Automatic Chord Estimation from Audio: A Review of the State of the Art

Matt McVicar, Raúl Santos-Rodríguez, Yizhao Ni, and Tijl De Bie

***Abstract*—In this overview article, we review research on the task of Automatic Chord Estimation (ACE). The major contributions from the last 14 years of research are summarized, with detailed discussions of the following topics: feature extraction, modeling strategies, model training and datasets, and evaluation strategies. Results from the annual benchmarking evaluation Music Information Retrieval Evaluation eXchange (MIREX) are also discussed as well as developments in software implementations and the impact of ACE within MIR. We conclude with possible directions for future research.**

***Index Terms*— *[Author]*, please supply index terms/keywords for your paper. To download the IEEE Taxonomy go to http://www.ieee.org/documents/2009Taxonomy_v101.pdf. .**

I. Introduction

CHORDS are mid–level musical features which concisely describe the harmonic content of a piece. This is evidenced by chord sequences often being sufficient for musicians to play together in an unrehearsed situation [1]. In addition to their use by professional and amateur musicians as *lead sheets* (succinct written summaries typically containing chordal arrangement, melody, and lyrics [2]), chord sequences have been used by the research community in high–level tasks such as *cover song identification* (identifying different versions of the same song e.g.[3], [4]), *key detection* [5]–[8], *genre classification* (identifying style [9]), *lyric interpretation* [10] and *audio–to–lyrics alignment* [11], [12]. A typical chord annotation for a popular music track, as used in *Automatic Chord Estimation* (ACE) research, is shown in Fig. 1.

```
0.000000 2.612267 N
2.612267 11.459070 E
11.459070 12.921927 A
12.921927 17.443474 E
17.443474 20.410362 B
20.410362 21.908049 E
21.908049 23.370907 E:7/3
23.370907 24.856984 A
...
```

Fig. 1. Section of a typical chord annotation, showing onset time (first column), offset time (second column), and chord label (third column).

Unfortunately, annotating chord sequences manually is a time–consuming and expensive process: typically it requires two or more experts and an average annotation time of eight to 18 minutes per annotator per song [13] and can only be conducted by individuals with sufficient musical training and/or practice. Because of this, in recent years ACE has become a very active area of research, attracting a wide range of researchers from electrical engineering, computer science, signal processing and machine learning. ACE systems have been benchmarked in the annual MIREX (Music Information Retrieval Evaluation eXchange) ACE subtask, which has seen a slow but steady improvement in accuracy since its inception in 2008, with the submissions in 2012 surpassing 72% accuracy on unseen test data, measured in terms of percentage of correctly identified frames on a set of songs for which the ground truth is known.

In the current paper, we conduct a thorough review of the task of ACE, with emphasis on feature extraction, modelling techniques, datasets, evaluation strategies and available software packages, covering all the aspects of ACE research (diagrammed in Fig. 2). We begin by providing an account of the chromagram feature matrices used by most modern systems as audio representations in Section II. These features began as simple octave and pitch–summed spectrograms, but have steadily incorporated optimizations such as *tuning*, *background spectrum removal* and *beat–synchronization*.

In parallel to audio feature design, decoding chromagrams into an estimated chord sequence also begun with simple Viterbi decoding under a *Hidden Markov Model* (HMM) architecture, but has in recent years become more complex, making the prediction of chords such as *seventh chords* and inversions possible via the use of *factorial–HMMs* and *Dynamic Bayesian Networks* (DBNs). We will provide a detailed discussion on these models and their structures in Section III.

This will lead us into a discussion of data–driven versus expert knowledge systems and the amount of fully and partially–labelled data available to the community for model training and how this may be utilized. From early hand–crafted sets of 180 songs by The Beatles, gradually the number of fully–annotated datasets has been steadily increasing, with the recent announcement of the Billboard set of close to 1,000

Manuscript received April 03, 2013; revised July 20, 2013; accepted September 19, 2013. Date of publication nulldate; date of current version nulldate. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Woon-Seng Gan.

The authors are with the Intelligent Systems Lab, Department of Engineering Mathematics, University of Bristol, Bristol BS8 1UB, U.K. (e-mail: mattjamesmcvicar@gmail.com; yizhao.ni@gmail.com; rsantos.uc3m@gmail.com; tijl.debie@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2013.2294580

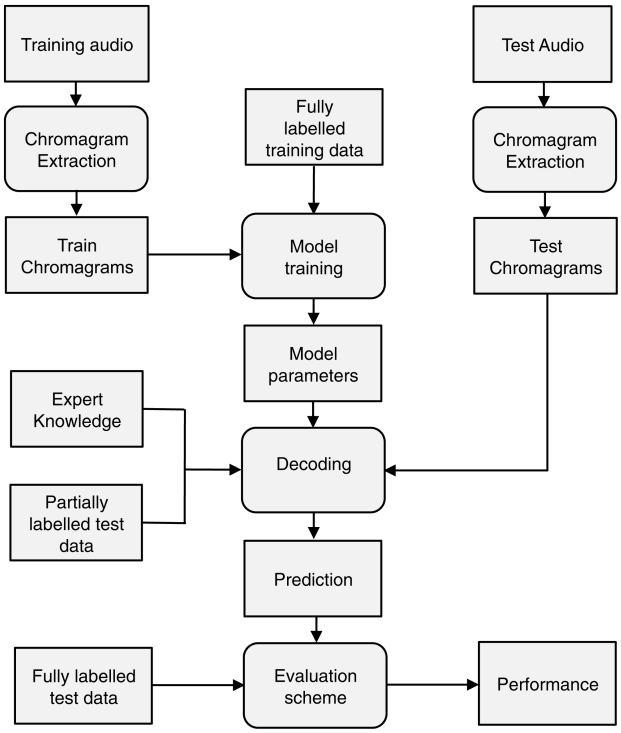

Fig. 2. Work flow normally associated with ACE research. Data are shown as rectangles, processes as rounded rectangles. First, chromagrams are extracted for the training data, which may be used to estimate model parameters. Either expert knowledge or these model parameters are then used to infer chord sequences from the test chromagrams, sometimes with partially–labelled test data. Predictions are then compared to hand-labelled examples to derive a performance measure.

chord and key annotations being the most significant in recent years [13]. Some authors have also been exploring the use of partially–labelled datasets as an additional source of information [14], [15]. An investigation of these data and their strengths and weaknesses will be presented in Section IV.

The wealth of information in chordal data available today (currently available data sources include 4 and 5–note chords beyond the octave including inversions) will prompt us to investigate evaluation strategies for ACE systems (Section V), including a discussion of the annual MIREX evaluations. Section VI and VII then deal with software implementations and the impact of ACE within the MIR domain. Finally, we conclude in Section VIII.

This review is structured logically rather than chronologically. However, for reference a chronological list of key developments in ACE research is provided as an Appendix.

A. Chords and their Musical Function

An in–depth discussion of chords and their function in music is beyond the scope of this paper (for detailed discussions, the reader is referred to the theses of Harte [16] or Mauch [17]). However, in this Subsection we provide a basic introduction to the definition and construction of chords used in popular music for those unfamiliar with music theory.

Broadly speaking, the tonal content of Western popular music can be seen to occupy two dimensions: *vertical* movement, which comprises relatively rapid changes in pitch known as melody, and *horizontal* movement, consisting of slower–changing sustained pitches played in unison, known as harmony, or chords.

Loosely then, a chord is simply two or more notes held together in unison. Using the familiar *pitch class* set of (C, C♯/D♭, D, D♯/E♭, E, F, F♯/G♭, G, G♯/A♭, A, A♯/B♭, B), chords comprise of a root (starting note chosen from the pitch class set) and chord quality. The most common chord types used in ACE research have quality *major* or *minor*, comprising of a perfect fifth (7 pitches above root) and a major third (4 pitches above root) or minor third (3 pitches above root) respectively. The set of intervals a chord contains is sometimes called a *degree list*, which can be useful for describing more arcane chords for which a concise quality name is not available.

In many musical styles, a subset of chords containing notes from an associated scale (a subset of the 12 possible pitches in Western music), are more prominent. This collection of chords defines a musical key, a global property characterizing the entire piece from which the chords derive their pitches. The methods by which the key is established are complex and have changed over music history. For the purposes of our discussion however it suffices to know that prior knowledge of musical key makes certain chords more likely than others and vice-versa. Key detection has also become a task unto itself in recent years [5]–[7], [18].

In notating chords, we adhere to the suggestion of Harte [19] and denote chords by their root note, degrees, (or shorthand) and optional inversion (order in which the notes appear). For example, a C major chord in first inversion will be written as either C:(1,3,5)/3 or C major/3. Note that more complex chords featuring four or more unique notes are also common, (see [16], Section 6.6, up to 20% frequency) some of which will be discussed in Sub. V-A.

II. Feature Extraction

The core representation of the audio used by most modern ACE systems is the *chromagram* [20]. Although many variants exist, they all describe how the pitch class saliences vary across the duration of the audio. Here, the meaning of 'salience' can be formalized in many different ways, as we will discuss below.

A chromagram can be represented by means of a real–valued matrix $\mathbf{X}$ containing a row for each pitch class considered, and column for each frame (i.e. discretized time point) considered. A vector containing the pitch class saliences at a specific time point, corresponding to a specific column $\mathbf{x}$ from $\mathbf{X}$, is known as a *chroma vector* or *chroma feature*.

To our knowledge, the first mention of the chromagram representation was by Shepard [21], where it was noticed that two dimensions, (*tone height* and *chroma*) were useful in explaining how the human auditory system functions. Here, the word *chroma* is used to describe pitch class, whereas *tone height* refers to the octave information. A typical chromagram feature matrix, with accompanying ground truth chord sequence, is shown in Fig. 3.

Early ACE methods were based on polyphonic note transcription [22]–[27], although it was Fujishima [28] who first considered ACE as a task unto itself. His chroma feature (which he called *Pitch Class Profile*, or PCP) involved taking

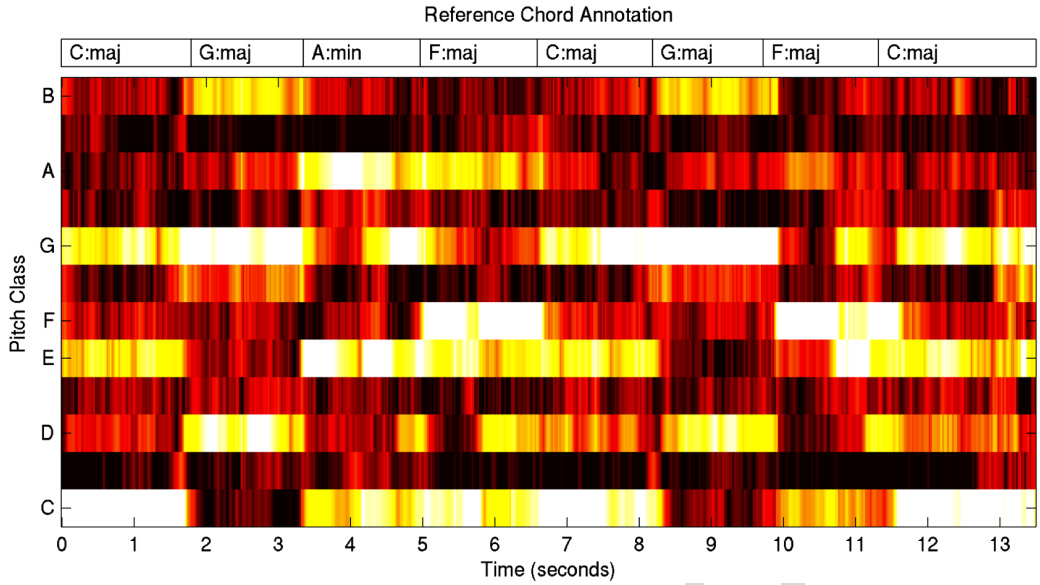

Fig. 3. A typical chromagram feature matrix, shown here for the opening to *let It Be* (Lennon/McCartney). Salience of pitch class p at time t is estimated by the intensity of $(p, t)^{th}$ entry of the chromagram. The reference (ground truth) chord annotation is also shown above for comparison, where we have reduced the chords to major and minor classes for simplicity.

a *Discrete Fourier Transform* of a segment of the input audio, and from this calculating the power evolution over a set of frequency bands. Frequencies which were close to each pitch class $(\mathrm{C}, \mathrm{C}\sharp, \ldots, \mathrm{B})$ were then collected and collapsed to form a 12–dimensional chroma vector for each time frame.

The main steps for the calculation of a chromagram are shown in Fig. 4. In the remainder of the current section we will discuss each of these steps in greater detail.

A. Transformation to Frequency Domain

Digital music is typically sampled at up to 44,100 samples per second (CD–quality), meaning that a typical 210 second pop song is represented by an extremely high–dimensional vector for each audio channel. In this raw form, it is also not directly informative of the harmonic content of the audio. There is evidence that the human auditory system performs a transform from the time to frequency domain and that we are more sensitive to frequency magnitude than phase information [29], endowing us with the ability to perceive melodic and harmonic information. Mimicking this, the first step in the chromagram computation is a transformation of the signal to a lower–dimensional representation that is more directly informative of the frequency content.

A simple Fourier transform magnitude of the waveform would lead to a *global* description of the frequencies present in our target audio, with loss of all timing information. Naturally, ACE researchers are interested in the *local* harmonic variations. Thus instead a *Short Time Fourier Transform* (STFT) of the audio is often used, which computes the frequency magnitudes in a sliding window across the signal. These magnitude spectra are then collected as columns of a matrix known as the spectrogram.

One of the limitations of the STFT is that it uses a fixed–length window. Setting this parameter involves trading off the frequency resolution with the time resolution [30]: with short windows, frequencies with long wavelengths cannot be distinguished, whilst with a long window, a poor time resolution is obtained. Since for ACE purposes frequencies that are half a semi–tone apart need to be distinguishable, this sets a lower–bound on the window–length and hence an inherent limit on the time resolution. This resolution will be particularly poor if one wishes to capture low frequencies with the required semi–tone frequency resolution, meaning that the choice of frequency range over which to take the transform is an important design choice (although systems which utilize A-weighting are less sensitive to this bias as frequencies outside the optimal human sensitivity range will be de-emphasized, see Sub. II-D).

