Probabilistic and Logic-Based Modelling of Harmony

Simon Dixon, Matthias Mauch, and Amélie Anglade

Centre for Digital Music,
Queen Mary University of London,
Mile End Rd, London E1 4NS, UK
simon.dixon@eecs.qmul.ac.uk
http://www.eecs.qmul.ac.uk/~simond

Abstract. Many computational models of music fail to capture essential aspects of the high-level musical structure and context, and this limits their usefulness, particularly for musically informed users. We describe two recent approaches to modelling musical harmony, using a probabilistic and a logic-based framework respectively, which attempt to reduce the gap between computational models and human understanding of music. The first is a chord transcription system which uses a high-level model of musical context in which chord, key, metrical position, bass note, chroma features and repetition structure are integrated in a Bayesian framework, achieving state-of-the-art performance. The second approach uses inductive logic programming to learn logical descriptions of harmonic sequences which characterise particular styles or genres. Each approach brings us one step closer to modelling music in the way it is conceptualised by musicians.

Keywords: Chord transcription, inductive logic programming, musical harmony.

1 Introduction

Music is a complex phenomenon. Although music is described as a "universal language", when viewed as a paradigm for communication it is difficult to find agreement on what constitutes a musical message (is it the composition or the performance?), let alone the meaning of such a message. Human understanding of music is at best incomplete, yet there is a vast body of knowledge and practice regarding how music is composed, performed, recorded, reproduced and analysed in ways that are appreciated in particular cultures and settings. It is the computational modelling of this "common practice" (rather than philosophical questions regarding the nature of music) which we address in this paper. In particular, we investigate harmony, which exists alongside melody, rhythm and timbre as one of the fundamental attributes of Western tonal music.

Our starting point in this paper is the observation that many of the computational models used in the music information retrieval and computer music research communities fail to capture much of what is understood about music.

K. Jensen et al. (Eds.): CMMR 2010, LNCS 6684, pp. 1–19, 2011.

Two examples are the bag-of-frames approach to music similarity [5], and the periodicity pattern approach to rhythm analysis [13], which are both independent of the order of musical notes, whereas temporal order is an essential feature of melody, rhythm and harmonic progression. Perhaps surprisingly, much progress has been made in music informatics in recent years[1], despite the naivete of the musical models used and the claims that some tasks have reached a "glass ceiling" [6].

The continuing progress can be explained in terms of a combination of factors: the high level of redundancy in music, the simplicity of many of the tasks which are attempted, and the limited scope of the algorithms which are developed. In this regard we agree with [14], who review the first 10 years of ISMIR conferences and list some challenges which the community "has not fully engaged with before". One of these challenges is to "dig deeper into the music itself", which would enable researchers to address more musically complex tasks; another is to "expand ... musical horizons", that is, broaden the scope of MIR systems.

In this paper we present two approaches to modelling musical harmony, aiming at capturing the type of musical knowledge and reasoning a musician might use in performing similar tasks. The first task we address is that of chord transcription from audio recordings. We present a system which uses a high-level model of musical context in which chord, key, metrical position, bass note, chroma features and repetition structure are integrated in a Bayesian framework, and generates the content of a "lead-sheet" containing the sequence of chord symbols, including their bass notes and metrical positions, and the key signature and any modulations over time. This system achieves state-of-the-art performance, being rated first in its category in the 2009 and 2010 MIREX evaluations. The second task to which we direct our attention is the machine learning of logical descriptions of harmonic sequences in order to characterise particular styles or genres. For this work we use inductive logic programming to obtain representations such as decision trees which can be used to classify unseen examples or provide insight into the characteristics of a data corpus.

Computational models of harmony are important for many application areas of music informatics, as well as for music psychology and musicology itself. For example, a harmony model is a necessary component of intelligent music notation software, for determining the correct key signature and pitch spelling of accidentals where music is obtained from digital keyboards or MIDI files. Likewise processes such as automatic transcription are benefitted by tracking the harmonic context at each point in the music [24]. It has been shown that harmonic modelling improves search and retrieval in music databases, for example in order to find variations of an example query [36], which is useful for musicological research. Theories of music cognition, if expressed unambiguously, can be implemented and tested on large data corpora and compared with human annotations, in order to verify or refine concepts in the theory.

[1] Progress is evident for example in the annual MIREX series of evaluations of music information retrieval systems (`http://www.music-ir.org/mirex/wiki/2010:Main_Page`)

The remainder of the paper is structured as follows. The next section provides an overview of research in harmony modelling. This is followed by a section describing our probabilistic model of chord transcription. In section 4, we present our logic-based approach to modelling of harmony, and show how this can be used to characterise and classify music. The final section is a brief conclusion and outline of future work.

2 Background

Research into computational analysis of harmony has a history of over four decades since [44] proposed a grammar-based analysis that required the user to manually remove any non-harmonic notes (e.g. passing notes, suspensions and ornaments) before the algorithm processed the remaining chord sequence. A grammar-based approach was also taken by [40], who developed a set of chord substitution rules, in the form of a context-free grammar, for generating 12-bar Blues sequences. [31] addressed the problem of extracting patterns and substitution rules automatically from jazz standard chord sequences, and discussed how the notions of expectation and surprise are related to the use of these patterns and rules.

Closely related to grammar-based approaches are rule-based approaches, which were used widely in early artificial intelligence systems. [21] used an elimination process combined with heuristic rules in order to infer the tonality given a fugue melody from Bach's Well-Tempered Clavier. [15] presents an expert system consisting of about 350 rules for generating 4-part harmonisations of melodies in the style of Bach Chorales. The rules cover the chord sequences, including cadences and modulations, as well as the melodic lines of individual parts, including voice leading. [28] developed an expert system with a complex set of rules for recognising consonances and dissonances in order to infer the chord sequence. Maxwell's approach was not able to infer harmony from a melodic sequence, as it considered the harmony at any point in time to be defined by a subset of the simultaneously sounding notes.

