LEVERAGING NOISY ONLINE DATABASES FOR USE IN CHORD RECOGNITION

Matt McVicar, Yizhao Ni, Tijl De Bie
Intelligent Systems Lab
University of Bristol
matt.mcvicar@bris.ac.uk
{yizhao.ni, tijl.debie}@gmail.com

Raul Santos-Rodriguez
University Carlos III of Madrid
rsrodriguez@tsc.uc3m.es

ABSTRACT

The most significant problem faced by Machine Learning-based chord recognition systems is arguably the lack of high-quality training examples. In this paper, we address this problem by leveraging the availability of chord annotations from guitarist websites. We show that such annotations can be used as partial supervision of a semi-supervised chord recognition method—*partial* since accurate timing information is lacking. A particular challenge in the exploitation of these data is their low quality, potentially even leading to a performance degradation if used directly. We demonstrate however that a curriculum learning strategy can be used to automatically rank annotations according to their potential for improving the performance. Using this strategy, our experiments show a modest improvement for a simple major/minor chord alphabet, but a highly significant improvement for a much larger chord alphabet.

1. INTRODUCTION

Chords are musical features which compactly describe the harmonic content of Western music. They have been used to successfully identify keys [17], cover songs [2] and genres [1], confirming their use in understanding and analysing musical harmony, underscoring the importance of systems able to recognize chords from music audio. An important aspect of the chord recognition problem is the limited amount of high-quality audio annotations on which to train machine learning systems, currently limited to 218 songs by The Beatles, Queen and Zweieck. [1] The result is that the performance of machine learning systems for chord recognition are starting to stagnate at around 80% in the MIREX evaluation metric for an alphabet of major and minor chords only.

In this paper, we propose a system that complements the valuable available data with annotations found in large online chord databases. In particular, here we make use of the chord database e-chords.com [2], a guitarist website containing approximately 140,000 partially labelled chord annotations. Exploiting this data is non-trivial though: it does not contain timing information, and the quality of the annotations is highly variable.

The proof-of-concept that such information can be exploited in a semi-supervised learning setting has already been provided in a very small-scale study [15]. Unfortunately, it turns out that after scaling this up to more data this approach by itself is insufficiently robust to overcome the quality issues with the online annotations. In the current paper, we therefore adopt a *curriculum learning* approach, which attempts to add 'easy' data points first and 'hard' ones only later (if at all). To quantify 'easiness', we also introduce a new metric to evaluate chord recognition performance when no ground truth annotation is available, but an online annotation is. This new metric by itself is a valuable contribution, as it allows one to evaluate chord recognition systems on artists other than The Beatles, Queen and Zweieck.

2. PRELIMINARIES

In this section we describe our overall approach to chord recognition, the audio features we make use of, as well as the data we were able to extract from e-chords.

2.1 Model Architecture

As a baseline system, we make use of a Hidden Markov Model (HMM), which has been used extensively and successfully for chord recognition [7, 17]. Here, the hidden chain represents the sequence of chords in a sequence of time *frames* the song is segmented in. Assuming that chords rarely change between beats, we chose our frames to be

[1] available at http://isophonics.net/

[2] www.e-chords.com

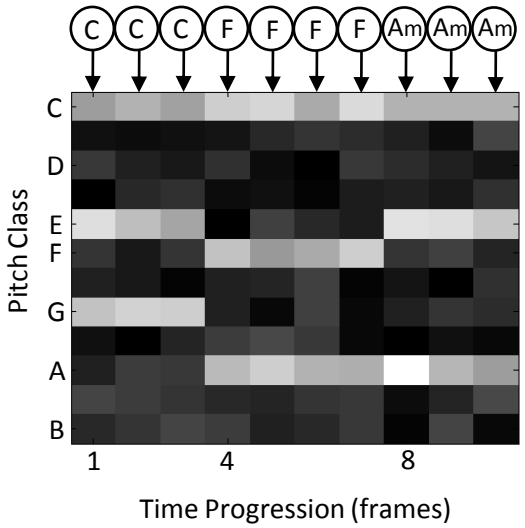

Figure 1. The HMM topology of our model, showing the hidden nodes of the HMM (chords) emitting 12-dimensional feature vectors (chromagrams).

the time periods between consecutive beats as estimated using BeatRoot [6]. The observed chain corresponds to 12-dimensional *chromagram* feature vectors [6, 12] in the corresponding frames. The chromagram represents the distribution of energy across pitch classes of the harmonic content of the audio. The model is depicted in Fig. 2.1.