An alternative to the STFT that partially resolves this problem by making use of a frequency–dependent window length is the *Constant–Q* spectrum—first used in a musical context by Brown [31]. In terms of ACE, it was used by Nawab *et al.* [32]. This frequency representation has become very popular in recent years [33]–[37]. For reasons of brevity, the readers are referred to the original work by Brown [31] for the details of the Constant-Q spectrum.

B. Preprocessing Techniques

When considering a polyphonic musical excerpt, it is clear that not all of the signal will be beneficial in the understanding of harmony. Some authors [38]–[40] have defined the unhelpful part of the spectrum as the *background spectrum*, and attempted to remove it in order to enhance the clarity of their features. Removing the background spectrum has the potential advantage of cleaning up the resulting chromagram, at the risk of removing information which is useful for ACE. One must be therefore ensure that the content removed is not relevant to the task at hand.

1) Background Spectrum: One example of removing a general background spectrum filtering is median filtering of the

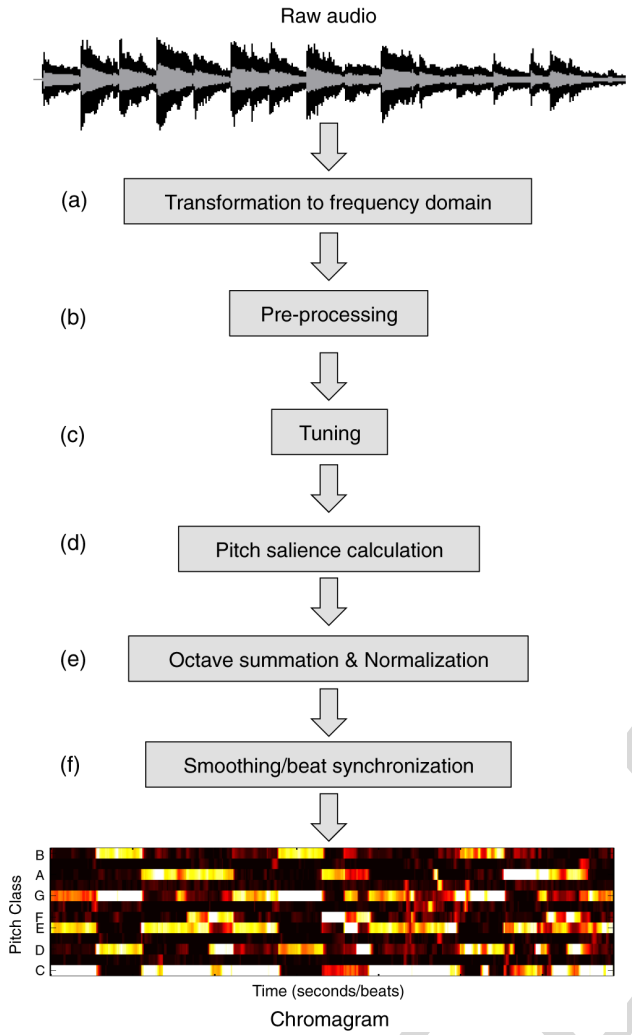

Fig. 4. Common steps to convert a digital audio file into its chromagram representation. The raw audio is converted from a time series to a frequency representation, pre–processed (e.g. removal of background spectrum/percussive elements/harmonics), tuned to standard pitch, and smoothed by mean/median filtering or beat synchronization before the pitch salience is calculated. Pitches belonging to the same pitch classes (C, C♯, . . . , B) are then summed and normalized to yield a chromagram feature matrix which captures the pitch evolution of the audio over time. Letters to the left of main processes refer to subsections, discussed in more detail in Section II.

spectrogram, as conducted by Mauch *et al.* [17]. A specific example of background noise when working in harmony–related tasks could be considered the percussive elements of the music. An attempt to remove the part of the spectrum due to percussive sounds was introduced in by Ono *et al.* [39] and used to increase ACE accuracy by Reed and collaborators [40]. It is assumed that the percussive elements of a spectrum (drums etc.) occupy a wide frequency range but are narrow in the time domain, and harmony (melody, chords, bassline) conversely. The spectrum is assumed to be a simple sum of percussive and harmonic material and can be diffused into two constituent spectra, from which the harmonic content can be used for chordal analysis. This process is known as Harmonic Percussive Source Separation (HPSS). It is shown by Reed and Ueda [40], [41] that HPSS improves ACE accuracy significantly, and is now employed in some modern feature extraction systems (see, for example, [36], [37]).

2) Harmonics: It is known that musical instruments emit not only a pure tone f_0, but a series of harmonics at higher frequencies, and subharmonics at lower frequencies. Such harmonics can easily confuse feature extraction techniques, and some authors have attempted to remove them in the feature extraction process [38], [42]–[44]. While we discuss it here, note that accounting for the presence of harmonics can be done before but also after tuning (see Section II-C).

A method of removing the background spectra and harmonics simultaneously was proposed by Varewyck *et al.*, based on multiple pitch tracking techniques [45]. They note that their new features matched chord profiles and perform better than unprocessed chromagrams, a technique which was also employed by Mauch [44].

A more recent method introduced in the same work is based on the assumption that each column in the spectrogram can be approximated well by a linear combination of note spectra (each of which includes harmonic frequencies above an f_0 frequency) [17]. Each weight in this linear combination corresponds to the *activation* of the corresponding note. The activation value of a note can then be ascribed to the pitch corresponding to its f_0 frequency. The activation vector can be estimated as the one minimizing the 2–norm distance between the linear combination of the note profiles and the actual spectrum observed. Considering that a note cannot be negatively activated, this amounts to solving a *Non–Negative Least Squares* (NNLS) problem [46].

Chromagrams computed in this way were shown to result in an improvement of six percentage points over the then state of the art system by the same authors [17] and are an interesting departure from energy-summed chromagrams.

C. Tuning

In 2003, Sheh and Ellis identified that some popular music tracks are not tuned to standard pitch $\mathrm{A}4 = 440$ Hz [47]. To compensate for this, they computed a spectrogram at twice the required frequency resolution (i.e. at half semi–tone resolution), allowing for some flexibility in the tuning of the piece. Harte introduced a tuning algorithm which computed the spectrogram over an even finer granularity of 3 frequency bands per semitone, and searched for the tuning maximizing the in–tune energy [48]. The actual saliences can then be inferred by interpolation. This method was also used by Bello and Pickens [34] and in Harte's own work [49] and is now a staple of most modern algorithms.

D. Capturing Pitch Class Salience

Although the pre–processed and tuned spectrogram of a signal is intuitively a good representation of the pitch evolution, some authors have been exploring ways of mapping this feature to something which more closely represents the human perception of pitch saliences.

Pauws made an early attempt to map the spectrum to the human auditory system by re–weighing the spectrum by an arc–tangent function in the context of audio key estimation [38].

A similar approach was taken by Ni *et al.*, where the loudness of the spectrum was calculated using A–weighting [50], resulting in loudness–based chromagrams, considerably improving ACE accuracy [36].

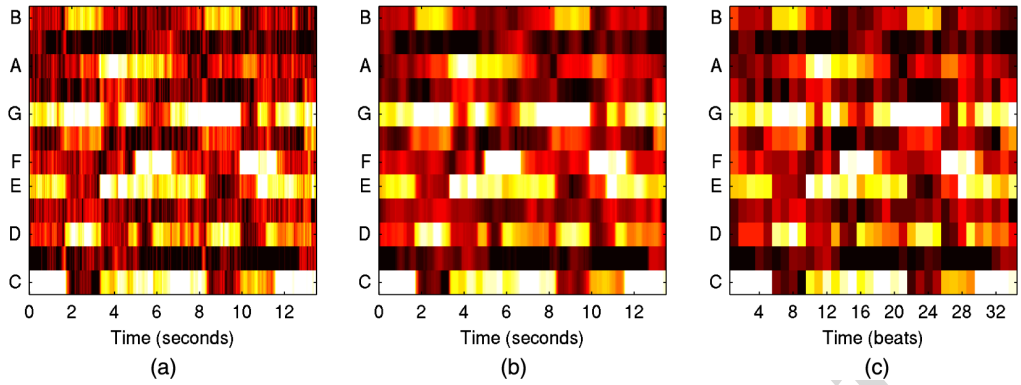

Fig. 5. Smoothing techniques for chromagram features. In 5a, we see a standard chromagram feature. Fig. 5b shows a median filter over 20 frames, 5c shows a beat–synchronized chromagram.

E. Octave Summation and Normalization

The final stage of chromagram calculation involves summing all pitch saliences belonging to the same pitch class, followed by a normalization. The first of these allows practitioners to work with a concise, 12–dimensional representation of the pitch evolution of the audio, disregarding the octave information which is often seen as irrelevant in ACE (although note that this implies that different positions of the same chord cannot be distinguished). A subsequent normalization per frame makes the result independent of (changes in) the volume of the track. Common normalization schemes include enforcing unit L^1, L^2, or L^∞ norm on each frame [51].

F. Smoothing/Beat Synchronization

It was noticed by Fujishima that using instantaneous chroma features led to chord predictions with frequent chord changes, owing to transients and noise [28]. As an initial solution, he introduced smoothing of the chroma vectors as a post–processing step. This heuristic was adopted by other authors using template–based ACE systems (see Section III).

In work by Bello, the fact that chords are usually stable between beats [52] was exploited to create *beat–synchronous* chromagrams, where the time resolution is reduced to that of the main pulse [34]. This method was shown to be superior in terms of accuracy, and had the additional advantage of reducing the computation cost, owing to the reduction in total number of frames.

Popular methods of smoothing chromagrams are to take the mean [34] or median [44] salience of each of the pitch classes between beats. In more recent work, Bello used recurrence plots within similar segments and showed it to be superior to beat synchronization or mean/median filtering [53]. Examples of smoothing techniques are shown in Fig. 5.

Papadopoulus and Peeters noted that a simultaneous estimate of beats led to an improvement in chords and vice–versa, supporting an argument that an integrated model of harmony and rhythm may offer improved performance in both tasks [54]. A comparative study of post–processing techniques was conducted by Cho *et al.*, who also compared different pre–filtering and modelling techniques [55].

G. Other Work on Features for ACE Research

Worth noting are two further techniques that do not naturally fit within the chromagram computation pipeline.

1) Tonal Centroid Vectors: An interesting departure from traditional chromagrams was presented by Harte *et al.*, notably a transform of the chromagram known as the *Tonal Centroid* feature [49]. This feature is based on the idea that close harmonic relationships such as perfect fifths and major/minor thirds have large Euclidean distance in a chromagram representation of pitch, and that a feature which places these pitches closer together may offer superior performance. To this end, the authors suggest mapping the 12 pitch classes onto a six–dimensional hypertorus which corresponds closely to Chew's spiral array model [56]. This feature vector has also been explored for key estimation [57], [58].

2) Integration of Bass Information: In some ACE systems two chromagrams are used: one for the treble range, and one for the bass range. The benefit of doing this was first recognized by Sumi *et al.* [59]. Within this work they estimate bass pitches from audio and add a bass probability into an existing hypothesis–search–based method and discovered an increase in accuracy of, on average, of 7.9 percentage points when including bass information [33]. Parallel treble and bass chroma examples are shown in Fig. 6.

Bass frequencies of 55–220 Hz were also considered in early work by Mauch, although this time by calculating a distinct *bass chromagram* over this frequency range [60]. Using a bass chromagram has the advantage of allowing one identify inversions of chords, which is used by the following two works: [36], [44].

III. Modelling Strategies

In this section, we review the next major step in ACE: assigning labels to chromagram (or related feature) frames. We begin with a discussion of simple pattern–matching techniques.

A. Template Matching

Template matching involves comparing feature vectors against the known distribution of notes in a chord, under the assumption that the chromagram feature matrix will closely resemble the underlying chords to the song. Typically, a 12–dimensional chroma vector is compared to a binary vector containing ones where a trial chord has notes present. For

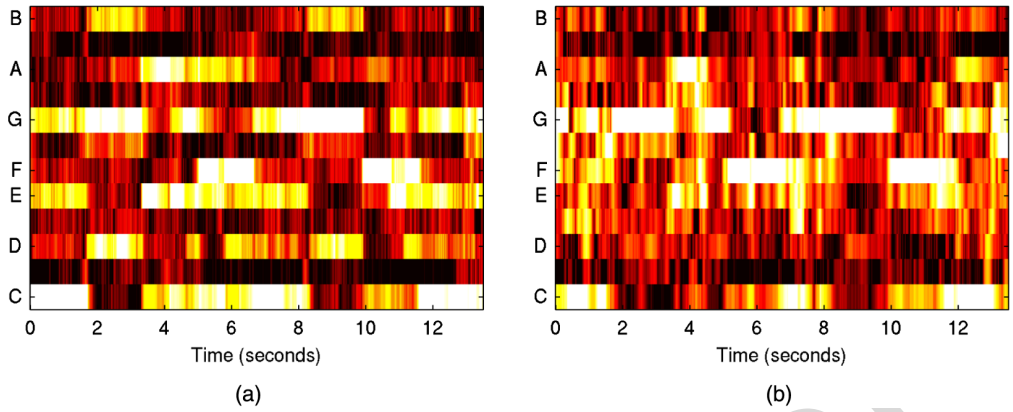

Fig. 6. Treble (6a) and Bass (6b) Chromagrams, with the bass feature taken over a frequency range of 55–207 Hz in an attempt to capture inversions.