[41] addressed some of the weaknesses of earlier systems with a combined rhythmic and harmonic analysis system based on preference rules [20]. The system assigns a numerical score to each possible interpretation based on the preference rules which the interpretation satisfies, and searches the space of all solutions using dynamic programming restricted with a beam search. The system benefits from the implementation of rules relating harmony and metre, such as the preference rule which favours non-harmonic notes occurring on weak metrical positions. One claimed strength of the approach is the transparency of the preference rules, but this is offset by the opacity of the system parameters such as the numeric scores which are assigned to each rule.

[33] proposed a counting scheme for matching performed notes to chord templates for variable-length segments of music. The system is intentionally simplistic, in order that the framework might easily be extended or modified. The main contributions of the work are the graph search algorithms, inspired by Temperley's dynamic programming approach, which determine the segmentation to be

used in the analysis. The proposed graph search algorithm is shown to be much more efficient than standard algorithms without differing greatly in the quality of analyses it produces.

As an alternative to the rule-based approaches, which suffer from the cumulative effects of errors, [38] proposed a probabilistic approach to functional harmonic analysis, using a hidden Markov model. For each time unit (measure or half-measure), their system outputs the current key and the scale degree of the current chord. In order to make the computation tractable, a number of simplifying assumptions were made, such as the symmetry of all musical keys. Although this reduced the number of parameters by at least two orders of magnitude, the training algorithm was only successful on a subset of the parameters, and the remaining parameters were set by hand.

An alternative stream of research has been concerned with multidimensional representations of polyphonic music [10,11,42] based on the Viewpoints approach of [12]. This representation scheme is for example able to preserve information about voice leading which is otherwise lost by approaches that treat harmony as a sequence of chord symbols.

Although most research has focussed on analysing musical works, some work investigates the properties of entire corpora. [25] compared two corpora of chord sequences, belonging to jazz standards and popular (Beatles) songs respectively, and found key- and context-independent patterns of chords which occurred frequently in each corpus. [26] examined the statistics of the chord sequences of several thousand songs, and compared the results to those from a standard natural language corpus in an attempt to find lexical units in harmony that correspond to words in language. [34,35] investigated whether stochastic language models including naive Bayes classifiers and 2-, 3- and 4-grams could be used for automatic genre classification. The models were tested on both symbolic and audio data, where an off-the-shelf chord transcription algorithm was used to convert the audio data to a symbolic representation. [39] analysed the Beatles corpus using probabilistic N-grams in order to show that the dependency of a chord on its context extends beyond the immediately preceding chord (the first-order Markov assumption). [9] studied differences in the use of harmony across various periods of classical music history, using root progressions (i.e. the sequence of root notes of chords in a progression) reduced to 2 categories (dominant and subdominant) to give a representation called harmonic vectors. The use of root progressions is one of the representations we use in our own work in section 4 [2].

All of the above systems process symbolic input, such as that found in a score, although most of the systems do not require the level of detail provided by the score (e.g. key signature, pitch spelling), which they are able to reconstruct from the pitch and timing data. In recent years, the focus of research has shifted to the analysis of audio files, starting with the work of [16], who computed a chroma representation (salience of frequencies representing the 12 Western pitch classes, independent of octave) which was matched to a set of chord templates using the inner product. Alternatively, [7] modelled chords with a 12-dimensional Gaussian distribution, where chord notes had a mean of 1, non-chord notes had a mean of 0,

and the covariance matrix had high values between pairs of chord notes. A hidden Markov model was used to infer the most likely sequence of chords, where state transition probabilities were initialised based on the distance between chords on a special circle of fifths which included minor chords near to their relative major chord. Further work on audio-based harmony analysis is reviewed thoroughly in three recent doctoral theses, to which the interested reader is referred [22,18,32].

3 A Probabilistic Model for Chord Transcription

Music theory, perceptual studies, and musicians themselves generally agree that no musical quality can be treated individually. When a musician transcribes the chords of a piece of music, the chord labels are not assigned solely on the basis of local pitch content of the signal. Musical context such as the key, metrical position and even the large-scale structure of the music play an important role in the interpretation of harmony. [17, Chapter 4] conducted a survey among human music transcription experts, and found that they use several musical context elements to guide the transcription process: not only is a prior rough chord detection the basis for accurate note transcription, but the chord transcription itself depends on the tonal context and other parameters such as beats, instrumentation and structure.

The goal of our recent work on chord transcription [24,22,23] is to propose computational models that integrate musical context into the automatic chord estimation process. We employ a dynamic Bayesian network (DBN) to combine models of metrical position, key, chord, bass note and beat-synchronous bass and treble chroma into a single high-level musical context model. The most probable sequence of metrical positions, keys, chords and bass notes is estimated via Viterbi inference.

A DBN is a graphical model representing a succession of simple Bayesian networks in time. These are assumed to be Markovian and time-invariant, so the model can be expressed recursively in two time slices: the initial slice and the recursive slice. Our DBN is shown in Figure 1. Each node in the network represents a random variable, which might be an observed node (in our case the bass and treble chroma) or a hidden node (the key, metrical position, chord and bass pitch class nodes). Edges in the graph denote dependencies between variables. In our DBN the musically interesting behaviour is modelled in the recursive slice, which represents the progress of all variables from one beat to the next. In the following paragraphs we explain the function of each node.

