_u chords in the UCS, and the columns correspond to the n_s consecutive chromagram feature vectors. The table is filled in such that entry at position (i, j) contains the likelihood of the optimal alignment of the length i prefix of the chord sequence with the first j chromagram feature vectors. We give the details without proof of correctness due to space restrictions.

Let $u(i)$ be the i'th chord in the UCS and $f(j)$ the j'th chromagram feature vector in the song. With $P_e(f|c)$ the emission probability of chromagram feature vector f given chord c and $P_t(c_2|c_1)$ the transition probability from chord c_1 to chord c_2 the algorithm then proceeds as follows:

$$\textbf{Initializition:} \quad \begin{aligned} DP(1,1) &= P_e(f(1)|u(1))), \\ DP(i,1) &= 0,\ \forall i > 1, \\ DP(1,j) &= 0,\ \forall j > 1. \end{aligned}$$

For $i = 1 : n_u$ **and For** $j = 1 : n_s$**:**

$$\begin{aligned} & DP(i+1, j+1) \\ = \ & P_e(f(j+1)|u(i+1))) \\ & \times \max \left\{ \begin{array}{l} DP(i,j)P_t(u(i+1)|u(i)), \\ DP(i+1,j)P_t(u(i+1)|u(i+1)). \end{array} \right. \end{aligned}$$

The likelihood of the optimal alignment is found in $DP(n_u, n_s)$. The actual chord labeling can be found by maintaining a separate trace-back matrix that keeps track of which of the two arguments lead to the maximum in the DP iteration.

2.3 Dealing with redundancies

Given several redundant UCS's and their transpositions, it is a priori unclear which of these to use in the above three methods. Ideally, the best variant in terms of chord prediction accuracy is used, but of course this is unknown if the ground truth is not given. Instead, we suggest to use the likelihood of the predicted chord sequence as a proxy. We will investigate the effectiveness of this proxy in Sec. 3.

3. EMPIRICAL RESULTS

3.1 Data and setup

We evaluated our approaches on the Beatles data, for which detailed and accurate chord annotations have been made available [3]. Our goal was to evaluate the effectiveness of each of the three methods discussed in Sec. 2.2. Furthermore, we wished to assess whether the amount of noise in data drawn from `e-chords.com` is sufficiently low to remain useful. Lastly, we wanted to investigate whether redundant annotations can be used to improve results.

To achieve these goals, we split the set of 180 Beatles songs into a test set of 26 songs for which two or three UCS's were available from `e-chords.com`. The ground truth annotations for remaining 154 songs were used for training the HMM parameters.

3.2 Details of the baseline method, and evaluation metric

As mentioned, our methods operate on top of an HMM method, that we will also use as a baseline for comparison. Our chromagram feature vectors were computed by first splitting the harmonic and percussive components of the audio as suggested by [8], after downsampling to 11025 Hz. To compensate for some songs not being in standard tunings we estimated the discrepancy between the song tunings and standard pitch ($A4 = 440Hz$) using Dan Ellis' method (`http://labrosa.ee.columbia.edu/`). We then calculated chromagram feature representations using the MIRToolbox [5], using a Hamming window length of 4096 frames and hop length 1024 frames and wrapped the resulting power spectrum to one octave. The chromagram feature vectors were then normalised to unit infinity norm.

We restricted the chord alphabet to 12 major chords, 12 minor chords and a no-chord symbol. This was achieved by binning all chords with a major third into the major chord class, and analogously for minor chords. Chords which featured no third were classified as major and any chords which obviously scraped incorrectly from e-chords were discarded.

In all cases the MIREX-style evaluation of performance was used, which corresponds to the total number of correctly predicted frames divided by the total number of frames.

3.3 Results

In order to benchmark the potential for this method in an idealized setting we modelled the situation where the UCS are noise-free. We did this by setting the UCS's to be the ground truth devoid of repetitions. We then ran the four experiments V, ACV, ATCV, and UCSA, the results of which

Table 1: Performances for each of our three methods. P-values are shown between brackets.