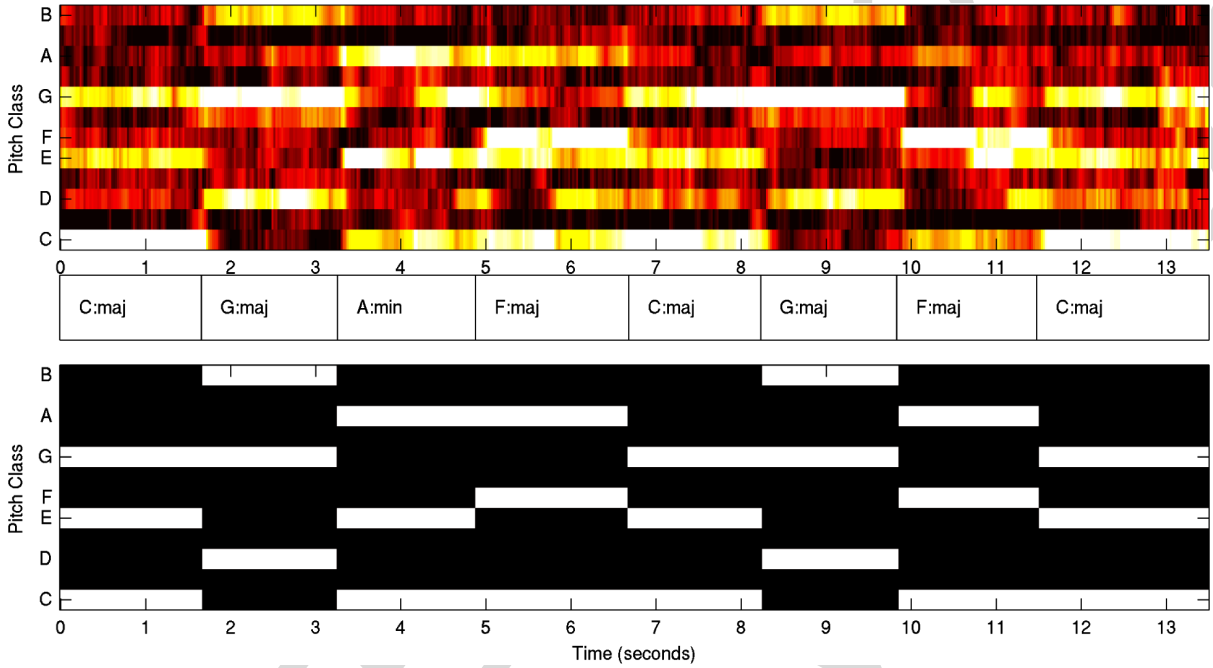

Fig. 7. Template–based approach to ACE, showing chromagram feature vectors, reference chord annotation and bit mask of optimal chord templates.

example, the template for a C Major chord would be [1 0 0 0 1 0 0 1 0 0 0 0]. Each frame of the chromagram is compared to a set of templates, and the template with maximal similarity to the chroma is output as the label for this frame (see Fig. 7). This technique was first proposed by Fujishima, where he used either the nearest neighbor template or a weighted sum of the PCP and chord template as a similarity measure between templates and chroma frames [28]. Similarly, this technique was used by Cabral and collaborators who compared it to the *Extractor Discovery System* (EDS) software to classify chords in Bossa Nova songs [61].

An alternative approach to template matching was proposed by Su and Jeng, who used a self–organizing map, trained using expert knowledge [62]. Although their system perfectly recognized the input signal's chord sequence, it is possible that the system is overfitted as it was measured on just one song instance. A more modern example of a template–based method is presented by Oudre and collaborators, who compared three distance measures and two post–processing smoothing types and found that Kullback–Leibler divergence [63] and median filtering offered an improvement over the then state of the art [64]. Further examples of template–based ACE systems can be found in later work by the same author and De Haas [65], [66].

B. Hidden Markov Models

Individual pattern matching techniques such as template matching fail to model the continuous nature of chord sequences. This can be combated either by using smoothing methods as seen in Section II or by including some notion of duration in the underlying model. One of the most common ways of incorporating smoothness in the model is to use a *Hidden Markov Model* (HMM). HMMs have become the most common method for assigning chord labels to frames in the ACE domain (see summary of MIREX submission in Section V-E).

An HMM is a probabilistic model for a sequence of observed variables, called the *observed variables*. The particular structure of the HMM model embodies certain assumptions on how these

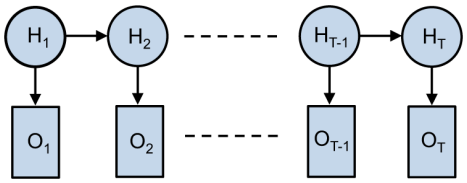

Fig. 8. Visualization of a first order Hidden Markov Model (HMM) of length T. Hidden states (chords) are shown as circular nodes, which emit observable states (e.g. rectangular nodes and chroma frames).

variables are probabilistically dependent on each other. In particular, it is assumed that there is a sequence of hidden variables, paired with the observed variables, and that each observed variable is independent of all others when conditioned on its corresponding hidden variable. Additionally, it is assumed that the hidden variables form a Markov chain of order 1.

Fig. 8 depicts a representation of the dependency structure of an HMM in the form of a *probabilistic graphical model*, applied to the ACE problem setting: the hidden variables are the chords in subsequent frames, and the observed variables are the chroma (or similar) features in the corresponding frame.

We briefly discuss the mathematical details of the HMM for ACE. For more details in HMMs in general, the reader is referred to the tutorial by Rabiner [67], whereas the HMM for ACE is covered in detail in e.g. [36].

Recall that we denote the chromagram of a particular song as $\mathbf{X}$ with 12 rows and as many columns as there are frames. Let us use the symbol $\mathbf{y}$ to denote a sequence of chord symbols (the *chord annotation*), with length equal to the number of frames. Each chord symbol comes from an agreed alphabet of chords considered (see Section V). HMMs can be used to formalize a probability distribution $P(\mathbf{y}, \mathbf{X}|\Theta)$ jointly for the chromagram $\mathbf{X}$ and the annotation $\mathbf{y}$ of a song, where Θ are the parameters of this distribution.

In this model, the chords are modelled as a first–order Markovian process, meaning that future chords are independent of the past given the present. Furthermore, given a chord, the 12–dimensional chromagram feature vectors in the corresponding time window is assumed to be independent of all other variables in the model. The chords are referred to as the *hidden variables* of the model and the chromagram frames as the *observed variables*.

Mathematically, the Markov and conditional independence assumptions allow the factorization of the joint probability of the feature vectors and chords $(\mathbf{X}, \mathbf{y})$ of a song into the following form:

$$P(\mathbf{X}, \mathbf{y}|\Theta) = P_{\text{ini}}(y_1) \cdot P_{\text{obs}}(\mathbf{x}_1|y_1) \cdot \prod_t P_{\text{tr}}(y_t|y_{t-1}) P_{\text{obs}}(\mathbf{x_t}|y_t) \quad (1)$$

Here, $P_{\text{ini}}(y_1)$ is the probability that the first chord is equal to y_1 (the *initial distribution* or *prior*), $P_{\text{tr}}(y_t|y_{t-1})$ is the probability that a chord y_{t-1} is followed by chord y_t in the subsequent frame (the *transition* probabilities, corresponding to the horizontal arrows in Fig. 8), and $P_{\text{obs}}(\mathbf{x}_t|y_t)$ is the probability density for chroma vector $\mathbf{x}_t$ given that the chord of the tth frame is y_t (the *emission probabilities*, indicated by the vertical arrows in Fig. 8).

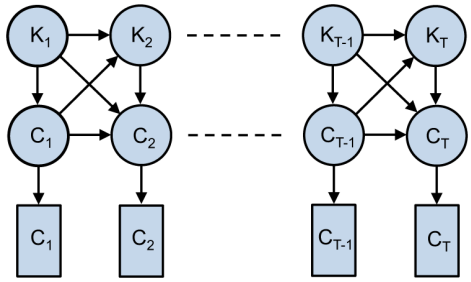

Fig. 9. Two–chain HMM, here representing hidden nodes for Keys and Chords, emitting Observed nodes. All possible hidden transitions are shown in this figure, although these are rarely considered by researchers.

It is common to assume that the HMM is stationary, which means that P_{tr} and P_{obs} are independent of t. Furthermore, it is common to model the emission probabilities as a 12–dimensional Gaussian distribution, meaning that the parameter set Θ of an HMM used for ACE are commonly given by

$$\Theta = \{\mathbf{P}_{\text{tr}}, \mathbf{P}_{\text{ini}}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}, \quad (2)$$

where it is convenient to gather the parameters into matrix form: $\mathbf{P}_{\text{tr}} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ are the transition probabilities, $\mathbf{P}_{\text{ini}} \in \mathbb{R}^{|\mathcal{A}|}$ is the initial distribution, and $\boldsymbol{\mu} \in \mathbb{R}^{12\times|\mathcal{A}|}$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{12\times12\times|\mathcal{A}|}$ are mean vectors and covariance matrices for a multivariate Gaussian distribution respectively.

Although HMMs are very common in the domain of speech estimation [67], we found the first example of an HMM in the domain of music transcription to be by Martin, where the task was to transcribe piano notation directly from audio [24]. In terms of ACE, the first example can be seen in the work by Sheh and Ellis, where HMMs and the Expectation–Maximization algorithm [68] are used to train a model for chord boundary prediction and labelling [47]. Although initial results were quite poor (maximum accuracy of 26.4%), this work inspired the subsequently dominant use of the HMM architecture in ACE.

A real–time adaptation of the HMM architecture was proposed by Cho and Bello, who found that with a relatively small lag of 20 frames (less than 1 second), performance is less than 1% worse than an HMM with access to the entire signal [69]. The idea of real–time analysis was also explored by Stark and collaborators, who employ a simpler, template–based approach [70].

C. Incorporating Key Information

Simultaneous estimation of chords and keys can be obtained by including an additional hidden chain into an HMM architecture. An example of this can be seen in Fig. 9. This *two–chain* HMM clearly has many more conditional probabilities than the simpler HMM, owing to the inclusion of a key chain, which may be used model, e.g. dominant chords preceding a change in key. This is an issue for both expert systems and data–driven systems, since there may be insufficient knowledge or training data to accurately estimate these distributions. As such, most authors disregard the diagonal dependencies in Fig. 9 [6], [36], [44].

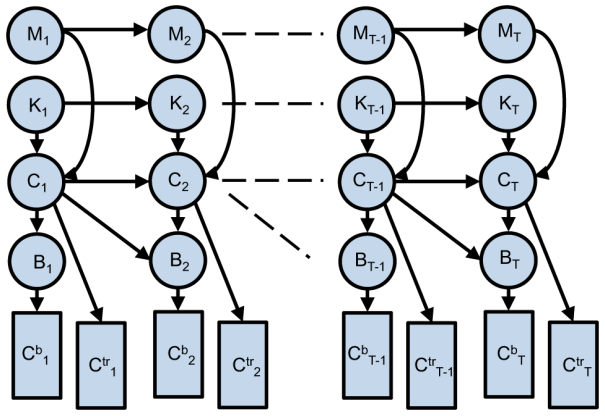

Fig. 10. Mauch's DBN, the Musical Probabilistic Model. Hidden nodes M_t, K_t, C_t, B_t represent metric position, key, chord and bass annotations, whilst observed nodes C_t^{tr} and C_t^b represent treble and bass chromagrams.

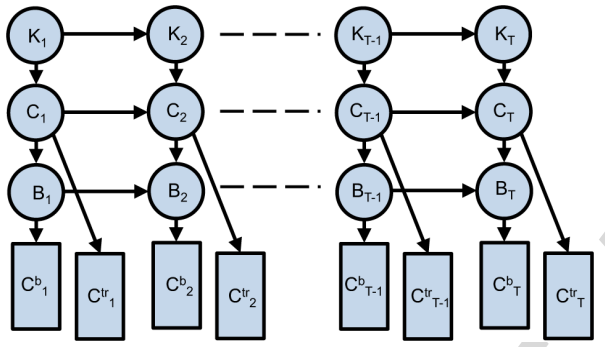

Fig. 11. Ni *et. al.*'s Harmony Progression Analyzer. Hidden nodes K_t, C_t, B_t represent key, chord and bass annotations, whilst observed nodes C_t^{tr} and C_t^b represent treble and bass chromagrams.

D. Dynamic Bayesian Networks

A significant advance in modelling strategies came in 2010 with the introduction of Mauch's *Dynamic Bayesian Network* model [17], [44], shown in Fig. 10. This sophisticated model has hidden nodes representing metric position, musical key, chord, and bass note, as well as observed treble and bass chromagrams. Dependencies between chords and treble chromagrams are as in a standard HMM, but with additional emissions from bass nodes to lower frequency–range chroma features, and interplay between metric position, keys and chords. This model was shown to be extremely effective in the ACE task in the MIREX evaluation in 2010, attaining performance of 80.22% chord overlap ratio on the MIREX dataset (see MIREX evaluations III).

In 2011, Ni *et al.* designed a DBN–based ACE system named the Harmony Progression Analyzer (HPA) [36]. The model architecture has hidden nodes for chord, inversion and musical key and emits a bass and treble chromagram at each frame (see Fig. 11). This model was top-performing in the most recent MIREX evaluation of 2012 (see Section V).

E. High-order HMMs

A high-order model for ACE was proposed by Scholz and collaborators [71], based on earlier work [72], [73]. In particular, they suggest that the typical first–order Markov assumption is insufficient to model the complexity of music, and instead suggest using higher–order statistics such as *second-order HMMs*. They found that high-order models offer lower perplexities[1] than first-order HMMs (suggesting superior generalization), but that results were sensitive to the type of smoothing used, and that high memory complexity was also an issue.

This idea was further expanded by Khadkevich and Omologo, where an improvement of around 2% absolute was seen by using a n-order ($n = 2, 3$) model [74], and further in [75] where chord idioms similar to Mauch's findings [73] are discovered, although within this work they use an infinity–order model where a specification of n is not required.

F. Discriminative Models

In 2007, Burgoyne *et al.* suggested that generative HMMs are suboptimal for use in ACE, preferring instead the use of discriminative *Conditional Random Fields* (CRF) [76].

During decoding, an HMM seeks to maximize the overall joint distribution over the chords and feature vectors $P(\mathbf{X}, \mathbf{y})$. However, for a given song example the observation is always fixed, so it may be more sensible to model the conditional $P(\mathbf{y}|\mathbf{X})$, relaxing the necessity for the components of the observations to be conditionally independent. In this way, discriminative models attempt to achieve accurate input (chromagram) to output (chord sequence) mappings.

An additional potential benefit to this modelling strategy is that one may address the balance between, for example, the hidden and observation probabilities, or take into account more than one frame (or indeed an entire chromagram) in labelling a particular frame. This last approach was explored by Weller *et al.* [77], where the recently developed *SVM struct* algorithm was used as opposed to CRF, in addition to incorporating information about future chroma vectors to show an improvement over a standard HMM.

G. Genre–Specific Models

Lee [7] has suggested that training a single model on a wide range of genres may lead to poor generalization, an idea which was expanded on in later work [58], wherein it was found that if genre information was given (for a range of six genres), performance increased almost ten percentage points. Also, they note that their method can be used to identify genre in a probabilistic way, by simply testing all genre–specific models and choosing the model with largest likelihood.