Chord. Technically, the dependencies of the random variables are described in the conditional probability distribution of the dependent variable. Since the highest number of dependencies join at the chord variable, it takes a central position in the network. Its conditional probability distribution is also the most complex: it depends not only on the key and the metrical position, but also on the chord variable in the previous slice. The chord variable has 121 different chord states (see below), and its dependency on the previous chord variable enables

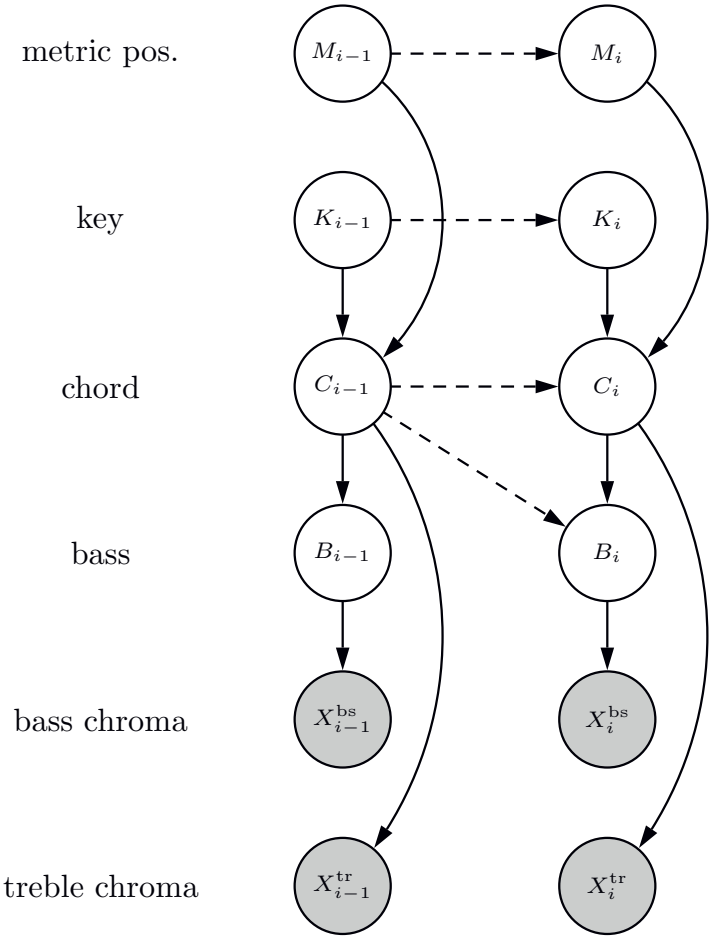
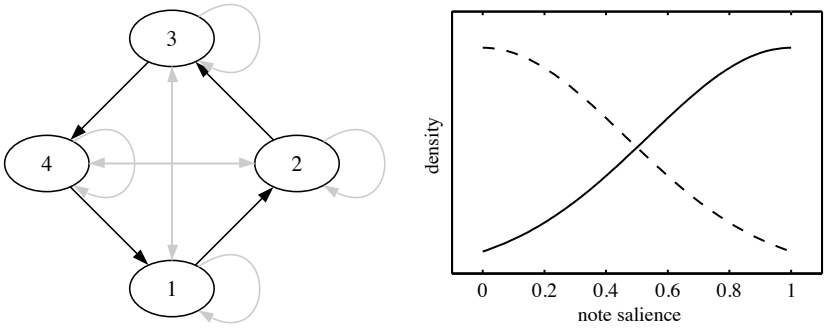


Fig. 1. Our network model topology, represented as a DBN with two slices and six layers. The clear nodes represent random variables, while the observed ones are shaded grey. The directed edges represent the dependency structure. Intra-slice dependency edges are drawn solid, inter-slice dependency edges are dashed.

the reinforcement of smooth sequences of these states. The probability distribution of chords conditional on the previous chord strongly favours the chord that was active in the previous slice, similar to a high self-transition probability in a hidden Markov model. While leading to a chord transcription that is stable over time, dependence on the previous chord alone is not sufficient to model adherence to the key. Instead, it is modelled conditionally on the key variable: the probability distribution depends on the chord's fit with the current key, based on an expert function motivated by Krumhansl's chord-key ratings [19, page 171]. Finally, the chord variable's dependency on the metrical position node allows us to favour chord changes at strong metrical positions to achieve a transcription that resembles more closely that of a human transcriber.



(a) metrical position model (b) model of a single chroma pitch class

Fig. 2

Key and metrical position. The dependency structure of the key and metrical position variables are comparatively simpler, since they depend only on the respective predecessor. The emphasis on smooth, stable key sequences is handled in the same way as it is in chords, but the 24 states representing major and minor keys have even higher self-transition probability, and hence they will persist for longer stretches of time. The metrical position model represents a $\frac{4}{4}$ meter and hence has four states. The conditional probability distribution strongly favours "normal" beat transitions, i.e. from one beat to the next, but it also allows for irregular transitions in order to accommodate temporary deviations from $\frac{4}{4}$ meter and occasional beat tracking errors. In Figure 2a black arrows represent a transition probability of $1-\varepsilon$ (where $\varepsilon = 0.05$) to the following beat. Grey arrows represent a probability of $\varepsilon/2$ to jump to different beats through self-transition or omission of the expected beat.

Bass. The random variable that models the bass has 13 states, one for each of the pitch classes, and one "no bass" state. It depends on both the current chord and the previous chord. The current chord is the basis of the most probable bass notes that can be chosen. The highest probability is assigned to the "nominal" chord bass pitch class[2], lower probabilities to the remaining chord pitch classes, and the rest of the probability mass is distributed between the remaining pitch classes. The additional use of the dependency on the previous chord allows us to model the behaviour of the bass note on the first beat of the chord differently from its behaviour on later beats. We can thus model the tendency for the played bass note to coincide with the "nominal" bass note of the chord (e.g. the note B in the `B7` chord), while there is more variation in the bass notes played during the rest of the duration of the chord.

Chroma. The chroma nodes provide models of the bass and treble chroma audio features. Unlike the discrete nodes previously discussed, they are continuous because the 12 elements of the chroma vector represent relative salience, which

[2] The chord symbol itself always implies a bass note, but the bass line might include other notes not specified by the chord symbol, as in the case of walking bass.

can assume any value between zero and unity. We represent both bass and treble chroma as multidimensional Gaussian random variables. The bass chroma variable has 13 different Gaussians, one for every bass state, and the treble chroma node has 121 Gaussians, one for every chord state. The means of the Gaussians are set to reflect the nature of the chords: to unity for pitch classes that are part of the chord, and to zero for the rest. A single variate in the 12-dimensional Gaussian treble chroma distribution models one pitch class, as illustrated in Figure 2b. Since the chroma values are normalised to the unit interval, the Gaussian model functions similar to a regression model: for a given chord the Gaussian density increases with increasing salience of the chord notes (solid line), and decreases with increasing salience of non-chord notes (dashed line). For more details see [22].