2.2 Feature Extraction: the Loudness-Based Chromagram

There is no single method to compute a chromagram feature vector, but the most popular ones are based on the Fourier and constant-Q transforms [4, 9, 11]. In this paper we will employ a newly proposed variant, called the *loudness-based chromagram* [16]. The salient feature of this chromagram is that it is closer to how humans perceive the strength of pitches. Similar to existing variants, the loudness chroma extraction process outputs a matrix $\mathbf{C} \in \mathbb{R}^{12\times T}$ from a monaural signal $\mathbf{x}$, where T is the length of the feature in number of frames.

2.3 Ground Truth Extraction

For each song for which a ground truth is available, we constructed the chromagram $\mathbf{C} \in \mathbb{R}^{12\times T}$ feature vector, where T is the number of (estimated) beats. This is complemented with a corresponding chord annotation $\mathbf{A} \in \mathcal{A}^T$ extracted from the ground truth annotations, where $\mathcal{A}$ is a chord alphabet set. The fully annotated songs from The Beatles, Queen and Zweieck thus make for three sets of training data, denoted as $\{\mathbf{C}_\mathrm{B}, \mathbf{GT}_\mathrm{B}\}$, $\{\mathbf{C}_\mathrm{Q}, \mathbf{GT}_\mathrm{Q}\}$ and $\{\mathbf{C}_\mathrm{Z}, \mathbf{GT}_\mathrm{Z}\}$.

C F
You know I love you
C G7
And I'll prove my love is true
Dm9 G11
Want to show how I feel

Hoping that you love me
'Cos it's plain to see
That our love is real

Figure 2. Example Untimed Chord Sequence (UCS) for 'Our love is real' (Matt McVicar), showing chord labels above lyrics.

2.4 E-chords extraction

As in [15], we extracted Untimed Chord Sequences (UCSs) from the chord database e-chords.com. These UCS are referred to as 'untimed' as they only contain (noisy) information about the ordering of the chords, with no additional information on exact timing. From the e-chords website we were able to scrape over 140,000 such UCSs, but we could only use those for which we had access to the audio as well. We combined our personal music collections and found the overlap with the UCS database to be 2008 tracks. Note that although it is unfortunate that we were only able to extract a small proportion of UCSs from the database (2008), this number is significantly larger than the number of currently available training examples (218).

We calculated a loudness-based chromagram for each of these 2008 songs in the echords dataset and refer to the e-chords chromagram/UCS set as $\{\mathbf{C}_\mathrm{EC}, \mathbf{UCS}\}$.

3. EXPLOITING UCS'S AS PARTIAL SUPERVISION DURING TRAINING

The UCSs clearly provide information about the true chords in an audio file, albeit only partial information. They convey information on the chords of many songs, but unfortunately the explicit timings of the sequences are not known. Making use of unlabelled (or partially labelled) data together with labelled data for training is known as *Semi-Supervised Learning* (SSL) [5].

3.1 The semi-supervised learning approach

The general approach of exploiting UCSs during training was introduced in [15], and we briefly summarize it here. The approach works by initially training the chord recognition system (the HMM) based on the fully labelled training data, here called the Core Training Set (CTS).

Subsequently, it attempts to reconstruct the timings of the UCSs by aligning them to the chromagram feature vectors

extracted from the corresponding audio. An example UCS is shown in Figure 2. The first six chords are to be repeated, although it is hard to infer this automatically without prior knowledge of the song. Unfortunately, this source of 'structural noise' is hard to capture using automatic methods to scrape UCSs from websites, so we would miss this information.

To overcome this, the Jump Alignment (JA) algorithm (see [15]) can be used. The JA algorithm is able to align UCSs to audio, while allowing for jumps to the start of other lines (e.g. to allow a section to be repeated). The probabilities of jumping forward or back in an annotation, as well as the key transposition and version are all chosen by maximum likelihood. A different approach to dealing with structural noise in online annotations has recently been proposed by the authors of [13], which could be combined with our alignment method to yield further improvements.

After aligning our UCSs to their audio, they are in the form of fully labeled training data and can be added to the CTS. We refer to the resulting set of annotated data as the Expanded Training Set (ETS). Finally, the chord recognition system can be retrained based on the ETS. The hope is that this approach will allow one to train a chord recognition system to be able to recognize chords in genres that are different from those for which fully annotated chord sequences are available.