H. Emission Probabilities

When considering the probability of a chord emitting a feature vector in graphical models, as is commonly required [47], [60], [78] one must specify a probability distribution for a chromagram frame, given a list of candidate chords. A common method for doing this is to use a 12–dimensional Gaussian distribution, i.e. the probability of a chord c emitting a chromagram frame $\mathbf{x}$ is set as $P(\mathbf{x}|c) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, with $\boldsymbol{\mu}$ a 12–dimensional mean vector for each chord and $\boldsymbol{\Sigma}$ a collection of covariance matrices for each chord. One may then estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from data or expert knowledge and infer the emission probability for a (chord, chroma) pair.

[1]the perplexity of a probability distribution p with entropy $H(p)$ is defined as $2^{H(p)}$

TABLE I
GROUND TRUTH DATASETS AVAILABLE FOR RESEARCHERS IN ACE, INCLUDING NUMBER OF UNIQUE TRACKS AND UNIQUE ARTISTS

Dataset name	Authors	# tracks	# artists	Source
MIREX	[19]	217	3	http://www.isophonics.net/content/reference-annotations
USpop	[37]	195	115	https://github.com/tmc323/Chord-Annotations
McGill	[13]	649	348	http://ddmal.music.mcgill.ca/billboard

This technique has been very widely used in the literature (see, for example [34], [47], [74], [79]). A slightly more sophisticated emission model is to consider a mixture of Gaussians, instead of one per chord. This has been explored in, for example, the work by Sumi, Bello and Reed [40], [53], [59].

A different emission model was proposed in early work by Burgoyne [80], that of a Dirichlet model. Given a chromagram with pitch classes $p = \{c_1, \ldots, c_{12}\}$, each with probability $\{p_1, \ldots, p_{12}\}$ and $\sum_{i=1}^{12} p_i = 1, p_i > 0 \quad \forall i$, a *Dirichlet distribution* with parameters $u = \{u_1, \ldots, u_{12}\}$ is defined as

$$P(\mathbf{x}|c) = \frac{1}{N_u} \prod_{i=1}^{12} {p_i}^{u_i - 1} \tag{3}$$

where N_u is a normalization term. Thus, a Dirichlet distribution is a distribution over numbers which sum to one, and a good candidate for a chromagram feature vector. This emission model was implemented for ACE by Burgoyne *et al.*, with encouraging results [76]. One final development in emission modelling came when Ni and collaborators trained emission probabilities over a range of genres, allowing for parameter sharing between genres which fell under the same 'hyper genre' [81].

IV. MODEL TRAINING AND DATASETS

Ground truth chord data in the style of Fig. 1 are essential for testing the accuracy of an ACE system; for data–driven systems, they also serve as a training source. In this section, we review the data available to ACE researchers, how the data can be used for training, and discuss the benefits and drawbacks of systems based on expert knowledge versus data–driven systems.

A. Available Datasets

The first dataset made available to researchers was released by Harte and collaborators in 2005, which consisted of 180 annotations to songs by the pop group The Beatles, later expanded to include works by Queen and Zweieck [19]. In this work they also introduced a syntax for annotating chords in flat text, which has since become standard practice.

This dataset was used extensively within the community [36], [37], [82] but one concern was that the variation in chord labels and instrumentation/style was limited. Perhaps because of this, other researchers began working on datasets covering a wider range of artists, although mostly within the pop/rock genre. A 195 song subset of the 'USpop' dataset ([83], 8,752 songs total) were hand–annotated by Cho [37] and released to the public. Around the same time, research from McGill university [13], [84] yielded a set of 649 available titles, with at least a further 197 kept unreleased for MIREX evaluations (see Section V-E). A summary of the three main datasets available to researchers is shown in Table I.

B. Training Using Expert Knowledge

In early ACE research, when training data was very scarce, an HMM was used by Bello and Pickens [34], where model parameters such as the transition probabilities, mean and covariance matrices were set initially by hand, and then enhanced using the Expectation–Maximization algorithm [67].

A large amount of knowledge was injected into Shenoy and Wang's key/chord/rhythm extraction algorithm [6]. For example, they set high weights to common chords in each key (see Sub. I-A), additionally specifying that if the first three measures of a bar are a single chord, the last measure must also be this chord, and that chords non–diatonic to the current key are not permissible. They noticed that by making a rough estimate of the chord sequence, they were able to extract the global key of a piece (assuming no modulations) with high accuracy (28/30 song examples). Using this key, ACE accuracy increased by an absolute 15.07%.

Expert tuning of key–chord dependencies was also explored by Catteau and collaborators [5], following the theory set out in Lerdahl [85]. A study of expert knowledge versus training was conducted by Papadopoulus and Peeters, who compared expert setting of Gaussian emissions and transition probabilities, and found that expert tuning with representation of harmonics performed the best [43]. However, they only used 110 songs in the evaluation, and it is possible that with the additional data now available, a data–driven approach may be superior.

Mauch and Dixon also opted for an expert–based approach to ACE parameter setting, defining chord emission and emission models according to musical theory or heuristics (such as setting the tonal key to have self–transition probability equal to 0.98 [44]). More recently, De Haas and collaborators employed a template–based approach to chose likely chord candidates and broke close ties with musical theory [66].

C. Training Using Fully Labelled Datasets

Recall that the parameters for an HMM are referred to as Θ. We now turn attention to learning Θ. To infer a suitable value for Θ using a set of fully labelled training examples $\{\mathcal{X}, \mathcal{Y}\}$, Maximum Likelihood Estimation can be used [67]. In order to make the most of the available training data, some authors exploit symmetry in musical harmony by transposing all chord types to the same tonic before training [86], [87]. This means that one may learn a generic 'major chord' (for example) model, rather than individual C major, C♯ major, ... models, effectively increasing the amount of training data for each chord type by a

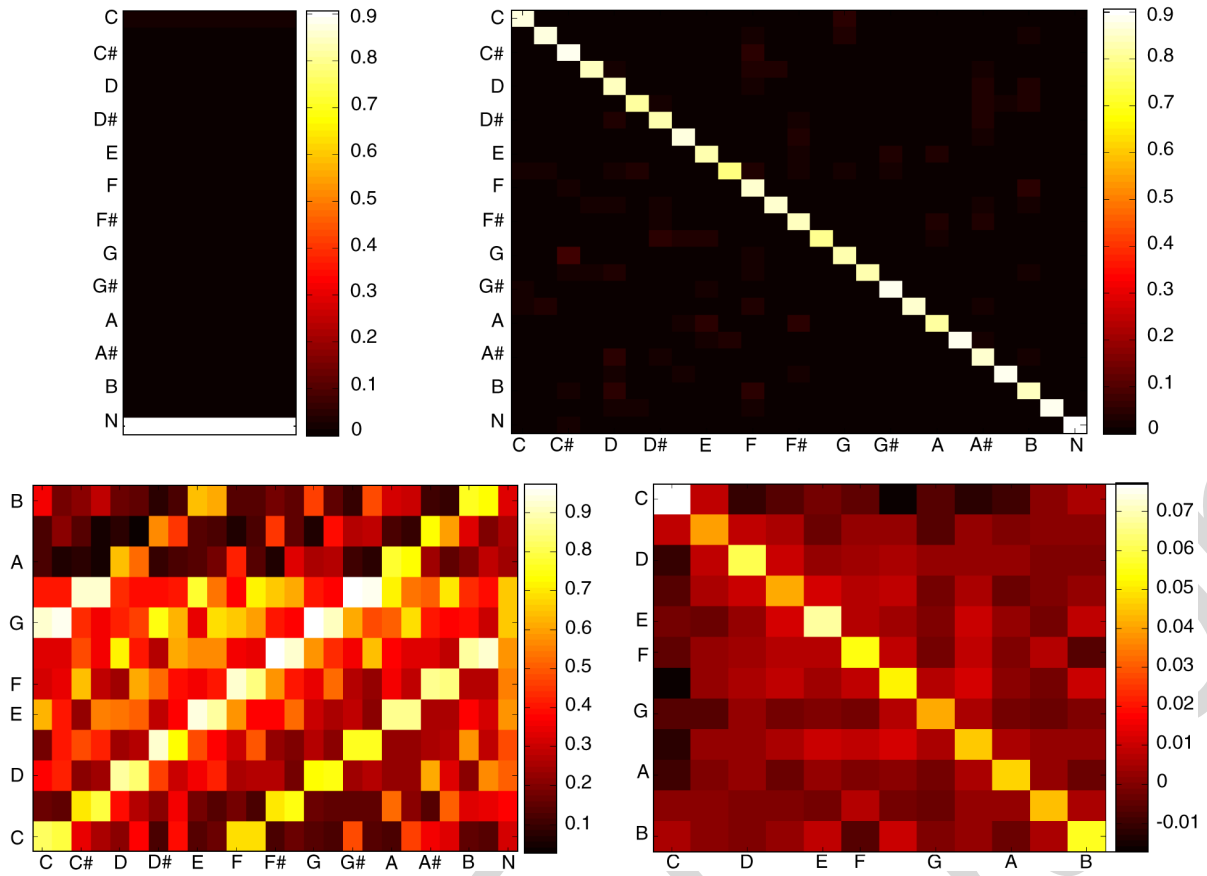

Fig. 12. HMM parameters, trained using Maximum likelihood on the MIREX dataset. Above, left: initial distribution $\mathbf{p}^*_{ini}$. Above, right: transition probabilities $\mathbf{P}^*_{tr}$. Below, left: mean vectors for each chord $\boldsymbol{\mu}^*$. Below, right: covariance matrix $\boldsymbol{\Sigma}^*$ for a C major chord. In all cases to preserve clarity, parallel minors for each chord and accidentals follow to the right and below.

factor of 12. These parameters may then be transposed 12 times to yield a model for each pitch class.

We show example parameters (trained on the ground truths from the 2011 MIREX dataset, without transposition) in Fig. 12. Inspection of these features reveals that musically meaningful parameters can be learned from the data, without the need of expert knowledge. Notice, for example, how the initial distribution is strongly peaked to starting on *no chord*, as expected (most songs begin with silence). Furthermore, we see strong self–transitions in line with our expectation that chords are constant over several beats. The mean vectors bear close resemblance to the pitches present within each chord and the covariance matrix is almost diagonal, meaning there is little covariance between notes in chords.

D. Learning From Partially–labelled Datasets

Some authors have been exploring the use of readily–available chord transcriptions from guitar tab websites to aid in testing, training, ranking, musical education, and score following of chords [78], [88], [89].

Such annotations are of course noisy and, lacking any chord timing information other than their ordering, they are harder to exploit for training ACE systems. Even so, in work by McVicar it is shown that they represent a valuable resource for ACE, owing to the volume of such data available [90]. A further help in using them is the fact that a large number of examples of each song are available on such sites. For example, Macrae and Dixon found 24,746 versions for songs by The Beatles, or an average of 137.5 tabs per song [15].

E. Discussion of Expert vs Data–driven systems

With the two classes of ACE systems now clear (expert and data–driven), we discuss the strengths and weaknesses of each in the current subsection. The first thing to note is that both systems employ some musical and psychoacoustic knowledge in their implementation. For example, all modern systems are based on modifying the spectrogram to match the equal–tempered scale, and most search for deviations from the standard $\mathrm{A4} = 440$ Hz. Further to this, summing pitches which belong to the same pitch class to form a chromagram is now standard practice, derived from the human perception of sound. Musical theory is also injected into choice of hidden nodes in HMMs or DBNs.

However, the inference of model parameters is where the two systems begin to differ. The performance attained using either system variant is upper–bounded by the quality and quantity of the knowledge contained within, and the choice over which paradigm should be used depends on the availability and trustworthiness of the sources. Considering an extreme example, if there is no training data available, an expert system is the only choice. As the number and variation of training examples increases, more can be learned from ground truth annotations. At

the other extreme, in the case of an infinitely large corpus of annotations, the parameters estimated will converge to the 'true' values and be more refined than a subjective notion of musical theory.

It is possible for both types of training to overfit data and attain poor generalization. The case for data–driven systems is clearest to see, since the maximum likelihood solution to the model training problem assumes that the test distribution is identical to that of the training. However, the same can be said for expert systems. There is no universally agreed–upon musical theory of chord transitions, or how chords interact with beat position, keys or basslines. As such, designers of expert systems may pick and choose particular musical facets which produce favorable results on their test set. Publication bias towards positive research results will have the same hard–to–quantify effect.

Since the training and test data are typically known to researchers in order to evaluate their systems, it is particularly difficult to estimate to what extent researchers are overfitting their data. This in theory can be solved by having a held–out test set which is used solely for evaluation and not used for training in any way. However, this is difficult to do in practice since reviewing which mistakes are being made on the test set can often yield improvements, and it is impossible to tell to what extent this has happened in a particular research paper. The same can be said for most iterations of the MIREX ACE task, since all candidates have had access to the test data, except in the most recent incarnation in 2012 (see Section V-E).

V. Evaluation Strategies

Given the output of an ACE system and a known and trusted ground truth, methods of performance evaluation are required to compare algorithms and define the state of the art. We discuss strategies for this in the current section, focusing on frame-based analysis (an overview of alternative evaluation strategies can be found in the work by Pauwels [91] or Konz [92]). We will begin by reviewing the different chord alphabets used by researchers in the domain of ACE. We then discuss how one might compare a single predicted and trusted ground truth and chord alphabet, before moving on to a single song instance and finally a corpus of songs. In the current Section we will assume that there exists a predicted $\mathbf{Y}$ and ground truth $\mathbf{G}$ chord corpus sampled at the same resolution for $k = 1, \ldots, K$ songs given by:

$$\mathbf{Y} = \{\mathbf{y}^k | \mathbf{y}^k = [y_1^k, \ldots, y_t^k, \ldots, y_{T^k}^k]\}, \quad k = 1, \ldots, K$$
$$\mathbf{G} = \{\mathbf{g}^k | \mathbf{g}^k = [g_1^k, \ldots, g_t^k, \ldots, g_{T^k}^k]\}, \quad k = 1, \ldots, K$$

where T^k indicates the number of samples in the k'th song. Each predicted chord symbol y_t^k, g_t^k comes from a chord alphabet $\mathcal{A}$.

A. Chord Detail

Considering chords within a single octave, there are 12 pitch classes which may or may not be present, leaving us with 2^{12} possible chords. Such a chord alphabet is clearly prohibitive for modelling (owing to the computational complexity) and also poses issues in terms of evaluation. For these reasons, researchers in the field have reduced their reference chord annotations to a subset of workable alphabet.