One important aspect of the model is the wide variety of chords it uses. It models ten different chord types (`maj`, `min`, `maj/3`, `maj/5`, `maj6`, `7`, `maj7`, `min7`, `dim`, `aug`) and the "no chord" class `N`. The chord labels with slashes denote chords whose bass note differs from the chord root, for example `D/3` represents a D major chord in first inversion (sometimes written `D/F`♯). The recognition of these chords is a novel feature of our chord recognition algorithm. Figure 3 shows a score rendered using exclusively the information in our model. In the last four bars, marked with a box, the second chord is correctly annotated as `D/F`♯. The position of the bar lines is obtained from the metrical position variable, the key signature from the key variable, and the bass notes from the bass variable. The chord labels are obtained from the chord variable, replicated as notes in the treble staff for better visualisation. The crotchet rest on the first beat of the piece indicates that here, the Viterbi algorithm inferred that the "no chord" model fits best.

Using a standard test set of 210 songs used in the MIREX chord detection task, our basic model achieved an accuracy of 73%, with each component of the model contributing significantly to the result. This improves on the best result at

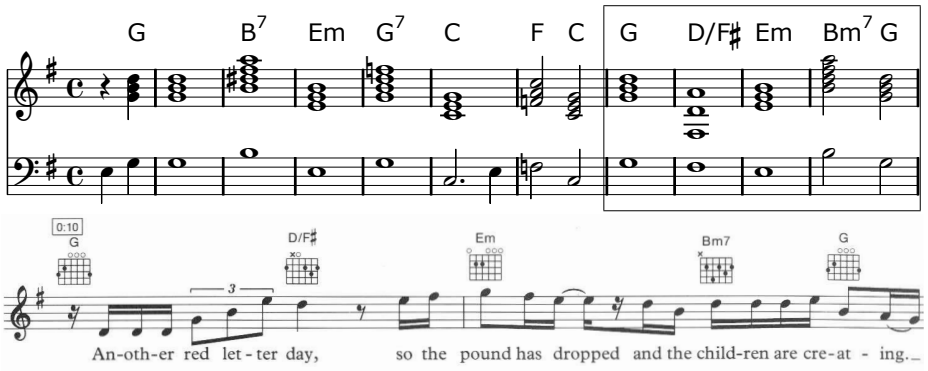


Fig. 3. Excerpt of automatic output of our algorithm (top) and song book version (bottom) of the pop song "Friends Will Be Friends" (Deacon/Mercury). The song book excerpt corresponds to the four bars marked with a box.

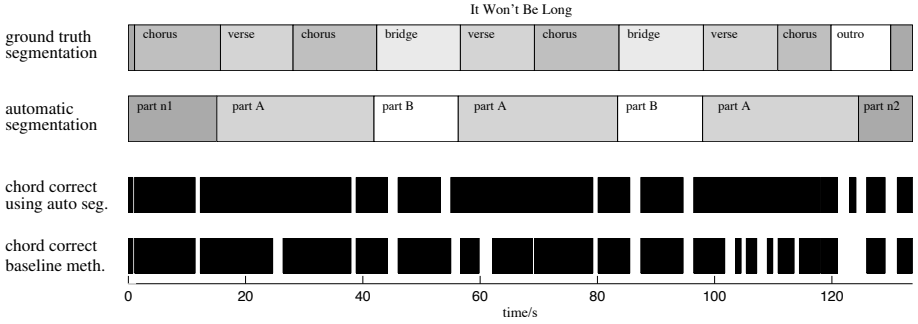


Fig. 4. Segmentation and its effect on chord transcription for the Beatles' song "It Won't Be Long" (Lennon/McCartney). The top 2 rows show the human and automatic segmentation respectively. Although the structure is different, the main repetitions are correctly identified. The bottom 2 rows show (in black) where the chord was transcribed correctly by our algorithm using (respectively not using) the segmentation information.

MIREX 2009 for pre-trained systems. Further improvements have been made via two extensions of this model: taking advantage of repeated structural segments (e.g. verses or choruses), and refining the front-end audio processing.

Most musical pieces have segments which occur more than once in the piece, and there are two reasons for wishing to identify these repetitions. First, multiple sets of data provide us with extra information which can be shared between the repeated segments to improve detection performance. Second, in the interest of consistency, we can ensure that the repeated sections are labelled with the same set of chord symbols. We developed an algorithm that automatically extracts the repetition structure from a beat-synchronous chroma representation [27], which ranked first in the 2009 MIREX Structural Segmentation task.

After building a similarity matrix based on the correlation between beat-synchronous chroma vectors, the method finds sets of repetitions whose elements have the same length in beats. A repetition set composed of n elements with length d receives a score of $(n-1)d$, reflecting how much space a hypothetical music editor could save by typesetting a repeated segment only once. The repetition set with the maximum score ("part A" in Figure 4) is added to the final list of structural elements, and the process is repeated on the remainder of the song until no valid repetition sets are left.

The resulting structural segmentation is then used to merge the chroma representations of matching segments. Despite the inevitable errors propagated from incorrect segmentation, we found a significant performance increase (to 75% on the MIREX score) by using the segmentation. In Figure 4 the beneficial effect of using the structural segmentation can clearly be observed: many of the white stripes representing chord recognition errors are eliminated by the structural segmentation method, compared to the baseline method.

A further improvement was achieved by modifying the front end audio processing. We found that by learning chord profiles as Gaussian mixtures, the recognition rate of some chords can be improved. However this did not result in an overall improvement, as the performance on the most common chords decreased. Instead, an approximate pitch transcription method using non-negative least squares was employed to reduce the effect of upper harmonics in the chroma representations [23]. This results in both a qualitative (reduction of specific errors) and quantitative (a substantial overall increase in accuracy) improvement in results, with a MIREX score of 79% (without using segmentation), which again is significantly better than the state of the art. By combining both of the above enhancements we reach an accuracy of 81%, a statistically significant improvement over the best result (74%) in the 2009 MIREX Chord Detection tasks and over our own previously mentioned results.

