GENRE CLASSIFICATION USING HARMONY RULES INDUCED FROM AUTOMATIC CHORD TRANSCRIPTIONS

Amélie Anglade
Queen Mary
University of London
Centre for Digital Music
amelie.anglade@elec.qmul.ac.uk

Rafael Ramirez
Universitat Pompeu Fabra
Music Technology Group
rramirez@iua.upf.edu

Simon Dixon
Queen Mary
University of London
Centre for Digital Music
simon.dixon@elec.qmul.ac.uk

ABSTRACT

We present an automatic genre classification technique making use of frequent chord sequences that can be applied on symbolic as well as audio data. We adopt a first-order logic representation of harmony and musical genres: pieces of music are represented as lists of chords and musical genres are seen as context-free definite clause grammars using subsequences of these chord lists. To induce the context-free definite clause grammars characterising the genres we use a first-order logic decision tree induction algorithm. We report on the adaptation of this classification framework to audio data using an automatic chord transcription algorithm. We also introduce a high-level harmony representation scheme which describes the chords in term of both their degrees and chord categories. When compared to another high-level harmony representation scheme used in a previous study, it obtains better classification accuracies and shorter run times. We test this framework on 856 audio files synthesized from Band in a Box files and covering 3 main genres, and 9 subgenres. We perform 3-way and 2-way classification tasks on these audio files and obtain good classification results: between 67% and 79% accuracy for the 2-way classification tasks and between 58% and 72% accuracy for the 3-way classification tasks.

1. INTRODUCTION

To deal with the ever-increasing amount of digital music data in both personal and commercial musical libraries some automatic classification techniques are generally needed. Although metadata such as ID3 tags are often used to sort such collections, the MIR community has also shown a great interest in incorporating information extracted from the audio signal into the automatic classification process. While low-level representations of harmonic content have been used in several genre classification algorithms (e.g. chroma feature representation in [1]), little attention has been paid to how harmony in its temporal dimension, i.e. chord sequences, can help in this task. However, there seems to be a strong connection between musical genre and the use of different chord progressions [2]. For instance, it is well known that pop-rock tunes mainly follow the classical tonic-subdominant-dominant chord sequence, whereas jazz harmony books propose different series of chord progressions as a standard. We intend to test the extent to which harmonic progressions can be used for genre classification.

In a previous article [3] we have shown that efficient and transparent genre classification models entirely based on a high-level representation of harmony can be built using first-order logic. Music pieces were represented as lists of chords (obtained from symbolic files) and musical genres were seen as context-free definite-clause grammar using subsequences of any length of these chord lists. The grammar representing the genres were built using a first-order logic decision tree induction algorithm. These resulting models not only obtained good classification results when tested on symbolic data (between 72% and 86% accuracy on 2-class problems) but also provided a transparent explanation of the classification to the user. Indeed thanks to the expressiveness of first-order logic the decision trees obtained with this technique can be presented to the user as sets of human readable rules.

In this paper we extend our harmony-based approach to automatic genre classification by introducing a richer harmony representation and present the results of audio data classification. In our previous article we used the intervals between the root notes of consecutive chords. Root interval progressions capture some degree information and do not depend on the tonality. Thus when using root intervals no key extraction is necessary. However, one root interval progression can cover several degree sequences. For instance the degree sequences “IV-I-IV” and “I-V-I” are both represented by the root interval sequence “perfect fifth-perfect fourth”. To avoid such generalisations we introduce here another representation of harmony based on the degrees (i.e. I, V, etc.) and chord categories (i.e. min, 7, maj7, etc.). In addition such a representation matches the western representation of harmony and thus our classification models (i.e. decision trees or sets of classification rules describing the harmony) can be more easily interpreted by the users. Finally since degrees are relative to the key, a key estimation step is now needed. This is a requirement but not a limitation as nowadays many chord transcription algorithms from audio (e.g. [4,5]) do also perform key estimation.

The paper is organised as follows: In Section 2 we review some existing studies using high-level representation of harmony for automatic genre classification. In Section 3 we present the details of our methodology, including the knowledge representation and the learning algorithm employed in this study. In Section 4 we present the classification results of our first-order logic classification technique before concluding in Section 5.

2. RELATED WORK

Only a few studies have considered using higher level harmonic structures, such as chord progressions, for automatic genre recognition.

In [6], a rule-based system is used to classify sequences of chords belonging to three categories: Enya, Beatles and Chinese folk songs. A vocabulary of 60 different chords was used, including triads and seventh chords. Classification accuracy ranged from 70% to 84% using two-way classification, and the best results were obtained when trying to distinguish Chinese folk music from the other two styles, which is a reasonable result as both western styles should be closer in terms of harmony.