In early work, Fujishima considered 27 chord types, including advanced examples such as A:(1, 3, ♯5, 7)/G [28]. A step forward to a more workable alphabet came in 2003, where Sheh and Ellis [47] considered seven chord types (maj, min, maj7, min7, dom7, aug, dim), although other authors have explored using just the four main triads: maj, min, aug and dim [33], [76]. Suspended chords were identified by Sumi and Mauch [59], [60], the latter study additionally containing a 'no chord' symbol for silence, speaking or other times when no chord can be assigned.

A large chord alphabet of ten chord types including inversions were recognized by Mauch [44]. However, by far the most common chord alphabet is the set of major and minor chords in addition to a 'no chord' symbol, which we collectively denote as *minmaj* [42], [43]. Note that as the sophistication of ACE systems improve, it is important to realize that retaining the simplistic minmaj alphabet will result in overfitting and a plateau in performance and so the publication of results on more complex chord types in future articles and MIREX evaluations should be encouraged.

B. Evaluating a Single Chord Label

Given a predicted and ground truth chord label pair (y, g) we must decide how to evaluate the similarity between them. The most natural choice is to have a binary correct/incorrect score indicating if the chord symbols are identical. This might seem appropriate for simple chord sets such as the collection of major and minor chords where there is little ambiguity, although for more complex chord alphabets this assumption is less clear.

Consider for example a chord alphabet which consists of major and minor chords with inversions. What should the evaluation of $y = \mathrm{C}$ major against $g = \mathrm{C}$ major/3 (i.e. a C major chord in first inversion) be? The pitch classes in both cases are identical (C,E,G) but their order differs. To combat this, Ni *et al.* have defined *note precision* to score equality between two chord labels if they share the same pitch classes, and *chord precision* to score equality only if the chord labels are identical [36].

A further complication occurs when dealing with chords with different pitch classes. Take $y = \mathrm{C}$ major and $g = \mathrm{C}$ major7 as representative examples. Clearly this prediction is more accurate than a prediction of, say, Bb major. However, this subtlety is not currently captured in any of the prevailing evaluation strategies. We are however aware of two other methods of evaluation, both of which have featured in the MIREX evaluations. The first method considers a predicted chord label to be correct if it shares the tonic and third with the true label. In this evaluation, which we refer to as the MIREX evaluation, labelling a C7 (C dominant 7th) frame as C major is considered correct. Finally, in the early years of ACE, a generous evaluation which only matched the tonic of the predicted chord was employed (see Sub. V-E)

C. Evaluating on a Song Instance

Fujishima first introduced the concept of the 'Relative Correct Overlap' measure for evaluating ACE accuracy on a song level, defined as the mean number of correctly identified frames [28]. Letting $E(y, g)$ be an evaluation strategy for single chord labels such as those mentioned in Sub. V-B, we may define the

TABLE II
MIREX SYSTEMS FROM 2008–2009, SORTED IN EACH YEAR BY TOTAL RELATIVE CORRECT OVERLAP IN THE MERGED EVALUATION. THE BEST–PERFORMING PRETRAINED/EXPERT SYSTEMS ARE Underlined, BEST TRAIN/TEST SYSTEMS ARE IN **BOLDFACE**. SYSTEMS WHERE NO DATA IS AVAILABLE ARE SHOWN BY A DASH (-)

					Performance	
Year	**Category**	**Sub.**	**Author(s)**	**Approach**	Unmerged	Merged
2008	Train/Test	UMS	Y. Uchiyama et al.	Chroma, HMM	**0.72**	**0.77**
		DE	D. Ellis	Chroma, HMM	0.66	0.70
		WD2	J. Weil	Tonal Centroid, HMM	0.66	0.70
	Pretrained	BP	J. P. Bello, J. Pickens	Chroma, HMM	0.66	0.69
		MM	M. Mehnert	Circular Pitch Space, HMM	0.65	0.68
		RK	M. Ryynnen, A. Klapuri	Bass/Treble Chroma, HMM	0.64	0.69
		PP	H.Papadopoulos, G. Peeters	Chroma, HMM	0.63	0.66
	Train/test	KO	M. Khadkevich, M. Omologo	Chroma, HMM	0.62	0.65
		WD1	J. Weil	Tonal Centroid, HMM	0.60	0.66
	Pretrained	KL2	K.Lee	-	0.59	0.65
	Train/test	KL	K. Lee	-	0.58	0.65
	Pretrained	KL1	K.Lee	-	0.56	0.60
	Train/test	ZL	X. Jhang, C. Lash	Chroma, HMM	0.36	0.46
2009	Train/Test	WEJ4	A. Weller et al.	Chroma, SVMstruct+	**0.742**	**0.777**
		WEJ2	A. Weller et al.	Chroma, SVMstruct	0.723	0.762
		WEJ3	A. Weller et al.	Chroma, Max–γ	0.723	0.760
	Expert	MD	M. Mauch et al.	Bass/Treble Chroma, DBN	0.712	0.748
	Pretrained	OGF2	L. Oudre et al.	Chroma, Template	0.711	0.777
		KO2	M. Khadkevich & M. Omologo	Chroma, HMM	0.708	0.741
		OGF1	L. Oudre et al.	Chroma, Template	0.706	0.770
	Train/Test	WEJ1	A. Weller et al.	Chroma, HMM	0.704	0.743
		RUSUSL	J.T.Reed et al.	Chroma, HMM	0.701	0.760
	Pretrained	KO1	M. Khadkevich & M. Omologo	Chroma, HMM	0.697	0.734
		DE	D. Ellis	Chroma, HMM	0.697	0.731
		PVM1	J. Pauwels et al.	Chroma, Key-HMM	0.682	0.710
		PVM2	J. Pauwels et al.	Chroma, Template	0.654	0.698
		CH	C. Harte	Chroma + Centroid, Template	0.654	0.698

Relative Correct Overlap (RCO) for the k'th song in terms of the notation in Eqn. (4) as:

$$\mathrm{RCO}^k = \frac{1}{T^k} \sum_{t=1}^{T^k} E(y_t^k, g_t^k) \tag{4}$$

D. Evaluating on a Song Corpus

When dealing with a collection of more than one song, one may either average the performances over each song, or concatenate all frames together and measure performance on this collection. The former treats each song equally independent of song length, whilst the latter gives more weight to longer songs.

We define these global and local averages as the Total Relative Correct Overlap and Average Relative Correct Overlap respectively. Letting $T = \sum_{k=1}^{K} T^k$ be the total number of frames in the corpus,

$$\mathrm{TRCO} = \frac{1}{T} \sum_{k=1}^{K} T^k \cdot \mathrm{RCO}^k \tag{5}$$

is the Total Relative Correct Overlap, and

$$\mathrm{ARCO} = \frac{1}{K} \sum_{k=1}^{K} \mathrm{RCO}^k \tag{6}$$

is the Average Relative Correct Overlap.

Finally, worth mentioning is that human experts do not always agree on the correct chord labels for a given song, as investigated experimentally by Ni [93].

TABLE III
MIREX SYSTEMS FROM 2010–2011, SORTED IN EACH YEAR BY TOTAL RELATIVE CORRECT OVERLAP. THE BEST–PERFORMING PRETRAINED/EXPERT SYSTEMS ARE Underlined, BEST TRAIN/TEST SYSTEMS ARE IN **BOLDFACE**. FOR 2011, SYSTEMS WHICH OBTAINED LESS THAN 0.35 TRCO ARE OMITTED

					Performance	
Year	**Category**	**Sub.**	**Author(s)**	**Approach**	TRCO	ARCO
2010	Expert	MD1	M. Mauch and S. Dixon	Bass/Treble Chroma, DBN	0.8022	0.7945
		MM1	M. Mauch	Bass/Treble Chroma, HMM	0.7963	0.7855
	Train/Test	CWB1	T. Cho et al.	-	**0.7937**	**0.7843**
		KO1	M. Khadkevich, M. Omologo	Bass/Treble Chroma, Language Model	0.7887	0.7761
	Pretrained	EW4	D. Ellis and A. Weller	Chroma, SVMstruct	0.7802	0.7691
		EW3	D. Ellis and A. Weller	Chroma, SVMstruct	0.7718	0.7587
	-	UUOS1	Y. Ueda et al.	Chroma, Key-HMM	0.7688	0.7567
	Hybrid	OFG1	L. Oudre et al.	Chroma, Template	0.7551	0.7404
	Train/Test	MK1	M. Khadkevich, M. Omologo	Chroma, HMM	0.7511	0.7363
		EW1	D. Ellis and A. Weller	Chroma, SVMstruct	0.7476	0.7337
	-	PVM1	J. Pauwels et al.	-	0.7366	0.7270
	Train/Test	EW2	D. Ellis and A. Weller	Chroma, SVMstruct	0.7296	0.7158
	Expert	PP1	H. Papadopoulos, G. Peeters	Chroma, Joint downbeat/chord estimate	0.5863	0.5729
2011	Pretrained	NMSD2	Y. Ni et al.	Memorization of Ground Truth	0.9760	0.9736
	Pretrained	KO1	M. Khadkevich, M. Omologo	Chroma, HMM	0.8285	0.8163
		NMSD3	Y. Ni et al.	Bass/Treble Chroma, DBN	0.8277	0.8197
		NM1	Y. Ni et al.	Bass/Treble Chroma, DBN	0.8199	0.8114
		CB2	T. Cho, J. P. Bello	Chroma, HMM	0.8137	0.8000
	Train/Test	CB3	T. Cho, J. P. Bello	Chroma, HMM	**0.8091**	**0.7957**
		KO2	M. Khadkevich, M. Omologo	Chroma, HMM	0.7977	0.7822
	Expert	CB1	T. Cho, J. P. Bello	Chroma, HMM	0.7955	0.7786
	Train/Test	NMSD1	Y. Ni et al.	Bass/Treble Chroma, DBN	0.7938	0.7829
		UUOS1	Y. Ueda et al.	Chroma, Language Model	0.7689	0.7564
	-	PVM1	J. Pauwels et al.	-	0.7396	0.7296
	Expert	RHRC1	T. Rocher et al.	Chroma, Key-HMM + Templates	0.7289	0.7151

E. The Music Information Retrieval Evaluation eXchange (MIREX)

Since 2008, ACE systems have been compared in an annual evaluation held in conjunction with the International Society for Music Information Retrieval.[2]Authors submit algorithms which are tested on a dataset of audio and ground truth. For ACE systems that require training, the dataset is split into a training set for training and a test set for evaluating the performance. We present a summary of the algorithms submitted in Tables II-III.

1) MIREX 2008: Ground truth data for the first MIREX evaluation was provided by Harte and consisted of 176 songs from The Beatles' back catalogue [19]. Approximately 2/3 of each of the 12 studio albums in the dataset was used for training and the remaining 1/3 for testing. Carrying out the split in this way avoided particularly easy/hard albums to end up primarily in either the training or test set, ensuring that the training and test sets are maximally can be regarded as independently sampled from identical distributions. Chord detail considered was either the set of major and minor chords, or a 'merged' set, where parallel major/minor chords in the predictions and ground truth were considered equal (i.e. classifying a C major chord as C minor was not considered an error).

Bello and Pickens achieved 0.69 overlap and 0.69 merged scores using a simple chroma and HMM approach, with Ryynnen and Klapuri achieving a similar merged performance using a combination of bass and treble chromagrams. Interestingly, Uchiyama *et al.* obtained higher scores under the train/test scenario (0.72/0.77 for overlap/merged). Given that the training and test data were known in this evaluation, the fact that the train/test scores are higher suggests that the pretrained systems did not make sufficient use of the available data in calibrating their models.

2) MIREX 2009: In 2009, the same evaluations were used, although the dataset increased to include 37 songs by Queen and Zweieck. Unfortunately, 7 songs whose average performance across all algorithms was less than 0.25 were removed, leaving a total of 210 song instances. Train/test scenarios were also evaluated, under the same major/minor or merged chord details.

This year, the top performing algorithm in terms of both evaluations was Weller *et al.*'s system, where chroma fea-

[2]http://www.music-ir.org/mirex/wiki/MIREX_HOME

TABLE IV
:MIREX SYSTEMS FROM 2012, SORTED IN EACH YEAR BY TOTAL RELATIVE CORRECT OVERLAP ON THE MCGILL DATASET

				MIREX		McGill	
Catagory	Sub	Authors	Approach	TRCO	ARCO	TRCO	ARCO
Pretrained	NMSD4	Y. Ni et al.	Bass/Treble Chroma, DBN, Hyper genre training	82.72	81.98	73.47	72.49
	NMSD3	Y. Ni et al.	Bass/Treble Chroma, DBN, Universal training	82.10	81.21	73.29	72.33
	NMSD2	Y. Ni et al.	Bass/Treble Chroma, DBN, McGill training	81.04	81.07	73.02	72.06
	NMSD1	Y. Ni et al.	Bass/Treble Chroma, DBN, MIREX training	83.51	81.73	72.39	71.40
	KO1	M. Khadkevich, M. Omologo	Time–frequency reassignment chroma, HMM	82.85	81.63	71.28	69.80
	CCSS1	Shen et al.	Chroma, duration–explicit HMM	79.40	77.91	67.36	66.19
Expert	PMP1	Johan Pauwels et al.	Bass/Treble Chroma, Chord/Key HMM	74.70	73.42	66.95	65.56
	PMP2	Johan Pauwels et al.	Bass/Treble Chroma, Chord/Key HMM	73.67	72.41	66.09	64.78
	PMP3	Johan Pauwels et al.	Bass/Treble Chroma, Chord/Key HMM	72.90	71.59	65.32	64.23
	DMW1	De Haas et al.	Bass/Treble NNLS Chroma, Template matching	73.68	71.99	64.33	62.49
	NG1	Nikolay Glazyrin	Chroma, Template matching	76.03	73.94	64.18	62.48

tures and a structured output predictor which accounted for interactions between neighboring frames was their method of choice. Pretrained and expert systems again failed to match the performances of train/test systems, although the OGF2 submission matched WEJ4 on the merged class. The introduction of Mauch's Dynamic Bayesian Network (submission MD) shows the first use of a complex graphical model for decoding, and attained the best score for a pretrained system, 0.712 overlap.

3) MIREX 2010: Moving to the evaluation of 2010, the evaluation database stabilized to a set of 217 tracks consisting of 179 tracks by The Beatles ('Revolution 9', Lennon/McCartney, was removed as it was deemed to have no harmonic content), 20 songs by Queen and 18 by Zweieck. We shall refer to this collection of audio and ground truth as the 'MIREX dataset'. Evaluation in this year was performed using major and minor triads with either the Total Relative Correct Overlap (TRCO) or Average Relative Correct Overlap (ARCO) summary.