	Viterbi	ACV	ATCV	UCSA
Using the true UCS from the ground truth				
True UCS	77.03%	81.18% (6e-6)	84.53% (4e-6)	88.00% (4e-6)
Using two or three redundant `e-chords.com` annotations				
Best-Accuracy UCS	77.03%	80.41% (5e-5)	81.96% (1e-4)	71.60% (0.92)
Best-Likelihood UCS	77.03%	79.65% (3e-4)	81.63% (2e-4)	71.60% (0.92)
Using only one `e-chords.com` annotation				
Best-Accuracy UCS	77.03%	79.38% (3e-3)	80.60% (2e-3)	69.44% (0.99)
Best-Likelihood UCS	77.03%	78.86% (1e-2)	80.33% (3e-3)	68.94% (0.99)

can be seen as the first row of Table 3.3. We see a baseline performance of 77%, followed by steady and significant improvements to a maximum of 88% by restricting the alphabet and transitions and aligning.

We then turned our attention to the more realistic setting where UCS scraped from `e-chords.com` are used, thereby assessing whether the noise therein is sufficiently low to still be of use. As mentioned in Sec. 2.3, we choose either the best version of the UCS in terms of prediction accuracy (considering all versions found on `e-chords.com` in all 12 transpositions), or the version with the highest likelihood. The first of these we call the Best-Accuracy UCS and the latter the Best-Likelihood UCS. The Best-Likelihood UCS is more realistic, since in practice the prediction accuracy is unknown and thus the Best-Accuracy UCS is unknown. However, the difference between the Best-Likelihood and the Best-Accuracy UCS performances indicates how good the likelihood is as a proxy for the prediction accuracy.

These results are summarised in rows 3 and 4 of Table 1. We see that the Best-Accuracy UCS only marginally outperforms our Best-Likelihood UCS in the ACV and ATCV experiments, and that both outperform the unconstrained Viterbi algorithm. The fact that the Best-Accuracy UCS only slightly outperforms the Best-Likelihood UCS indicates we can predict the best redundancy and key transposition without knowledge of the ground truth fairly well.

When we aligned the `e-chords.com` UCS's to the audio, performance dropped in both cases. Upon investigation this was because although the sequences shared many chords with the ground truth, the ordering was not accurate enough to achieve a good performance. One reason for this is that some annotations neglect to include repetition information (*i.e.* 'Play verse chords twice'), which were not understood by our scraper. This tells us that the quality of `e-chords.com` data is such that it offers an improvement on cutting-edge techniques by constraining the Viterbi algorithm, but not so much that we can rely on it to simply alignment the UCS's to audio.

The last two rows of table 1 show the performance that is achieved if only one randomly chosen `e-chords.com` file is considered in all 12 transpositions, rather than two or three alternative versions. This simulates the scenario that will hold for most songs, where only one version is available. The performance drops slightly, but remains considerably above the baseline Viterbi performance.

Table 1 also shows p-values between brackets (computed using the signed rank test), showing that each performance increase is highly significant.

Figure 1 shows these performance changes on a song-by-song basis. In these experiments, all two or three alternative `e-chords.com` versions for each song were used. Clearly, the lower performance when using `e-chords.com` data as compared to the true UCS's is due to a few songs only, for which the `e-chords.com` UCS is of low quality.

4. DISCUSSION AND CONCLUSIONS

In this paper, we have attempted to simplify the chord recognition problem by drawing in additional information that is freely and abundantly available on the web. We have shown there is significant potential for this approach, and performance can be improved by about 3% when no redundancy is available, and nearly 5% when there is redundancy, at no extra computational cost. Furthermore, there is a potential for an improvement of 11% with high quality UCS data, the generation of which is still much less labour-intensive than full chord annotations.

Our methods can directly be used to leverage the data available from `e-chords.com`, totaling to $75,863$ UCS's (counting duplicates) at the time we scraped it. Our results show that it is wasteful to ignore this data when a song is available on such websites, despite the potentially low quality of these annotations. Even if they are noisy, using the ACV or ATCV methods will typically lead to a higher quality chord labeling than can be achieved without.

Furthermore, our methods can be used in a manner not unlike image segmentation methods that make use of additional easy-to-provide information of the approximate location of the edges sought. They allow a user to specify just the chords present in a song, perhaps also the transitions, or if possible the UCS, without going through the laborious process of accurately annotating chord onset times. Then the approaches suggested in this paper can be used to complete the task.

We believe this paper may open up another line of research, on the use of lower-quality but abundant annotations such as those from `e-chords.com` in addition to the high-quality but scarce annotations made by hand. While new machine learning methods to utilize this redundant, noisy, and incomplete source of label information will need to be developed to make this happen, the current paper shows such approaches may have significant potential. This would resolve possibly the most important bottleneck in chord recognition research relying on machine learning techniques.

5. ACKNOWLEDGMENTS

Matt McVicar is funded by EPSRC through the Bristol Centre for Complexity Sciences. This work is partially supported by the EPSRC project EP/G056447/1.