Paiement et al. [7] also used chord progressions to build probabilistic models. In that work, a set of 52 jazz standards was encoded as sequences of 4-note chords. The authors compared the generalization capabilities of a probabilistic tree model against a Hidden Markov Model (HMM), both capturing stochastic properties of harmony in jazz, and the results suggested that chord structures are a suitable source of information to represent musical genres.

More recently, Lee [8] has proposed genre-specific HMMs that learn chord progression characteristics for each genre. Although the ultimate goal of this work is using the genre models to improve the chord recognition rate, he also presented some results on the genre classification task. For that task a reduced set of chords (major, minor, and diminished) was used.

Finally, Perez-Sancho et al. [9] have investigated if 2, 3 and 4-grams of chords can be used for automatic genre classification on both symbolic and audio data. They report better classification results when using a richer vocabulary (seventh chords) and longer n-grams.

3. METHODOLOGY

Contrary to n-grams that are limited to sequences of length n the first-order logic representation scheme that we adopt can employ chord sequences of variable length to characterise a musical genre. A musical piece is represented as a list of chords. Each musical genre is illustrated by a series of musical pieces. The objective is to find interesting patterns, i.e. chord sequences, that appear in many songs of one genre and do not (frequently) appear in the other genres and use such sets of patterns to classify unknown musical pieces into genres. As there can be several independent patterns and each of them can be of any length we use a context-free definite-clause grammar formalism. Finally to induce such grammars we use TILDE [10], a first-order logic decision tree induction algorithm.

3.1 Knowledge representation

In the definite clause grammar (DCG) formalism a sequence over a finite alphabet of letters is represented as a list of letters. Here the chords (e.g. G7, Db, BM7, F#m7, etc.) are the letters of our alphabet. A DCG is described using predicates. For each predicate `p/2` (or `p/3`) of the form `p(X,Y)` (or `p(c,X,Y)`), `X` is a list representing the sequence to analyse (input) and `Y` is the remaining part of the list `X` when its prefix matching the predicate `p` (or property `c` of the predicate `p`) is removed (output). In the context-free grammar (CFG) formalism, a target concept is defined with a set of rules.

Here our target predicate is `genre/4`, where `genre(g, A,B,Key)` means the song `A` (represented as its full list of chords) in the tonality `Key` belongs to genre `g`. The argument `B`, the output list (i.e. an empty list) is necessary to comply with the definite-clause grammar representation. We are interested in degrees and chord categories to characterise a chord sequence. So the predicates considered to build the rules are `degreeAndCategory/5` and `gap/2`, defined in the background knowledge (cf. Table 1). `degreeAndCategory(d,c,A,B,Key)` means that the first chord of the list `A` has degree `d` and category `c`. The `gap/2` predicate matches any chord sequence of any length, allowing to skip uninteresting subsequences (not characterised by the grammar rules) and to handle large sequences (for which otherwise we would need very large grammars). In addition we constrain the system to use at least two consecutive `degreeAndCategory` predicates between two `gap` predicates. This guarantees that we are considering local chord sequences of a least length 2 (but also larger) in the songs.

```
rootNote(c_,[c|T],T,Key).      rootNote(c_,[cm|T],T,Key).
rootNote(c_s,[cs|T],T,Key).    rootNote(c_s,[csm|T],T,Key).
...                            ...
category(min,[cm|T],T).        category(maj,[c|T],T).
category(min,[csm|T],T).       category(maj,[cs|T],T).
...                            ...
degree(1_,A,B,cmajor) :- rootNote(c_,A,B,cmajor).
degree(1_s,A,B,cmajor) :- rootNote(c_s,A,B,cmajor).
...
degreeAndCategory(Deg,Cat,A,B,Key) :-
degree(Deg,A,B,Key), category(Cat,A,B).

gap(A,A).
gap([_,A],B) :- gap(A,B).
```

Table 1. Background knowledge predicates used in the first-order logic decision tree induction algorithm. For each chord in a chord sequence its root note is identified using the `rootNote/4` predicate. The degrees are defined using the `degree/4` predicate and the key. The chord categories are identified using the `category/3` predicate and finally degrees and categories are united in a single predicate `degreeAndCategory/5`.