This year saw the first example of a state of the art pretrained system–Mauch's MD1 system performed top in terms of both TRCO and ARCO, beating all other systems by use of an advanced Dynamic Bayesian Network and NNLS chroma. Interestingly, some train/test systems performed close to MD1 (Cho *et al.*, CWB1).

4) MIREX 2011: Data included in this year's evaluation was the again the standard MIREX dataset of 217 tracks. By now, performance had steadily risen from early work in 2008, but the possibility of models overfitting these data were significant. This issue was highlighted by the authors of the NMSD2 submission, who exploited the fact that the ground truth of all songs is known. Given this knowledge, the optimal strategy is to simply find a map between the audio of the signal to the ground truth dataset. This can be obtained by, for example, audio fingerprinting [94]. They did not achieve 100% because they shifted the ground truth data to match their audio collection.

This year, the expected trend of pretrained systems outperforming their train/test counterparts continued, with system KO1 obtaining a performance of 0.8285 TRCO, compared to the train/test CB3, which reached 0.8091.

F. MIREX 2012

The ACE task changed significantly in 2012, with the inclusion of an unknown test set of songs from McGill [13]. Participants submitted either expert systems or pretrained systems (there was no train/test evaluation this year) and were evaluated on both the known MIREX dataset of 217 songs and an additional 197 unknown billboard tracks. Results for both test sets are shown in Table IV.

The first thing to notice from Table IV is that performances on the McGill dataset are lower than on the MIREX dataset. This effect is due either to the McGill dataset being more varied and challenging, or because authors have overfitted on the MIREX dataset in previous years (and most likely a combination of the two). Top performance (73.47% TRCO, McGill dataset) was attained by Ni *et al.*, by using a complex training scheme which takes advantage of multiple genres in the training stage [87]. The same authors claimed the next three spots, with differing training schemes. Interestingly, it seems that the training scheme and data did not make much difference in overall performance, with hyper–genre training offering just 1.08 percentage points more than simple training on the MIREX data.

Submissions PMP1–PMP3 performed the best of the expert systems, reaching between 65.32% and 65.95% TRCO on the McGill dataset using bass and treble chromagrams and a key–chord HMM. A clear separation of expert vs knowledge–based systems emerges on consulting Table IV, showing that machine learning systems are not in fact overfitting the MIREX dataset as has been claimed [66]. It also seems that more complex models such as DBNs or Key–HMMs thrive in the unseen data test setting, with just 4 of the 11 systems now deploying a simple HMM.

G. Summary and Evolution of MIREX Performance

We show the evolution of MIREX performances as a series of box and whiskers plots in Fig. 13. From this figure, we see a slow steady improvement in performance from 2008 to 2011, although the rate of improvement diminishes as the years pass. It

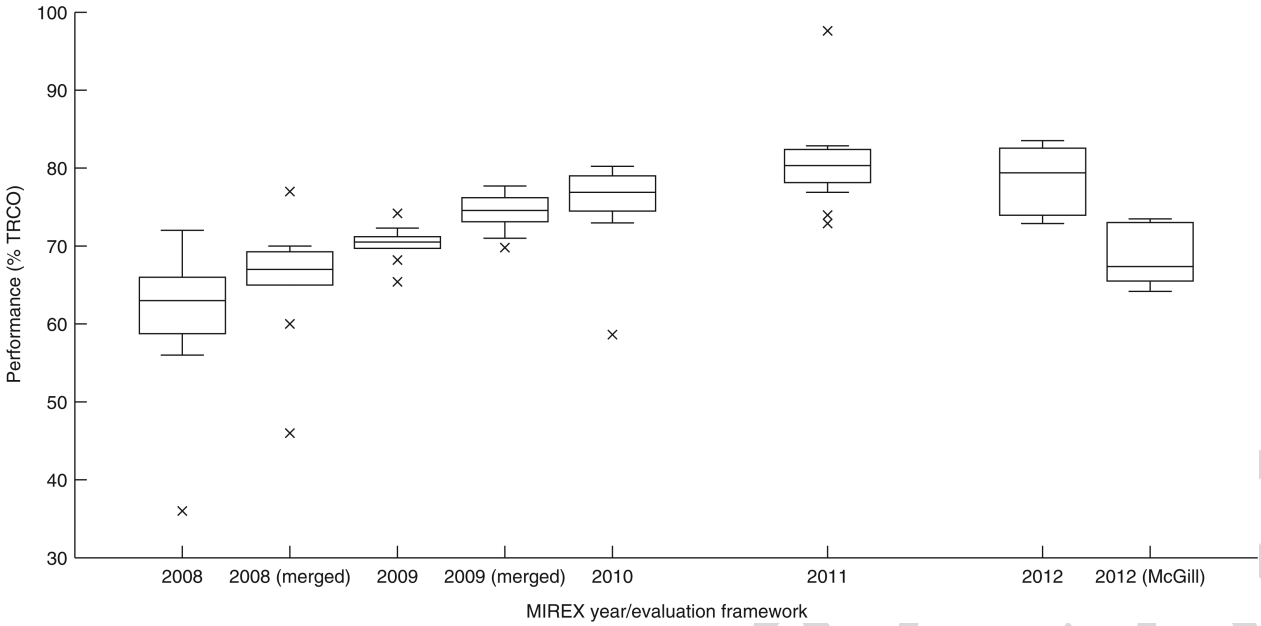

Fig. 13. Box plots and whiskers showing performance in the MIREX ACE task from years 2008 to 2012. Median performances are shown as the centers of the boxes, with height defined by 25th and 75th percentiles. Outliers which fall more than 1.5 times the Interquartile range are shown as crosses. Performance is measured using TRCO on the MIREX dataset (Beatles, Queen and Zweieck annotations) as they became available. Merged evaluations in 2008/2009 and performance on the McGill dataset in 2012 are shown offset to the right of their corresponding year.

is also clear from the figure that evaluation on the hidden McGill data now offers researchers an extra 10% 'headroom' to aim for before performance on this varied dataset reaches that on the MIREX dataset (recall that songs in this collection are heavily biased towards the pop group The Beatles).

VI. Software Packages

A number of online resources and software packages have been released in the past few years to address ACE. In this section we gather a non–comprehensive list of some of the most relevant contributions.

Since the turn of this century there has been gradual but steady improvement regarding available ACE implementations. For instance, Melisma Music Analyzer,[3] first released in 2000, offers in its last version C source code that uses probabilistic logic to identify metrical, stream and harmonic information from audio [95].

More recently, the labROSA ACE repository[4] compiled a collection of MATLAB algorithms for supervised chord estimation that were submitted to MIREX 2008, 2009 and 2010, from a simple Gaussian–HMM chord estimation system to an implementation of an advanced discriminative HMM. They perform chord estimation on the basis of beat–synchronous chromagrams.

Another useful piece of software is Chordino.[5] It provides an expert system based on NNLS Chroma [82]. This software has been used by web applications such as Yanno,[6] which allows users to extract the chords of YouTube videos.

At present, the state of the art ACE software is the aforementioned Harmony Progression Analyzer[7] (HPA). This is a key, chord and bass simultaneous estimation system that purely relies on machine learning techniques [36]. Included in the software are a pretrained model and scripts for retraining the model given new ground truth.

Other general purpose music software have become very relevant to chord estimation. Vamp[8] is an audio processing plugin system for plugins that extract descriptive information from audio data. Based on this technology, Sonic Annotator[9] offers a tool for feature extraction and annotation of audio files. It will run available Vamp plugins on a wide range of audio file types, and can write the results in a selection of formats. Finally, Sonic Visualiser[10] provides an application for viewing and analyzing the contents of music audio files [96]. Sonic visualiser and Chordino have the advantage of allowing predicted chord sequences to be visualized, allowing users to play along intuitively with the analyzed music.

VII. Impact within Music Information Retrieval

Many of the modelling techniques presented in this paper are of interest not only for ACE, but also for MIR tasks that involve sequence labelling. We briefly discuss some of these options in the current Section.

Chords define the tonal backbone of western music, and as such it is likely that any MIR task which is based around pitch classes will benefit from an understanding of chords. Existing/proposed ways in which estimated chord sequences may be used in example tasks are discussed below.

[3]http://theory.esm.rochester.edu/temperley/melisma2/

[4]http://labrosa.ee.columbia.edu/projects/chords/

[5]http://isophonics.net/nnls-chroma

[6]http://yanno.eecs.qmul.ac.uk/

[7]https://patterns.enm.bris.ac.uk/hpa-software-package

[8]http://vamp-plugins.org/

[9]http://omras2.org/SonicAnnotator

[10]http://www.sonicvisualiser.org/

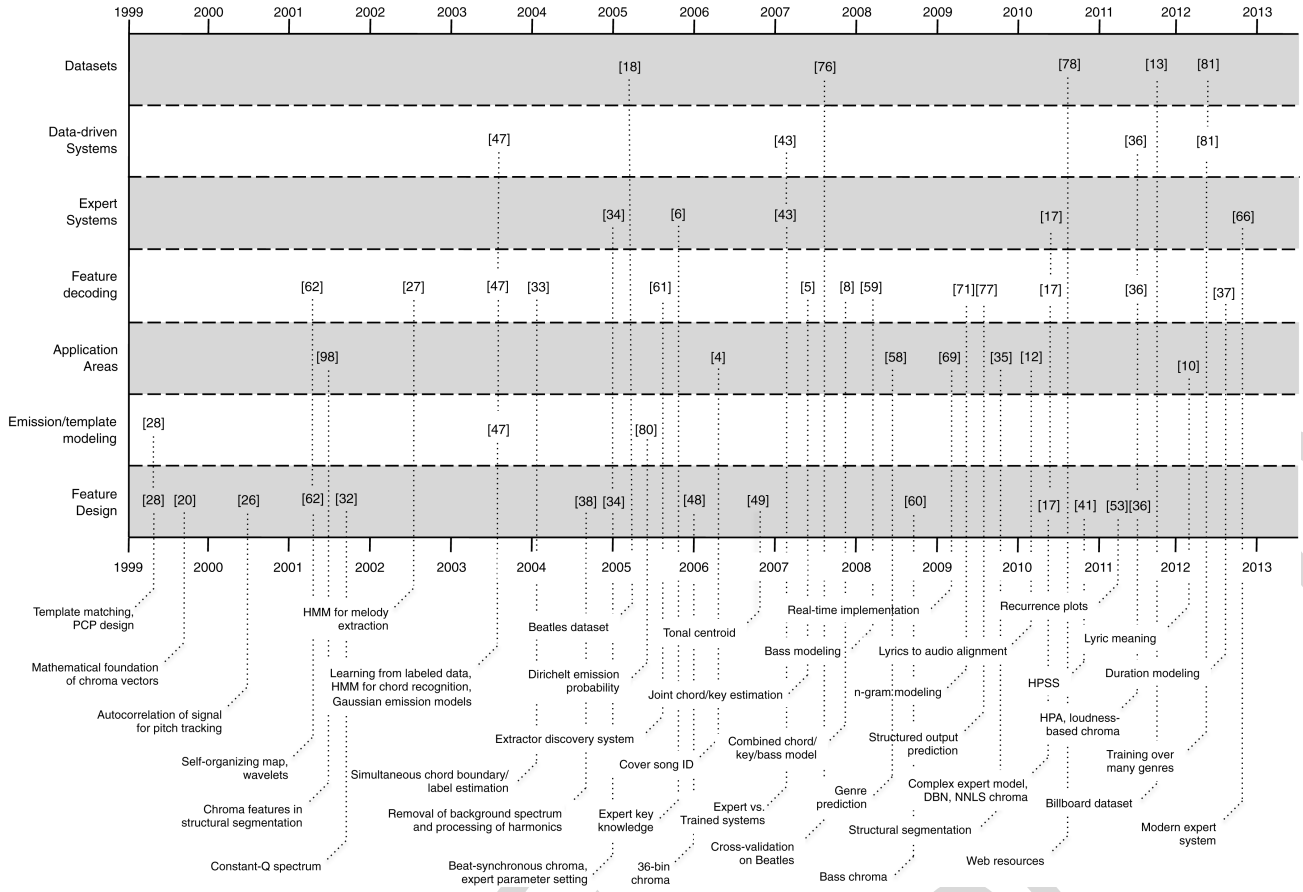

Fig. 14. Major publications in the field of ACE. Publication year increases along the horizontal axis, with research theme on the vertical axis. Main contributions are also annotated under publication year. Numbers in brackets ([·]) refer to reference number.

The development of tonal key estimation has proceeded in parallel to ACE, which is unsurprising given their intertwined nature. This was verified in Section II and III, where we showed that chromagram feature matrices and HMM architectures were developed simultaneously in both domains through the years. Recently, sophisticated approaches have begun to incorporate both chords and keys into a single model [97], further blurring the lines between the two domains [36], [82].

Structural segmentation (identifying verse, bridge, chorus etc) is another MIR task which has seen many advances as a result of the developments of ACE, although here the focus is generally on the use of chromagram features and not modelling techniques [98]. Briefly, the distance between all pairs of chromagram frames can be collected in a self–similarity matrix, where it is hoped that high–similarity off–diagonal stripes will correspond to repeated sections of a piece of music (see [99] for an excellent review).

One area in which ACE could have a major impact is in the detection of mood from audio. Major chords are often thought of as 'happy–sounding', minor chords as 'sad–sounding' with diminished chords indicating tension or unpleasantness, which was verified in experimental work on both musicians and non-musicians by Pallesen *et al.* [100]. However, most mood detection work is conducted at the song level, the notable exception being the work by Schmidt *et al.* [101]. A fruitful area of research may therefore be the investigation of correlations between predicted chord sequences and dynamic mood modelling and indeed, results by Cheng and collaborators [102] indicate that chordal features improve mood classification.

Music recommendation and playlisting are two MIR tasks which have music similarity at their core. The task is to construct novel song recommendation or songs, given a query instance. Two approaches have dominated the literature in these tasks: collaborative filtering [103], which ranks queries based on a database of users who have made similar queries; and content–based retrieval, where the goal is to find songs with similar audio features to the query [104]. Many existing techniques are based on *Mel–Frequency Cepstrum Coefficients*, which attempt to capture the instrumentation and/or timbre of the pieces. However, we have yet to see an application of chord sequences in this research challenge. To account for the time–varying nature of the predicted sequences, one would have to use summary statistics such as percentage of major chords, or a more general distribution over chord types.