4 Logic-Based Modelling of Harmony

First order logic (FOL) is a natural formalism for representing harmony, as it is sufficiently general for describing combinations and sequences of notes of arbitrary complexity, and there are well-studied approaches for performing inference, pattern matching and pattern discovery using subsets of FOL. A further advantage of logic-based representations is that a system's output can be presented in an intuitive way to non-expert users. For example, a decision tree generated by our learning approach provides much more intuition about what was learnt than would a matrix of state transition probabilities. In this work we focus in particular on inductive logic programming (ILP), which is a machine learning approach using logic programming (a subset of FOL) to uniformly represent examples, background knowledge and hypotheses. An ILP system takes as input a set of positive and negative examples of a concept, plus some background knowledge, and outputs a logic program which "explains" the concept, in the sense that all of the positive examples but (ideally) none of the negative examples can be derived from the logic program and background knowledge.

ILP has been used for various musical tasks, including inference of harmony [37] and counterpoint [30] rules from musical examples, as well as rules for expressive performance [43]. In our work, we use ILP to learn sequences of chords that might be characteristic of a musical style [2], and test the models on classification tasks [3,4,1]. In each case we represent the harmony of a piece of music by a list of chords, and learn models which characterise the various classes of training data in terms of features derived from subsequences of these chord lists.

4.1 Style Characterisation

In our first experiments [2], we analysed two chord corpora, consisting of the Beatles studio albums (180 songs, 14132 chords) and a set of jazz standards from the Real Book (244 songs, 24409 chords) to find harmonic patterns that differentiate the two corpora. Chord sequences were represented in terms of the interval between successive root notes or successive bass notes (to make the

sequences key-independent), plus the category of each chord (reduced to a triad except in the case of the dominant seventh chord). For the Beatles data, where the key had been annotated for each piece, we were also able to express the chord symbols in terms of the scale degree relative to the key, rather than its pitch class, giving a more musically satisfying representation. Chord sequences of length 4 were used, which we had previously found [25] to be a good compromise of sufficient length to capture the context (and thus the function) of the chords, without the sequences being overspecific, in which case few or no patterns would be found.

Two models were built, one using the Beatles corpus as positive examples and the other using the Real Book corpus as positive examples. The ILP system Aleph was employed, which finds a minimal set of rules which cover (i.e. describe) all positive examples (and a minimum number of negative examples). The models built by Aleph consisted of 250 rules for the Beatles corpus and 596 rules for the Real Book. Note that these rules cover every 4-chord sequence in every song, so it is only the rules which cover many examples that are relevant in terms of characterising the corpus. Also, once a sequence has been covered, it is not considered again by the system, so the output is dependent on the order of presentation of the examples.

We briefly discuss some examples of rules with the highest coverage. For the Beatles corpus, the highest coverage (35%) was the 4-chord sequence of major triads (regardless of roots). Other highly-ranked patterns of chord categories (5% coverage) had 3 major triads and one minor triad in the sequence. This is not surprising, in that popular music generally has a less rich harmonic vocabulary than jazz. Patterns of root intervals were also found, including a [perfect 4th, perfect 5th, perfect 4th] pattern (4%), which could for example be interpreted as a `I - IV - I - IV` progression or as `V - I - V - I`. Since the root interval does not encode the key, it is not possible to distinguish between these interpretations (and it is likely that the data contains instances of both). At 2% coverage, the interval sequence [perfect 4th, major 2nd, perfect 4th] (e.g. `I - IV - V - I`) is another well-known chord sequence.

No single rule covered as many Real Book sequences as the top rule for the Beatles, but some typical jazz patterns were found, such as [perfect 4th, perfect 4th, perfect 4th] (e.g. `ii - V - I - IV`, coverage 8%), a cycle of descending fifths, and [major 6th, perfect 4th, perfect 4th] (e.g. `I - vi - ii - V`, coverage 3%), a typical turnaround pattern.

One weakness with this first experiment, in terms of its goal as a pattern discovery method, is that the concept to learn and the vocabulary to describe it (defined in the background knowledge) need to be given in advance. Different vocabularies result in different concept descriptions, and a typical process of concept characterisation is interactive, involving several refinements of the vocabulary in order to obtain an interesting theory. Thus, as we refine the vocabulary we inevitably reduce the problem to a pattern matching task rather than pattern discovery. A second issue is that since musical styles have no formal definition, it is not possible to quantify the success of style characterisation

directly, but only indirectly, by using the learnt models to classify unseen examples. Thus the following harmony modelling experiments are evaluated via the task of genre classification.

4.2 Genre Classification

For the subsequent experiments we extended the representation to allow variable length patterns and used TILDE, a first-order logic decision tree induction algorithm for modelling harmony [3,4]. As test data we used a collection of 856 pieces (120510 chords) covering 3 genres, each of which was divided into a further 3 subgenres: academic music (Baroque, Classical, Romantic), popular music (Pop, Blues, Celtic) and jazz (Pre-bop, Bop, Bossa Nova). The data is represented in the Band in a Box format, containing a symbolic encoding of the chords, which were extracted and encoded using a definite clause grammar (DCG) formalism. The software Band in a Box is designed to produce an accompaniment based on the chord symbols, using a MIDI synthesiser. In further experiments we tested the classification method using automatic chord transcription (see section 3) from the synthesised audio data, in order to test the robustness of the system to errors in the chord symbols.

The DCG representation was developed for natural language processing to express syntax or grammar rules in a format which is both human-readable and machine-executable. Each predicate has two arguments (possibly among other arguments), an input list and an output list, where the output list is always a suffix of the input list. The difference between the two lists corresponds to the subsequence described by the predicate. For example, the predicate `gap(In,Out)` states that the input list of chords (`In`) commences with a subsequence corresponding to a "gap", and the remainder of the input list is equal to the output list (`Out`). In our representation, a gap is an arbitrary sequence of chords, which allows the representation to skip any number of chords at the beginning of the input list without matching them to any harmony concept. Extra arguments can encode parameters and/or context, so that the term `degreeAndCategory(Deg,Cat,In,Out,Key)` states that the list `In` begins with a chord of scale degree `Deg` and chord category `Cat` in the context of the key `Key`. Thus the sequence:

```
gap(S,T),
degreeAndCategory(2,min7,T,U,gMajor),
degreeAndCategory(5,7,U,V,gMajor),
degreeAndCategory(1,maj7,V,[],gMajor)
```

states that the list `S` starts with any chord subsequence (gap), followed by a minor 7th chord on the 2nd degree of G major (i.e. Amin7), followed by a (dominant) 7th chord on the 5th degree (D7) and ending with a major 7th chord on the tonic (Gmaj7).