3.2 Evaluation setup in this paper

This approach was introduced and tested on a small scale in [15], involving only songs for which a ground truth annotation is available. In this paper we test this approach on a significantly larger scale. In particular, as CTS, we use the Queen and Zweieck songs:

$$\mathbf{CTS} = \{\bigcup\{\mathbf{C}_{\mathrm{Q}}, \mathbf{C}_{\mathrm{Z}}\}, \bigcup\{\mathbf{GT}_{\mathrm{Q}}, \mathbf{GT}_{\mathrm{Z}}\}\}$$

The ETS is the union of the CTS and the set of 2008 songs for which we have the audio and a UCS from e-chords:

$$\mathbf{ETS} = \{\bigcup(\mathbf{C}_{\mathrm{Q}}, \mathbf{C}_{\mathrm{Z}}, \mathbf{C}_{\mathrm{EC}}), \bigcup(\mathbf{GT}_{\mathrm{Q}}, \mathbf{GT}_{\mathrm{Z}}, \mathbf{AUCS})\}$$

The test set consists of all The Beatles songs and their ground truth annotations.

The flow-chart of this set-up is shown in Fig. 3, which also shows the parameters that are inferred at various stages (the HMM initial and transition probability matrices $\mathbf{I}$ and $\mathbf{T}$, as well as the mean and covariance matrices for the Gaussian output probability densities, μ and Σ). After retraining based on the ETS, they are referred to as $\mathbf{I}'$, $\mathbf{T}'$, μ' and Σ'.

As the results in Sec. 5 show, unfortunately in this setting this basic approach deteriorates performance, rather than improving it. To resolve this issue, here we propose to additionally adopt a curriculum learning approach.

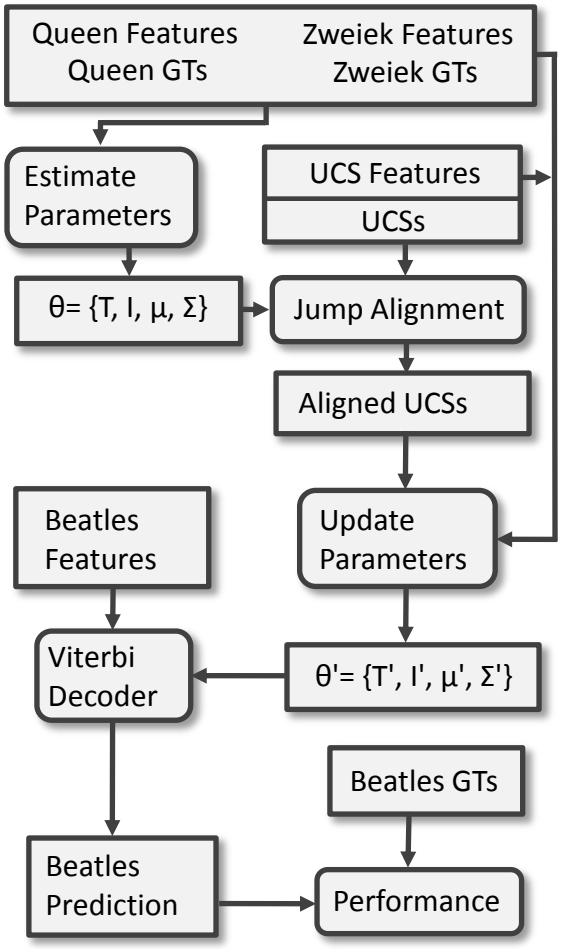

Figure 3. The schematic of our experiments. Data are shown in square boxes, processes in curved. Detailed descriptions of the processes are found in the text.

4. CURRICULUM LEARNING

In this section, we describe an addition to the scheme in Figure 3 which makes the most of the available data using curriculum learning. We also outline our new evaluation method. We begin with some background information on the subject.

4.1 Background

It has been shown that humans and animals learn more efficiently when training examples are presented in a meaningful way, rather than in a homogeneous manner [8, 10]. Exploiting this feature of learners is referred to as *Shaping* in the animal training community and *Curriculum Learning* in the machine learning discipline [3].