VIII. Conclusions and Future work

In this article, we discussed the task of Automatic Chord Estimation (ACE) from polyphonic western pop music. We listed the main contributions available in the literature, concentrating on feature extraction, modelling, evaluation, and model

TABLE V
CHRONOLOGICAL SUMMARY OF ADVANCES IN ACE FROM AUDIO, YEARS 1999–2012, SHOWING YEAR OF PUBLICATION, REFERENCE NUMBER, TITLE AND KEY CONTRIBUTION(S) TO THE FIELD

Year	Author(s)	Title	Key Contribution(s)
1999	[28]	Realtime Chord Recognition of Musical Sound: a System Common Lisp Music	PCP vector, template matching smoothing
	[20]	Mathematical Representation of Joint Time–chroma Distributions	Mathematical foundation of chromagrams feature vectors
2000	[26]	Techniques for Automatic Music Transcription	Autocorrelation function for pitch tracking
2001	[62]	Multi–timbre Chord Classification using Wavelet Transform and Self–Organized Neural Networks	Wavelets, Self–Organising–Map
	[32]	Identification of Musical Chords using Constant–Q spectra	Constant–Q Spectrum
	[98]	To Catch a Chorus: Using Chroma–based Representations for Thumbnailing	Chroma features for audio structural segmentation
2002	[27]	Automatic Transcription of Piano Music	HMM for melody extraction
2003	[47]	Chord Segmentation and Recognition using EM–Trained Hidden Markov Models	HMM for chord estimation, Gaussian emission probabilities, training from labelled data
2004	[33]	Automatic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries	Simultaneous boundary/label detection
	[38]	Musical Key Extraction from Audio	Removal of background spectrum and processing of harmonics
2005	[34]	A Robust Mid–Level Representation for Harmonic Content in Music Signals	Beat–synchronous chroma, expert parameter knowledge
	[48]	Automatic Chord Identification using a Quantised Chromagram	36–bin chromagram tuning algorithm
	[61]	Automatic X Traditional Descriptor Extraction: the Case of Chord Recognition	Use of Extractor Discovery system
	[6]	Key, Chord, and Rhythm Tracking of Popular Music Recordings	Expert key knowledge
	[80]	Learning Harmonic Relationships in Digital Audio with Dirichlet–based Hidden Markov Models	Dirichlet emission probability model
	[19]	Symbolic Representation of Musical chords: A Proposed syntax for Text Annotations	Textual notation of chords, Beatles dataset
2006	[4]	The Song Remains the Same: Identifying versions Transposed by Key Versions of the Same Piece using Tonal Descriptors	Cover–song identification using chroma vectors
	[42]	Automatic Chord Recognition from Audio using Enhanced Pitch Class Profile	Removal of harmonics to match PCP templates
	[49]	Detecting Harmonic Change in Musical Audio	Tonal centroid feature
2007	[5]	A Probabilistic Framework for Tonal Key and Chord Recognition	Rigorous framework for joint key/chord estimation
	[76]	A Cross–Validated Study of Modelling Strategies for Automatic Chord Recognition in Audio	Cross–validation on Beatles data, Conditional Random Fields
	[43]	Large–Scale study of Chord Estimation Algorithms Based on Chroma Representation and HMM	Comparative study of expert vs. trained systems
	[8]	Automatic Chord Detection Incorporating Beat and Key Detection	Combined key, beat and chord model
	[57]	A Unified System for Chord Transcription and Key Extraction using Hidden Markov Models	Key–specific HMMs, tonal centroid in key detection
2008	[59]	Automatic Chord Recognition based on Probabilistic Integration of Chord Transition and bass Pitch Estimation	Integration of bass pitch information
	[54]	Simultaneous Estimation of Chord Progression and Downbeats from an Audio File	Simultaneous beat/chord estimation
	[45]	A Novel Chroma Representation of Polyphonic Music Based on Multiple Pitch Tracking Techniques	Simultaneous background spectra & harmonic removal
	[58]	A System for Automatic Chord Transcription from Audio Using Genre–Specific Hidden Markov Models	Genre–specific HMMs
	[60]	A Discrete Mixture Model for Chord Labelling	Bass chromagram
2009	[71]	Robust Modelling of Musical Chord Sequences using Probabilistic N–Grams	n–gram language model
	[69]	Real–time Implementation of HMM–based Chord Estimation in Musical Audio	Real–time chord estimation system
	[64]	Template–Based Chord Recognition: Influence of the Chord Types	Comparison of template distance metrics and smoothing techniques
	[2]	Automatic Generation of Lead Sheets from Polyphonic Music Signals	Polyphonic extraction of lead sheets
	[77]	Structured Prediction Models for Chord Transcription of Music Audio	SVMstruct, incorporating future frame information
	[40]	Minimum Classification Error Training to Improve Isolated Chord Recognition	Harmonic and Percussive Source Separation (HPSS)
	[35]	Using Musical Structure to Enhance Automatic Chord Transcription	Structural segmentation as an additional information source
	[74]	Use of Hidden Markov Models and Factored Language Models for Automatic Chord Recognition	Factored language model
	[18]	Influences of Signal Processing, Tone Profiles, and Chord Progressions on a Model for Estimating the Musical Key from Audio	In–depth study on integrated chord and key dependencies
2010	[17]	Automatic Chord Transcription from Audio using Computational Models of Musical Context	DBN model, NNLS chroma
	[41]	HMM–based approach for Automatic Chord Detection using Refined Acoustic Features	HPSS with additional post–processing
	[55]	Exploring Common Variations in State of the Art Chord Recognition Systems	Comparison of pre and post–filtering techniques and models
	[92]	A Multi–perspective Evaluation Framework for Chord Recognition	Visualisation of evaluation techniques
	[12]	Lyrics–to–audio Alignment and Phrase–level Segmentation using Incomplete Internet–style Chord Annotations	Chord sequences in lyrics alignment
	[78]	Using online chord databases to enhance chord recognition	Use of partially–labelled data web data
	[82]	Approximate note transcription for the improved identification of difficult chords	NNLS chroma
2011	[13]	An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis	Billboard Hot 100 dataset of chord annotations
	[79]	Analysing Chroma Feature Types for Automated Chord Recognition	Comparison of modern chromagram types
	[15]	Guitar Tab Mining, Analysis and Ranking	Web–based chord labels
	[36]	An end–to–end machine learning system for harmonic analysis of music	First top-performing machine learning–based system
	[53]	A Feature Smoothing Method for Chord Recognition Using Recurrence Plots	Recurrence plot for smoothing
	[75]	A Vocabulary–Free Infinity–Gram Model for Non–parametric Bayesian Chord Progression Analysis	Infinity–gram language model
2012	[81]	Using Hyper–genre Training to Explore Genre Information for Automatic Chord Estimation	Training across multiple genres and subgenres
	[97]	Modeling Chord and Key Structure with Markov Logic	Markov logic to encapsulate expert knowledge into unified chord and key detection
	[66]	Improving Audio Chord Transcription by Exploiting Harmonic and Metric Knowledge	Use of music theory over an HMM
	[37]	Chord Recognition Using Duration–explicit Hidden Markov Models	Chord duration modelling
	[10]	Inferring Chord Sequence Meanings via Lyrics: Process and Evaluation	Paired analysis of lyrics and chords
	[105]	Unsupervised Chord–Sequence Generation from an Audio Example	Synthesis of music from chord sequence

training/datasets. We discovered that the dominant set up is to extract chromagrams directly from audio, and label using a Hidden Markov Model with Viterbi decoding.

Several advances have been made in the feature extraction and modelling stage, such that features now include aspects such as tuning, smoothing, removal of harmonics and loudness perceptual weighting. Models extend beyond the 1st order HMM to include duration–explicit HMMs, key–chord HMMs, and Dynamic Bayesian Networks. Training of these models is conducted using a combination of expert musical knowledge and parameter estimation from fully or partially–labelled data sources.

Upon investigating the annual benchmarking system MIREX, we found that a slow and steady increase in performance from 69% to 82.85% on a set of (up to) 217 tracks by The Beatles, Queen and Zweieck, although there is some

evidence that overfitting on this dataset is occurring. In the most recent evaluation, we saw scores above 73% for completely unseen data.

In suggesting areas for future work, we believe that a more towards a more inclusive evaluation strategy including the evaluation of complex chords will be fruitful. This will present some challenges, as it is not immediately obvious how one should score a prediction of, say C major7/E against a ground truth of C major. However, given the sophistication of current models and the amount of data available for training and testing, we think this will yield valuable results. In addition to this, major/minor ACE systems are competent enough we feel that they are ready to be fed more readily into application areas such as mood detection, cover song analysis, music recommendation and structure analysis.

Appendix

A concise chronological review of the associated literature together with the main contributions of each work is shown in Table V. We also provide a visualization of the advances made in various aspects of ACE in Fig. 14.

References

[1] Various, The Real Book6th ed. Milwaukee, WI, USA, Hal Leonard Corp., 2004.

[2] J. Weil, T. Sikora, J. Durrieu, and G. Richard, "Automatic generation of lead sheets from polyphonic music signals," in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 603–608.

[3] D. Ellis and G. Poliner, "Identifying 'cover songs' wisth chroma features and dynamic programming beat tracking," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2007, pp. 1429–1433.

[4] E. Gómez and P. Herrera, "The song remains the same: Identifying versions of the same piece using tonal descriptors," in *Proc. 7th Int. Soc. Music Inf. Retrieval*, 2006, pp. 180–185.

[5] B. Catteau, J. Martens, and M. Leman, "A probabilistic framework for audio-based tonal key and chord recognition," in *Proc. 30th Annu. Conf. Gesellschaft fur Klassikation*, 2007, pp. 637–644, Springer.

[6] A. Shenoy and Y. Wang, "Key, chord, and rhythm tracking of popular music recordings," *J. Comput. Music*, vol. 29, no. 3, pp. 75–86, 2005.

[7] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 2, pp. 291–301, Feb. 2008.

[8] V. Zenz and A. Rauber, "Automatic chord detection incorporating beat and key detection," in *Proc. IEEE Int. Conf. Signal Process. Commun.*, 2007, pp. 1175–1178.

[9] C. Perez-Sancho, D. Rizo, and J. Inesta, "Genre classification using chords and stochastic language models," *Connect. Sci.*, vol. 21, no. 2-3, pp. 145–159, 2009.

[10] T. O'Hara, "Inferring the meaning of chord sequences via lyrics," in *Proc. 2nd Worskshop Music Recommendation Discovery collocated with ACM-RecSys*, 2011, p. 34.

[11] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 200–210, Jan. 2012.

[12] M. Mauch, H. Fujihara, and M. Goto, "Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations," in *Proc. 7th Sound Music Comput. Conf.*, 2010, pp. 9–16.

[13] J. Burgoyne, J. Wild, and I. Fujinaga, "An expert ground truth set for audio chord recognition and music analysis," in *Proc. Int. Conf. Music Inf. Retrieval*, 2011, pp. 633–638.

[14] M. McVicar, Y. Ni, R. Santos-Rodriguez, and T. De Bie, "Using online chord databases to enhance chord recognition," *J. New Music Res.*, vol. 40, no. 2, pp. 139–152, 2011.

[15] R. Macrae and S. Dixon, "Guitar tab mining, analysis and ranking," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, 2011, pp. 453–458.

[16] C. Harte, "Towards automatic extraction of harmony information from music signals," Ph.D. dissertation, Univ. of London, London, U.K., 2010.

[17] M. Mauch, "Automatic chord transcription from audio using computational models of musical context," Ph.D. dissertation, Queen Mary Univ. of London, London, U.K., 2010.

[18] K. Noland and M. Sandler, "Influences of signal processing, tone profiles, and chord progressions on a model for estimating the musical key from audio," *J. Comput. Music*, vol. 33, no. 1, pp. 42–56, 2009.

[19] C. Harte, M. Sandler, S. Abdallah, and E. Gómez, "Symbolic representation of musical chords: A proposed syntax for text annotations," in *Proc. Int. Conf. Music Inf. Retrieval*, 2005, pp. 66–71.

[20] G. Wakefield, "Mathematical representation of joint time-chroma distributions," in *Proc. Int. Symp. Opt. Sci., Eng. Instrum.*, 1999, vol. 99, pp. 18–23.

[21] R. Shepard, "Circularity in judgments of relative pitch," *J. Acoust. Soc. Amer.*, vol. 36, p. 2346, 1964.

[22] C. Chafe, *Techniques for Note Identification in Polyphonic Music*. Stanford, CA, USA: CCRMA, Dept. of Music, Stanford Univ., 1985.

[23] C. Chafe and D. Jaffe, "Source separation and note identification in polyphonic music," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1986, vol. 11, pp. 1289–1292.

[24] K. Martin, A blackboard system for automatic transcription of simple polyphonic music Mass. Inst. of Technol. Media Lab. Perceptual Comput. Sec., Tech. Rep., no. 385, 1996.

[25] K. Kashino and N. Hagita, "A music scene analysis system with the MRF-based information integration scheme," in *Proc. 13th Int. Conf. Pattern Recogn.*, 1996, vol. 2, pp. 725–729.

[26] J. Bello, G. Monti, and M. Sandler, "Techniques for automatic music transcription," in *Proc. Int. Symp. Music Inf. Retrieval*, 2000, pp. 23–25.

[27] C. Raphael, "Automatic transcription of piano music," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, 2002, vol. 2, pp. 13–17.

[28] T. Fujishima, "Realtime chord recognition of musical sound: A system using Common Lisp Music," in *Proc. Int. Comput. Music Conf.*, 1999, pp. 464–467.

[29] D. Deutsch, *The Psychology of Music*. New York, NY, USA: Academic, 1999.

[30] W. Heisenberg, "Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik," *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 43, no. 3, pp. 172–198, 1927.

[31] J. Brown, "Calculation of a Constant–Q spectral transform," *J. Acoust. Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.

[32] S. Nawab, S. Ayyash, and R. Wotiz, "Identification of musical chords using Constant–Q spectra," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 5, pp. 3373–3376.

[33] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. Okuno, "Automatic chord transcription with concurrent recognition of chord symbols and boundaries," in *Proc. 5th Int. Conf. Music Inf. Retrieval*, 2004.