TILDE learns a classification model based on a vocabulary of predicates supplied by the user. In our case, we described the chords in terms of their root note,

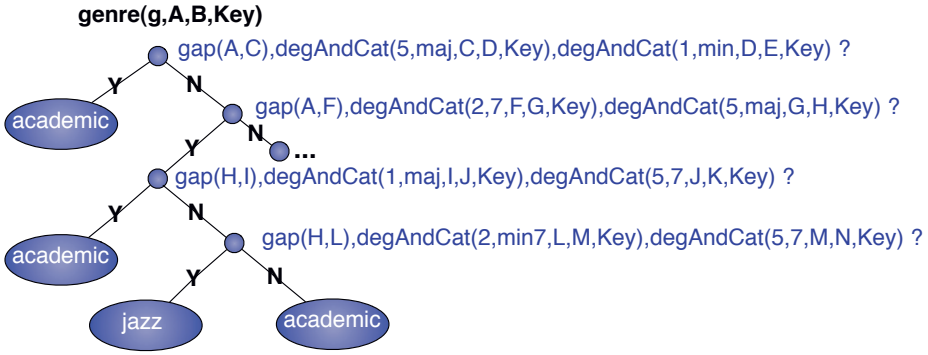


Fig. 5. Part of the decision tree for a binary classifier for the classes Jazz and Academic

Table 1. Results compared with the baseline for 2-class, 3-class and 9-class classification tasks

Classification Task	Baseline	Symbolic	Audio
Academic – Jazz	0.55	0.947	0.912
Academic – Popular	0.55	0.826	0.728
Jazz – Popular	0.61	0.891	0.807
Academic – Popular – Jazz	0.40	0.805	0.696
All 9 subgenres	0.21	0.525	0.415

scale degree, chord category, and intervals between successive root notes, and we constrained the learning algorithm to generate rules containing subsequences of length at least two chords. The model can be expressed as a decision tree, as shown in figure 5, where the choice of branch taken is based on whether or not the chord sequence matches the predicates at the current node, and the class to which the sequence belongs is given by the leaf of the decision tree reached by following these choices. The decision tree is equivalent to an ordered set of rules or a Prolog program. Note that a rule at a single node of a tree cannot necessarily be understood outside of its context in the tree. In particular, a rule by itself cannot be used as a classifier.

The results for various classification tasks are shown in Table 1. All results are significantly above the baseline, but performance clearly decreases for more difficult tasks. Perfect classification is not to be expected from harmony data, since other aspects of music such as instrumentation (timbre), rhythm and melody are also involved in defining and recognising musical styles.

Analysis of the most common rules extracted from the decision tree models built during these experiments reveals some interesting and well-known jazz, academic and popular music harmony patterns. For each rule shown below, the coverage expresses the fraction of songs in each class that match the rule. For example, while a perfect cadence is common to both academic and jazz styles, the chord categories distinguish the styles very well, with academic music using triads and jazz using seventh chords:

genre(**academic**,A,B,Key) :- **gap(A,C),**
degreeAndCategory(5,maj,C,D,Key),
degreeAndCategory(1,maj,D,E,Key),
gap(E,B).

[Coverage: academic=133/235; jazz=10/338]

genre(**jazz**,A,B,Key) :- **gap(A,C),**
degreeAndCategory(5,7,C,D,Key),
degreeAndCategory(1,maj7,D,E,Key),
gap(E,B).

[Coverage: jazz=146/338; academic=0/235]

A good indicator of blues is the sequence: `... - I7 - IV7 - ...`

genre(**blues**,A,B,Key) :- **gap(A,C),**
degreeAndCategory(1,7,C,D,Key),
degreeAndCategory(4,7,D,E,Key),
gap(E,B).

[Coverage: blues=42/84; celtic=0/99; pop=2/100]

On the other hand, jazz is characterised (but not exclusively) by the sequence: `... - ii7 - V7 - ...`

genre(**jazz**,A,B,Key) :- **gap(A,C),**
degreeAndCategory(2,min7,C,D,Key),
degreeAndCategory(5,7,D,E,Key),
gap(E,B).

[Coverage: jazz=273/338; academic=42/235; popular=52/283]

The representation also allows for longer rules to be expressed, such as the following rule describing a modulation to the dominant key and back again in academic music: `... - II7 - V - ... - I - V7 - ...`

genre(**academic**,A,B,Key) :- **gap(A,C),**
degreeAndCategory(2,7,C,D,Key),
degreeAndCategory(5,maj,D,E,Key),
gap(E,F),
degreeAndCategory(1,maj,F,G,Key),
degreeAndCategory(5,7,G,H,Key),
gap(H,B).

[Coverage: academic=75/235; jazz=0/338; popular=1/283]

Although none of the rules are particularly surprising, these examples illustrate some meaningful musicological concepts that are captured by the rules. In general, we observed that Academic music is characterised by rules establishing the tonality, e.g. via cadences, while Jazz is less about tonality, and more about harmonic colour, e.g. the use of 7th, 6th, augmented and more complex chords, and Popular music harmony tends to have simpler harmonic rules as melody is predominant in this style. The system is also able to find longer rules that a human might not spot easily. Working from audio data, even though the transcriptions are not fully accurate, the classification and rules still capture the same general trends as for symbolic data.

For genre classification we are not advocating a harmony-based approach alone. It is clear that other musical features are better predictors of genre. Nevertheless, the positive results encouraged a further experiment in which we integrated the current classification approach with a state-of-the-art genre classification system, to test whether the addition of a harmony feature could improve its performance.