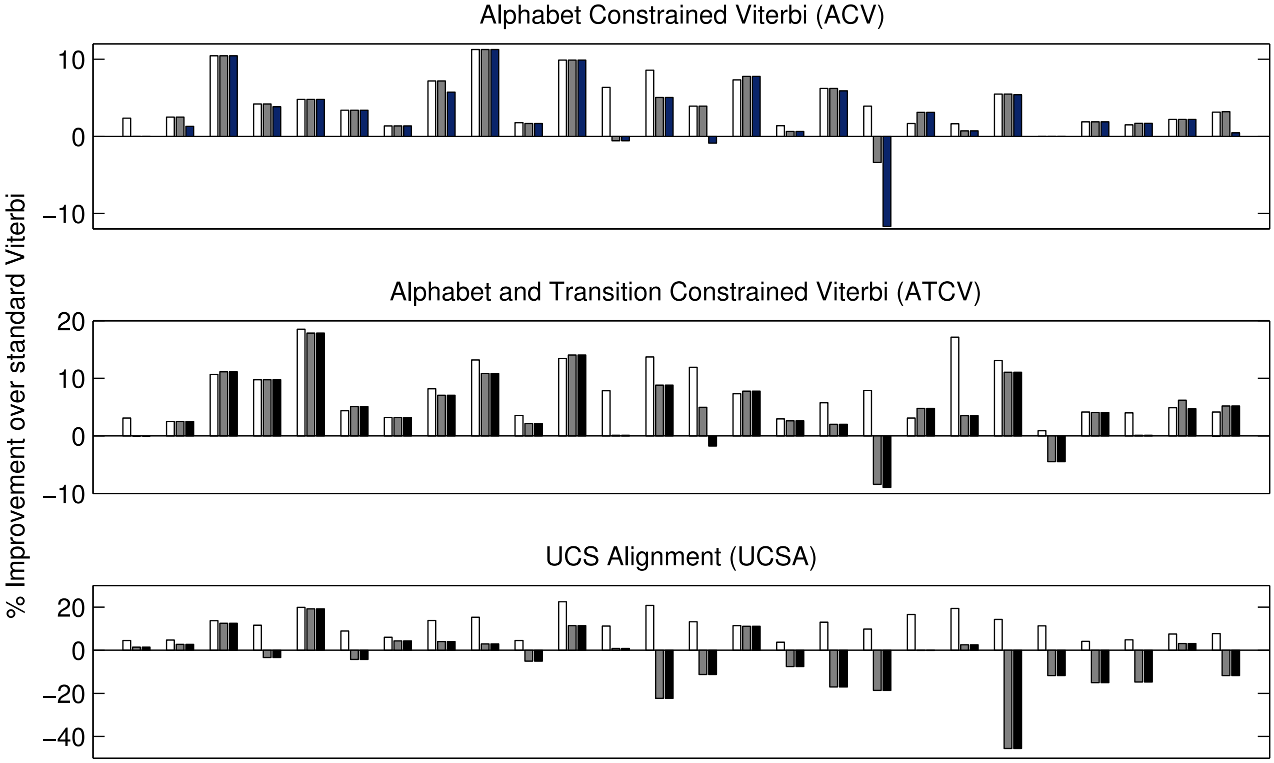

Figure 1: The increase in accuracy for 26 Beatles songs, for each of the three proposed ways of exploiting UCS's (ACV, ATCV, and UCSA). For each song, three performance differences are shown: using the correct UCS derived from the ground truth annotations (left bars in each triplet), using the Best-Accuracy UCS from `e-chords.com` (middle bars), and using the Best-Likelihood UCS from `e-chords.com` (right bars).

6. REFERENCES

[1] J. C. Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–Ű434, 1990.

[2] D. P. W. Ellis and G. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, volume 4, 2007.

[3] C. Harte, M. Sandler, S. Abdallah, and E. Gomez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR 2005)*, 2005.

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

[5] O. Lartillot and P. Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International Conference on Digital Audio Effects, Bordeaux*, 2007.

[6] M. Mauch and S. Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[7] M. Mauch and S. Dixon. Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 2010. accepted.

[8] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama. Hmm-based approach for automatic chord detection using refined acoustic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP*, 2010.

[9] A. Weller, D. Ellis, and T. Jebara. Structured prediction models for chord transcription of music audio. *Machine Learning and Applications, Fourth International Conference on*, 0:590–595, 2009.