The concept of the curriculum paradigm is that starting with *easy* examples and slowly generalising leads to more efficient learning, which can be realised in a machine learn-

ing setting by carefully selecting training data from a large set of examples. It was recently hypothesised that curriculum learning offers faster training (in both optimization and statistical terms) in online training settings, owing to the way the learner wastes less time with noisy or harder to predict examples, and that additionally guiding the training into a desirable parameter space will lead to greater generalization [3].

We introduce an additional step into Figure 3 to deal with curriculum learning in a novel way. Note that up to now we have not defined what we understand by easy examples, or equivalently, how to sort the available examples into a series of increasing difficulty samples. Therefore, after the UCSs have been aligned to the features, we will attempt to sort the expansion set by appropriateness for learning. We propose a new measure for evaluating how accurate the set **AUCS** compared to its (unknown) ground truth annotations.

Thus we have the two following assumptions:

1. Introducing 'easy' examples into the training set leads to faster learning.
2. It is possible to estimate which training examples from a varied set are 'easiest'.

We will address these assumptions in the following subsection.

4.2 Alignment Quality Proxy

When we created the **ETS**, we were unable to evaluate how well the UCSs aligned to the loudness-based chromagrams, since the ground truths are not available for these songs. However, we were able to estimate the accuracy of the alignment in a different way.

To begin with, we noticed that many alignments contained only a few chords and were therefore extremely unlikely to be accurate chord alignments. We therefore removed all alignments which contained fewer than 5 unique chords.

After this pruning, we looked into a quantitative estimate for the alignment quality. An output of the JA algorithm is the log-likelihood of UCS correctly aligning to the loudness chroma. For each UCS $\in$ **AUCS** we used the log-likelihood of the alignment normalised by the length of the alignment as a proxy for the performance, and stored these in the alignment quality proxy vector P_{aqp}:

$$P^i_{aqp} = \frac{\text{log-likelihood of } AUCS_i}{|AUCS_i|}, i = 1 \ldots |\textbf{AUCS}|$$

The results of the Alignment Quality Proxy performances on our songs are displayed as a histogram in Figure 4. There is a range from -1.79 (very poor alignment) to 7.03 (excellent alignment), and we notice a skew towards good quality alignments.

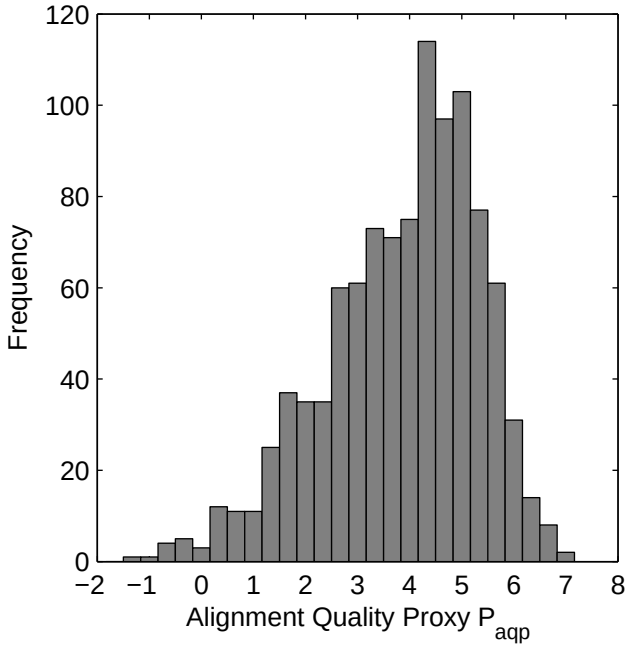

Figure 4. Histogram of our proposed alignment quality measure.

We then sorted the **ETS** with respect to P_{aqp} and segmented the set into bands according to alignment performance. In order to investigate the quality of the proposed alignment performance we ran JA on 173 Beatles songs for which we had UCSs, with the alignment parameters from Queen and Zweieck, yielding P^B_{aqp}. We also used these parameters to make an HMM prediction for each of the 173 songs and measured the performance P^B of these predictions against the Beatles ground truth sequences.

Finally, we measured the correlation between the P^B_{aqp} and P^B using Pearson's linear correlation coefficient, which gave a correlation of 0.73 with a p-value of 0.4×10^{-30}, indicating a highly significant result at the 5% level ($p < 0.05$). This result indicates that P_{aqp} is an excellent proxy for alignment accuracy, i.e. we have answered assumption 2 in Subsection 4.1 in the affirmative.