4.3 Genre Classification Using Harmony and Low-Level Features

In recent work [1] we developed a genre classification framework combining both low-level signal-based features and high-level harmony features. A state-of-the-art statistical genre classifier [8] using 206 features, covering spectral, temporal, energy, and pitch characteristics of the audio signal, was extended using a random forest classifier containing rules for each genre (classical, jazz and pop) derived from chord sequences. We extended our previous work using the first-order logic induction algorithm TILDE, to learn a random forest instead of a single decision tree from the chord sequence corpus described in the previous genre classification experiments. The random forest model achieved better classification rates (88% on the symbolic data and 76% on the audio data) for the three-class classification problem (previous results 81% and 70% respectively).

Having trained the harmony classifier, its output was added as an extra feature to the low-level classifier and the combined classifier was tested on three-genre subsets of two standard genre classification data sets (GTZAN and ISMIR04) containing 300 and 448 recordings respectively. Multilayer perceptrons and support vector machines were employed to classify the test data using 5×5-fold cross-validation and feature selection. Results are shown in table 2 for the support vector machine classifier, which outperformed the multilayer perceptrons.

Results indicate that the combination of low-level features with the harmony-based classifier produces improved genre classification results despite the fact

Table 2. Best mean classification results (and number of features used) for the two data sets using 5×5-fold cross-validation and feature selection

Classifier	GTZAN data set	ISMIR04 data set
SVM without harmony feature	0.887 (60 features)	0.938 (70 features)
SVM with harmony feature	0.911 (50 features)	0.953 (80 features)

that the classification rate of the harmony-based classifier alone is poor. For both datasets the improvements over the standard classifier (as shown in table 2) were found to be statistically significant.

5 Conclusion

We have looked at two approaches to the modelling of harmony which aim to "dig deeper into the music". In our probabilistic approach to chord transcription, we demonstrated the advantage of modelling musical context such as key, metrical structure and bass line, and simultaneously estimating all of these variables along with the chord. We also developed an audio feature using non-negative least squares that reflects the notes played better than the standard chroma feature, and therefore reduces interference from harmonically irrelevant partials and noise. A further improvement of the system was obtained by modelling the global structure of the music, identifying repeated sections and averaging features over these segments. One promising avenue of further work is the separation of the audio (low-level) and symbolic (high-level) models which are conceptually distinct but modelled together in current systems. A low-level model would be concerned only with the production or analysis of audio — the mapping from notes to features; while a high-level model would be a musical model handling the mapping from chord symbols to notes.

Using a logic-based approach, we showed that it is possible to automatically discover patterns in chord sequences which characterise a corpus of data, and to use such models as classifiers. The advantage with a logic-based approach is that models learnt by the system are transparent: the decision tree models can be presented to users as sets of human readable rules. This explanatory power is particularly relevant for applications such as music recommendation. The DCG representation allows chord sequences of any length to coexist in the same model, as well as context information such as key. Our experiments found that the more musically meaningful Degree-and-Category representation gave better classification results than using root intervals. The results using transcription from audio data were encouraging in that although some information was lost in the transcription process, the classification results remained well above the baseline, and thus this approach is still viable when symbolic representations of the music are not available. Finally, we showed that the combination of high-level harmony features with low-level features can lead to genre classification accuracy improvements in a state-of-the-art system, and believe that such high-level models provide a promising direction for genre classification research.

While these methods have advanced the state of the art in music informatics, it is clear that in several respects they are not yet close to an expert musician's understanding of harmony. Limiting the representation of harmony to a list of chord symbols is inadequate for many applications. Such a representation may be sufficient as a memory aid for jazz and pop musicians, but it allows only a very limited specification of chord voicing (via the bass note), and does not permit analysis of polyphonic texture such as voice leading, an important concept in many harmonic styles, unlike the recent work of [11] and [29]. Finally, we note

that the current work provides little insight into harmonic function, for example the ability to distinguish harmony notes from ornamental and passing notes and to recognise chord substitutions, both of which are essential characteristics of a system that models a musician's understanding of harmony. We hope to address these issues in future work.

Acknowledgements. This work was performed under the OMRAS2 project, supported by the Engineering and Physical Sciences Research Council, grant EP/E017614/1. We would like to thank Chris Harte, Matthew Davies and others at C4DM who contributed to the annotation of the audio data, and the Pattern Recognition and Artificial Intelligence Group at the University of Alicante, who provided the Band in a Box data.

References

1. Anglade, A., Benetos, E., Mauch, M., Dixon, S.: Improving music genre classification using automatically induced harmony rules. Journal of New Music Research 39(4), 349–361 (2010)
2. Anglade, A., Dixon, S.: Characterisation of harmony with inductive logic programming. In: 9th International Conference on Music Information Retrieval, pp. 63–68 (2008)
3. Anglade, A., Ramirez, R., Dixon, S.: First-order logic classification models of musical genres based on harmony. In: 6th Sound and Music Computing Conference, pp. 309–314 (2009)
4. Anglade, A., Ramirez, R., Dixon, S.: Genre classification using harmony rules induced from automatic chord transcriptions. In: 10th International Society for Music Information Retrieval Conference, pp. 669–674 (2009)
5. Aucouturier, J.J., Defréville, B., Pachet, F.: The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. Journal of the Acoustical Society of America 122(2), 881–891 (2007)
6. Aucouturier, J.J., Pachet, F.: Improving timbre similarity: How high is the sky? Journal of Negative Results in Speech and Audio Sciences 1(1) (2004)
7. Bello, J.P., Pickens, J.: A robust mid-level representation for harmonic content in music signals. In: 6th International Conference on Music Information Retrieval, pp. 304–311 (2005)
8. Benetos, E., Kotropoulos, C.: Non-negative tensor factorization applied to music genre classification. IEEE Transactions on Audio, Speech, and Language Processing 18(8), 1955–1967 (2010)
9. Cathé, P.: Harmonic vectors and stylistic analysis: A computer-aided analysis of the first movement of Brahms' String Quartet Op. 51-1. Journal of Mathematics and Music 4(2), 107–119 (2010)
10. Conklin, D.: Representation and discovery of vertical patterns in music. In: Anagnostopoulou, C., Ferrand, M., Smaill, A. (eds.) ICMAI 2002. LNCS (LNAI), vol. 2445, pp. 32–42. Springer, Heidelberg (2002)
11. Conklin, D., Bergeron, M.: Discovery of contrapuntal patterns. In: 11th International Society for Music Information Retrieval Conference, pp. 201–206 (2010)
12. Conklin, D., Witten, I.: Multiple viewpoint systems for music prediction. Journal of New Music Research 24(1), 51–73 (1995)