[34] J. Bello and J. Pickens, "A robust mid-level representation for harmonic content in music signals," in *Proc. 6th Int. Soc. Music Inf. Retrieval*, 2005, pp. 304–311.

[35] M. Mauch, K. Noland, and S. Dixon, "Using musical structure to enhance automatic chord transcription," in *Proc. 10th Int. Conf. Music Inf. Retrieval*, 2009, pp. 231–236.

[36] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "An end-to-end machine learning system for harmonic analysis of music," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 6, pp. 1771–1783, Aug. 2012.

[37] R. Chen, W. Shen, A. Srinivasamurthy, and P. Chordia, "Chord recognition using duration-explicit hidden markov models," in *Proc. 13th Int. Soc. Music Inf. Retrieval*, 2012, pp. 445–450.

[38] S. Pauws, "Musical key extraction from audio," in *Proc. 5th Int. Soc. Music Inf. Retrieval*, 2004, vol. 4, pp. 66–69.

[39] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," in *Proc. Euro. Signal Process. Conf.*, 2008, pp. 445–450.

[40] J. Reed, Y. Ueda, S. Siniscalchi, Y. Uchiyama, S. Sagayama, and C. Lee, "Minimum classification error training to improve isolated chord recognition," in *Proc. 10th Int. Soc. Music Inf. Retrieval*, 2009, pp. 609–614.

[41] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama, "HMM-based approach for automatic chord detection using refined acoustic features," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2010, pp. 5518–5521.

[42] K. Lee and M. Slaney, "Automatic chord recognition from audio using an HMM with supervised learning," in *Proc. 7th Int. Soc. Music Inf. Retrieval*, 2006, pp. 133–137.

[43] H. Papadopoulos and G. Peeters, "Large-scale study of chord estimation algorithms based on chroma representation and HMM," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, 2007, pp. 53–60.

[44] M. Mauch and S. Dixon, "Simultaneous estimation of chords and musical context from audio," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1280–1289, Aug. 2010.

[45] M. Varewyck, J. Pauwels, and J. Martens, "A novel chroma representation of polyphonic music based on multiple pitch tracking techniques," in *Proc. 16th Int. Conf. Multimedia*, 2008, pp. 667–670.

[46] C. Lawson and R. Hanson, *Solving Least Squares Problems*. Philadelphia, PA, USA: Soc. for Ind. Math., 1995, vol. 15.

[47] A. Sheh and D. Ellis, "Chord segmentation and recognition using em-trained Hidden Markov Models," in *Proc. 4th Int. Soc. Music Inf. Retrieval*, 2003, pp. 183–189.

[48] C. Harte and M. Sandler, "Automatic chord identification using a quantised chromagram," in *Proc. Audio Eng. Soc.*, 2005, pp. 291–301.

[49] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proc. 1st Workshop Audio Music Comput. Multimedia*, 2006, pp. 21–26.

[50] M. T. Smith, *Audio engineer's reference book*. Abingdon, U.K.: Focal Press, 1999.

[51] O. Lartillot and P. Toiviainen, "A MATLAB toolbox for musical feature extraction from audio," in *Proc. 10th Int. Conf. Digital Audio Effects*, Bordeaux, France, 2007, pp. 237–244.

[52] M. Goto and Y. Muraoka, "Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions," *Speech Commun.*, vol. 27, no. 3, pp. 311–335, 1999.

[53] T. Cho and J. Bello, "A feature smoothing method for chord recognition using recurrence plots," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, 2011, pp. 651–656.

[54] H. Papadopoulos and G. Peeters, "Simultaneous estimation of chord progression and downbeats from an audio file," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 121–124.

[55] T. Cho, R. Weiss, and J. Bello, "Exploring common variations in state of the art chord recognition systems," in *Proc. Sound Music Comput. Conf.*, 2010, vol. 1.

[56] E. Chew, "Towards a mathematical model of tonality," Ph.D. dissertation, Mass. Inst. of Technol., Cambridge, MA, USA, 2000.

[57] K. Lee and M. Slaney, "A unified system for chord transcription and key extraction using Hidden Markov Models," in *Proc. Int. Conf. Music Inf. Retrieval*, 2007, pp. 245–250.

[58] K. Lee, "A system for automatic chord transcription from audio using genre-specific Hidden Markov Models," *Adaptive Multimedial Retrieval: Retrieval, User, and Semantics*, pp. 134–146, 2008.

[59] K. Sumi, K. Itoyama, K. Yoshii, K. Komatani, T. Ogata, and H. Okuno, "Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation," in *Proc. Int. Conf. Music Inf. Retrieval*, 2008, pp. 39–44.

[60] M. Mauch and S. Dixon, "A discrete mixture model for chord labelling," in *Proc. 9th Int. Conf. Music Inf. Retrieval*, 2008, pp. 45–50.

[61] G. Cabral, F. Pachet, J. Briot, and S. Paris, "Automatic X traditional descriptor extraction: The case of chord recognition," in *Proc. 6th Int. Conf. Music Inf. Retrieval*, 2005, pp. 444–449.

[62] B. Su and S. Jeng, "Multi-timbre chord classification using wavelet transform and self-organized map neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 5, pp. 3377–3380.

[63] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[64] L. Oudre, Y. Grenier, and C. Févotte, "Template-based chord recognition: Influence of the chord types," in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 153–158.

[65] L. Oudre, Y. Grenier, and C. Févotte, "Chord recognition using measures of fit, chord templates and filtering methods," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2009, pp. 9–12.

[66] W. de Haas, J. Magalhães, and F. Wiering, "Improving audio chord transcription by exploiting harmonic and metric knowledge," in *Proc. 13th Int. Soc. Music Inf. Retrieval*, 2012.

[67] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[68] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[69] T. Cho and J. Bello, "Real-time implementation of HMM-based chord estimation in musical audio," in *Proc. Int. Comput. Music Conf.*, 2009, pp. 16–21.

[70] A. Stark and M. Plumbley, "Real-time chord recognition for live performance," in *Proc. Int. Comput. Music Conf.*, 2009, vol. 8, pp. 585–593.

[71] R. Scholz, E. Vincent, and F. Bimbot, "Robust modelling of musical chord sequences using probabilistic N-grams," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2009, pp. 53–56.

[72] E. Unal, P. Georgiou, S. Narayanan, and E. Chew, "Statistical modeling and retrieval of polyphonic music," in *Proc. 9th IEEE Workshop Multimedia Signal Process.*, 2007, pp. 405–409.

[73] M. Mauch, S. Dixon, and C. Harte, "Discovering chord idioms through Beatles and Real Book songs," in *Proc. 8th Int. Soc. Music Inf. Retrieval*, 2007, pp. 255–258.

[74] M. Khadkevich and M. Omologo, "Use of hidden Markov models and factored language models for automatic chord recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2009, pp. 561–566.

[75] K. Yoshii and M. Goto, "A vocabulary-free infinity-gram model for nonparametric Bayesian chord progression analysis," in *Proc. 12th Int. Soc. Music Inf. Retrieval Conf.*, 2011, pp. 645–650.

[76] J. Burgoyne, L. Pugin, C. Kereliuk, and I. Fujinaga, "A cross-validated study of modelling strategies for automatic chord recognition in audio," in *Proc. 8th Int. Conf. Music Inf. Retrieval*, 2007, p. 251.

[77] A. Weller, D. Ellis, and T. Jebara, "Structured prediction models for chord transcription of music audio," in *Proc. Int. Conf. Mach. Learn. Appl.*, 2009, pp. 590–595.

[78] M. McVicar, Y. Ni, R. Santos-Rodriguez, and T. De Bie, "Using online chord databases to enhance chord recognition," *J. New Music Res.*, vol. 40, no. 2, pp. 139–152, 2011.

[79] N. Jiang, P. Grosche, V. Konz, and M. Müller, "Analyzing chroma feature types for automated chord recognition," in *Proc. 42nd Audio Eng. Soc. Conf.*, 2011, pp. 1–10.

[80] J. Burgoyne and L. Saul, "Learning harmonic relationships in digital audio with Dirichlet-based Hidden Markov Models," in *Proc. Int. Conf. Music Inf. Retrieval*, 2005, pp. 438–443.

[81] Y. Ni, M. Mcvicar, R. Santos-Rodrguez, and T. De Bie, "Using hypergenre training to explore genre information for automatic chord estimation," in *Proc. 13th Int. Soc. Music Inf. Retrieval*, 2012, pp. 109–114.

[82] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. 11th Int. Soc. Music Inf. Retrieval Conf.*, 2010, pp. 135–140.

[83] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *J. Comput. Music*, vol. 28, no. 2, pp. 63–76, 2004.

[84] W. de Haas and J. Burgoyne, Parsing the Billboard chord transcriptions Univ. of Utrecht, Utrecht, The Netherlands, Tech. Rep., 2012.

[85] F. Lerdahl, *Tonal pitch space*. New York, NY, USA: Oxford Univ. Press, 2005.

[86] D. Ellis and A. Weller, "The 2010LabROSA chord recognition system," in *Proc. 11th Int. Soc. Music Inf. Retrieval (MIREX submission)*, 2010.

[87] Y. Ni, M. McVicar, R. Santos-Rodrguez, and T. De Bie, "Harmony progression analyzer for MIREX 2012," in *Proc. 13th Int. Soc. Music Inf. Retrieval (MIREX submission)*, 2012.

[88] R. Macrae and S. Dixon, "A guitar tablature score follower," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2010, pp. 725–726.

[89] M. Barthet, A. Anglade, G. Fazekas, S. Kolozali, and R. Macrae, "Music recommendation for music learning: Hotttabs, a multimedia guitar tutor," in *Proc. 2nd Worskshop Music Recommendation Discovery collocated with ACM-RecSys*, 2011, p. 7.

[90] M. McVicar and T. De Bie, "Enhancing chord recognition accuracy using web resources," in *Proc. 3rd Int. Workshop Mach. Learn. Music*, 2010, pp. 41–44.

[91] J. Pauwels and G. Peeters, "Evaluating automatically estimated chord sequences," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 749–753.

[92] V. Konz, M. Müller, and S. Ewert, "A multi-perspective evaluation framework for chord recognition," in *Proc. 11th Int. Conf. Music Inf. Retrieval*, 2010, pp. 9–14.

[93] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie, "Understanding effects of subjectivity in measuring chord estimation accuracy," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 12, pp. 2607–2615, Dec. 2013.

[94] A. Wang and J. Smith, "System and methods for recognizing sound and music signals in high noise and distortion," U.S. patent 6,990,453, Jan. 24, 2006, III.

[95] D. Temperley, "A unified probabilistic model for polyphonic music analysis," *J. New Music Res.*, vol. 38, no. 1, pp. 3–18, 2009.
[96] C. Cannam, C. Landone, and M. Sandler, "Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proc. ACM Multimedia 2010 Int. Conf.*, Oct. 2010, pp. 1467–1468.
[97] H. Papadopoulos and G. Tzanetakis, "Modeling chord and key structure with Markov logic," in *Proc. 13th Int. Soc. Music Inf. Retrieval*, 2012, pp. 121–126.
[98] M. Bartsch and G. Wakefield, "To catch a chorus: Using chroma-based representations for audio thumbnailing," in *Proc. Appl. Signal Process. Audio Acoust.*, 2001, pp. 15–18.
[99] J. Paulus, M. Müller, and A. Klapuri, "State of the art report: Audio-based music structure analysis," in *Proc. 11th Int. Soc. Music Inf. Retrieval Conf.*, 2010, pp. 625–36.
[100] K. J. Pallesen, E. Brattico, C. Bailey, A. Korvenoja, J. Koivisto, A. Gjedde, and S. Carlson, "Emotion processing of major, minor, and dissonant chords," *Ann. New York Acad. Sci.*, vol. 1060, no. 1, pp. 450–453, 2005.
[101] E. Schmidt and Y. Kim, "Modeling musical emotion dynamics with Conditional Random Fields," in *Proc. 12th Int. Soc. Music Inf. Retrieval*, 2011, pp. 777–782.
[102] H.-T. Cheng, Y.-H. Yang, Y.-C. Lin, I.-B. Liao, and H. H. Chen, "Automatic chord recognition for music classification and retrieval," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2008, pp. 1505–1508.
[103] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, Jan./Feb. 2003.
[104] B. McFee and G. Lanckriet, "Metric learning to rank," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 775–782.
[105] K. Kosta, M. Marchini, and H. Purwins, "Unsupervised chord-sequence generation from an audio example," in *Proc. 13th Int. Soc. Music Inf. Retrieval*, 2012, pp. 481–486.

Matt McVicar received the Ph.D. in Complexity Sciences in 2013 from the University of Bristol, where his focus was the automatic estimation of chords from polyphonic audio via machine learning methods. During his PhD he also worked on the interaction and correlations between different domains, notably audio, lyrics and social tags. In 2012 he was awarded a USUK Fulbright scholarship to conduct Music Information Retrieval research at the Laboratory for the Recognition and Organization of Speech and Audio (LabROSA) at Columbia University. His is currently a postdoctoral researcher at the Media Interaction Group at the National Institution of Advanced Industrial Science and Technology and has research interests including the automated analysis of aspects musical harmony, lyrics, and mood using data-driven approaches.

Raúl Santos-Rodríguez received the Ph.D. degree in Telecommunication Engineering from Universidad Carlos III de Madrid, Spain in 2011. He is currently a data scientist at Genexies Mobile and a Research Fellow at the Intelligent Systems Lab, University of Bristol. His main research interests include machine learning, Bayesian methods and their applications to signal processing and music information retrieval.

Yizhao Ni is a Research Associate at Cincinnati Children's Hospital Medical Center and a Visiting Fellow at the Department of Engineering Mathematics, University of Bristol. He completed his Ph.D. on machine learning for machine translation in 2010 at University of Southampton, after which he worked as a postdoctoral fellow at the University of Bristol. His current research interests lie in the development and application of machine learning methods to biomedical informatics, natural language processing and music information retrieval.

Tijl De Bie is a Reader in Computational Pattern Analysis at the University of Bristol, where he was first appointed as a Lecturer in January 2007. Before that, he was a research assistant at the University of Leuven and the University of Southampton. He completed his PhD on machine learning and advanced optimization techniques in 2005 at the University of Leuven, during which he spent research visits in U.C. Berkeley and U.C. Davis. His current research interests include the development of theoretical foundations for exploratory data mining, as well as the application of data mining and machine learning techniques to music information retrieval, web and text mining, and bioinformatics.