Satisfied that P_{aqp} offers an approximation of how well JA aligns UCSs, we decreased the size of the ETS by placing a threshold on the alignment quality. Mathematically, we allowed the i^{th} chromagram and aligned UCS pair $\{\mathbf{C}^i, \textbf{AUCS}^i\}$ into the training set if

$$P^i_{aqp} \geq \gamma$$

for $\gamma \in \mathbb{R}$. The value $\gamma = -\infty$ corresponds to being carefree with our data - all training examples are included. If

we wish to be stringent with our data, selecting a large γ will only allow high-quality alignments into the training set, although we may suffer from lack of examples in this scenario.

5. EXPERIMENTS

5.1 Simple Chord Prediction

In our first experiment we set the alphabet $\mathcal{A}$ to consist of major and minor chords, along with a 'No Chord' symbol. We refer to this alphabet as $minmaj$. All chords in the Core Training Set **CTS** and Expanded Training Set **ETS** were mapped to minor chords if they contained a minor third, otherwise they were mapped to the corresponding major chord. 'No Chord' symbols were added to the beginning and end of each of the Untimed Chord Sequences in **UCS** to account for the silences at the beginning and end of the pieces.

To re-iterate, we trained an HMM on the **ETS** and tested on all 180 Beatles songs. Performance was measured by number of correctly identified frames divided by the number of frames ($\times 100\%$), averaged over the 180 songs, and are shown in Table 1.

The results seen in Table 1 seem initially discouraging. The peak performance of 77.87% obtained using the 1021 best UCSs (in terms of alignment performance) only achieved an increase of 0.84%. However, upon performing a one-sided t-test of the performance of the system against the baseline performances (no expansion set), we obtained a p-value of 0.0435, indicating significance at the 5% level.

Using additional data in a system which is already performing well is unlikely to offer a large performance increase, since there is not much to be gained. On the contrary, when the difficulty of the task increases it is possible that extra data becomes beneficial. To investigate whether this is the case, we will increase the complexity of the model by using a larger library of chords.

5.2 Complex Chord Prediction

The results of subsection 5.1 showed that there is not much to be gained by using additional data sources on a simple chord model. To counteract this, we conducted the same experiments using an unrestricted chord alphabet $\mathcal{A} = full$. This meant that each unique chord in the Core and Expanded training sets were considered a unique state of our model, as well as the transpositions of each of these chords into each root pitch. This left us with 253 states, one order of magnitude larger than the major/minor chord alphabet.

As before we then retrained on the Expanded Set and tested on The Beatles. The results were measured as in Subsection 5.1. Figure 5 shows the results as well as the number of songs in the expansion set for each cut-off.

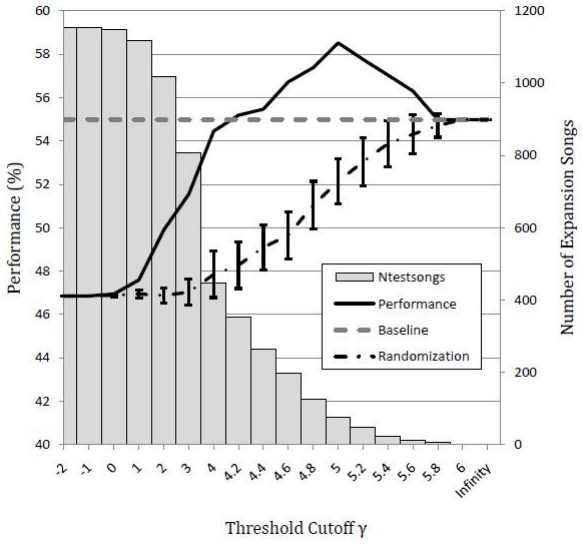

Figure 5. Performance of our model on The Beatles dataset with increasing alignment quality threshold quality γ. The baseline performance ($\gamma = \infty$) is shown as a dashed line. Values of γ for which the performance approaches or exceeds the baseline is shown in higher resolution steps of 0.2 increments. Randomizations of the same expansion set size are shown in the dot-and-dashed line.

Immediately from Figure 5 we see that blindly adding all of the available does not improve recognition. This is due to the large variety in style and genre seen in the database, along with the potentially poor alignments which we included in the expansion set when γ is small. Upon increasing γ we allowed heuristically better quality alignments into the training set, and saw a rapid increase in recognition accuracy, which peaks at 58.52%, 3.54% above the baseline of 54.98%. Although this increase may seem incremental, we performed a one-sided t-test of the performance of the system against the baseline at the optimal γ of 5 and found the p-value to be 1.28×10^{-7}, indicating a significant improvement at 5% confidence level. This corresponded to an improvement of 114 of the 180 songs.