13. Dixon, S., Pampalk, E., Widmer, G.: Classification of dance music by periodicity patterns. In: 4th International Conference on Music Information Retrieval, pp. 159–165 (2003)
14. Downie, J., Byrd, D., Crawford, T.: Ten years of ISMIR: Reflections on challenges and opportunities. In: 10th International Society for Music Information Retrieval Conference, pp. 13–18 (2009)
15. Ebcioğlu, K.: An expert system for harmonizing chorales in the style of J. S. Bach. In: Balaban, M., Ebcioiğlu, K., Laske, O. (eds.) Understanding Music with AI, pp. 294–333. MIT Press, Cambridge (1992)
16. Fujishima, T.: Realtime chord recognition of musical sound: A system using Common Lisp Music. In: Proceedings of the International Computer Music Conference, pp. 464–467 (1999)
17. Hainsworth, S.W.: Techniques for the Automated Analysis of Musical Audio. Ph.D. thesis, University of Cambridge, Cambridge, UK (2003)
18. Harte, C.: Towards Automatic Extraction of Harmony Information from Music Signals. Ph.D. thesis, Queen Mary University of London, Centre for Digital Music (2010)
19. Krumhansl, C.L.: Cognitive Foundations of Musical Pitch. Oxford University Press, Oxford (1990)
20. Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge (1983)
21. Longuet-Higgins, H., Steedman, M.: On interpreting Bach. Machine Intelligence 6, 221–241 (1971)
22. Mauch, M.: Automatic Chord Transcription from Audio Using Computational Models of Musical Context. Ph.D. thesis, Queen Mary University of London, Centre for Digital Music (2010)
23. Mauch, M., Dixon, S.: Approximate note transcription for the improved identification of difficult chords. In: 11th International Society for Music Information Retrieval Conference, pp. 135–140 (2010)
24. Mauch, M., Dixon, S.: Simultaneous estimation of chords and musical context from audio. IEEE Transactions on Audio, Speech and Language Processing 18(6), 1280–1289 (2010)
25. Mauch, M., Dixon, S., Harte, C., Casey, M., Fields, B.: Discovering chord idioms through Beatles and Real Book songs. In: 8th International Conference on Music Information Retrieval, pp. 111–114 (2007)
26. Mauch, M., Müllensiefen, D., Dixon, S., Wiggins, G.: Can statistical language models be used for the analysis of harmonic progressions? In: International Conference on Music Perception and Cognition (2008)
27. Mauch, M., Noland, K., Dixon, S.: Using musical structure to enhance automatic chord transcription. In: 10th International Society for Music Information Retrieval Conference, pp. 231–236 (2009)
28. Maxwell, H.: An expert system for harmonizing analysis of tonal music. In: Balaban, M., Ebcioiğlu, K., Laske, O. (eds.) Understanding Music with AI, pp. 334–353. MIT Press, Cambridge (1992)
29. Mearns, L., Tidhar, D., Dixon, S.: Characterisation of composer style using high-level musical features. In: 3rd ACM Workshop on Machine Learning and Music (2010)
30. Morales, E.: PAL: A pattern-based first-order inductive system. Machine Learning 26(2-3), 227–252 (1997)
31. Pachet, F.: Surprising harmonies. International Journal of Computing Anticipatory Systems 4 (February 1999)

32. Papadopoulos, H.: Joint Estimation of Musical Content Information from an Audio Signal. Ph.D. thesis, Université Pierre et Marie Curie – Paris 6 (2010)
33. Pardo, B., Birmingham, W.: Algorithms for chordal analysis. Computer Music Journal 26(2), 27–49 (2002)
34. Pérez-Sancho, C., Rizo, D., Iñesta, J.M.: Genre classification using chords and stochastic language models. Connection Science 21(2-3), 145–159 (2009)
35. Pérez-Sancho, C., Rizo, D., Iñesta, J.M., de León, P.J.P., Kersten, S., Ramirez, R.: Genre classification of music by tonal harmony. Intelligent Data Analysis 14, 533–545 (2010)
36. Pickens, J., Bello, J., Monti, G., Sandler, M., Crawford, T., Dovey, M., Byrd, D.: Polyphonic score retrieval using polyphonic audio queries: A harmonic modelling approach. Journal of New Music Research 32(2), 223–236 (2003)
37. Ramirez, R.: Inducing musical rules with ILP. In: Proceedings of the International Conference on Logic Programming, pp. 502–504 (2003)
38. Raphael, C., Stoddard, J.: Functional harmonic analysis using probabilistic models. Computer Music Journal 28(3), 45–52 (2004)
39. Scholz, R., Vincent, E., Bimbot, F.: Robust modeling of musical chord sequences using probabilistic N-grams. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 53–56 (2009)
40. Steedman, M.: A generative grammar for jazz chord sequences. Music Perception 2(1), 52–77 (1984)
41. Temperley, D., Sleator, D.: Modeling meter and harmony: A preference rule approach. Computer Music Journal 23(1), 10–27 (1999)
42. Whorley, R., Wiggins, G., Rhodes, C., Pearce, M.: Development of techniques for the computational modelling of harmony. In: First International Conference on Computational Creativity, pp. 11–15 (2010)
43. Widmer, G.: Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. Artificial Intelligence 146(2), 129–148 (2003)
44. Winograd, T.: Linguistics and the computer analysis of tonal harmony. Journal of Music Theory 12(1), 2–49 (1968)