To see if curriculum learning genuinely offered improvements over homogeneous learning, we also included aligned UCSs into the training set in random batches of the same size as the previous experiment, and repeated 100 times to account for random variations. The mean and standard deviations over the 100 repeats are shown as the dot-and-dashed line and bars in Figure 5. We can see that the specific ordering of the expansion set in section 4.2 offers substantial improvement over randomly selecting the expansion set. This is good evidence that curriculum learning is the method of choice for navigating a large set of training examples, and also demonstrates that assumption 1 in Subsection 4.1 holds.

Alignment Quality threshold γ	-2	-1	0	1	2	3	4	5	∞
Number of Expansion songs $\|\mathbf{AUCS}\|$	1027	1027	1021	993	899	705	390	67	0
Performance (%)	77.83	77.83	**77.87**	77.81	77.77	77.53	76.94	76.79	76.79
p-value of paired t-test	0.0516	0.0516	**0.0435**	0.0555	0.0561	0.1137	0.4779	0.6906	-

Table 1. Performance of our model on the simple chord alphabet, $\mathcal{A} = minmaj$. γ increases to the right, with the number of expansion songs this corresponds to underneath. Performances and corresponding p-values between the difference between the baseline level $\gamma = \infty$ are shown in the final two rows. Results which are significant at the 5% level are shown in bold.

6. CONCLUSIONS

In this paper we have made three breakthroughs. First of all, we demonstrated that chord databases can be used to create new sequences for training chord recognition algorithms. These sequences were shown to significantly improve recognition accuracy on an unseen test set.

Also, we demonstrated a new technique for estimating the quality of aligned chord sequences, which can be used to select training examples from a large, noisy training data set. This estimate allowed us to perform curriculum learning, which achieved faster learning and improved results.

Finally, we also showed that with more data we are able to make a more complex chord model, which led to a more significant improvement in recognition accuracy. In order to gain the most from these data we plan to further increase the complexity of the decoding model, by including distinct features for the bass and treble frequency range [14], including a hidden 'key chain' to model modulations [18] or using more complex emission probability models.

7. REFERENCES

[1] A. Anglade, R. Ramirez, and S. Dixon. Genre classification using harmony rules induced from automatic chord transcriptions. In *Proc. ISMIR*, 2009.

[2] J.P. Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proc. ISMIR*, pages 239–244, 2007.

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proc. ICML*, pages 41–48. ACM, 2009.

[4] J. Brown. Calculation of a constant q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[5] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[6] D. Ellis and G. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. of ICASSP*, pages 1429–1433, 2007.

[7] D. Ellis and A. Weller. The 2010 LABROSA chord recognition system. In *Proc. of ISMIR (MIREX submission)*, 2010.

[8] J.L. Elman. Learning and development in neural networks: the importance of starting small. *Cognition*, 48(1):71–99, 1993.

[9] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proc. ICMC, 1999*, pages 464–467, 1999.

[10] K.A. Krueger and P. Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009.

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

[12] K. Lee and M. Slaney. Automatic chord recognition from audio using an HMM with supervised learning. In *Proc. of ISMIR*, 2006.

[13] R. Macrae and S. Dixon. Guitar Tab Mining, Analysis and Ranking. In *ISMIR 2011, Miami, Florida*, 2011.

[14] M. Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London, 2010.

[15] M. McVicar, Yizhao. Ni, R. Santos-Rodriguez, and T. De Bie. Using online chord databases to enhance chord recognition. *JNMR, special issue on music and machine learning*, 2011.

[16] Y. Ni, M. Mcvicar, R. Santos-Rodriguez, and T. De Bie. An end-to-end machine learning system for harmonic analysis of music. In *http://arxiv.org/abs/1107.4969v1*, 2011.

[17] T. Rocher, M. Robine, P. Hanna, and L. Oudre. Concurrent Estimation of Chords and Keys from Audio. In *Proc. ISMIR*, 2010.

[18] T. Rocher, M. Robine, P. Hanna, L. Oudre, Y. Grenier, and C. Févotte. Concurrent estimation of chords and keys from audio. In *Proc. of ISMIR*, pages 141–146, 2010.