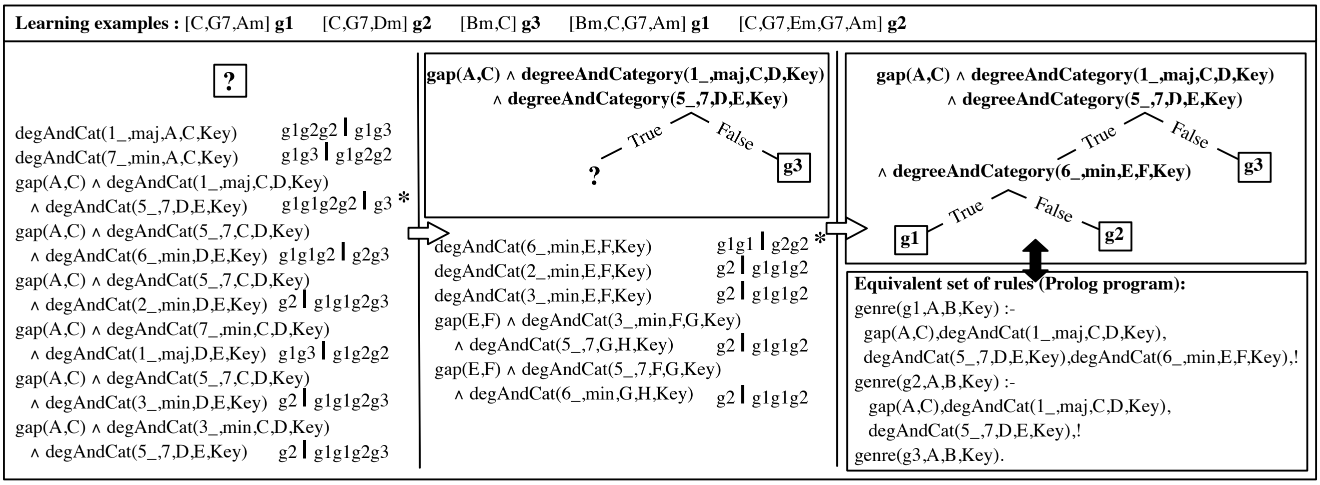

Figure 1. Schematic example illustrating the induction of a first-order logic tree for a 3-genre classification problem (based on the 5 learning examples on top). At each step the partial tree (top) and each literal (or conjunction of literals) considered for addition to the tree (bottom) are shown together with the split resulting from the choice of this literal (e.g. `g1g1g2|g2` means that two examples of g1 and one of g2 are in the left branch and one example of g2 is in the right branch). The literal resulting in a the best split is indicated with an asterisk. The final tree and the equivalent ordered set of rules (or Prolog program) are shown on the right. The key is C Major for all examples. For space reasons `degAndCat` is used to represent `degreeAngCategory`.

An example of a simple and short grammar rule we can get using this formalism is:

genre(genre1,A,B,Key) :- gap(A,C),degreeAndCategory(5_,7,C,D,Key), degreeAndCategory(1_,maj,D,E,Key),gap(E,B).

Which can be translated as : *"Some music pieces of genre1 contain a dominant 7th chord on the dominant followed by a major chord on the tonic"* (i.e. a perfect cadence).

But more complex rules combining several local patterns (of any length larger than or equal to 2) separated by `gaps` can also be constructed with this formalism.

3.2 Learning algorithm

To induce the harmony grammars we apply TILDE's decision tree induction algorithm [10]. TILDE is a first order logic extension of the C4.5 decision tree algorithm [11]. Like C4.5 it is a top-down decision tree induction algorithm: at each step the test resulting in the best split is used to partition the examples. The difference is that at each node of the trees instead of attribute-value pairs, conjunctions of literals are tested. TILDE uses by default the gain-ratio criterion [11] to determine the best split and the post-pruning is the one from C4.5. TILDE builds first-order logic decision trees which can also be represented as ordered sets of rules (or Prolog programs). In the case of classification, the target predicate of each model represents the classification problem. A simple example illustrating the induction of a tree from a set of examples covering three genres is given in Figure 1.

First-order logic enables us to use background knowledge (which is not possible with non relational data mining algorithms). It also provides a more expressive way to represent musical concepts/events/rules which can be transmitted as they are to the users. Thus the classification process can be made transparent to the user.

4. EXPERIMENTS AND RESULTS

4.1 Training data

4.1.1 Audio data

The data used in the experiments reported in this paper has been collected, annotated and kindly provided by the Pattern Recognition and Artificial Intelligence Group of the University of Alicante. It consists in a collection of Band in a Box [1] files (i.e. symbolic files containing chords) from which audio files have been synthesised and it covers three genres: popular, jazz, and academic music. The symbolic files have been converted into a text format in which only the chord changes are available. The Popular music set contains pop, blues, and celtic (mainly Irish jigs and reels) music; jazz consists of a pre-bop class grouping swing, early, and Broadway tunes, bop standards, and bossanovas; and academic music consists of Baroque, Classical and Romantic Period music. All the categories have been defined by music experts who have also collaborated in the task of assigning meta-data tags to the files and rejecting outliers. The total amount of pieces is 856 (Academic 235; Jazz 338; Popular 283) containing a total of 120,510 chords (141 chords per piece in average, a minimum of 3 and a maximum of 522 chords per piece).

The classification tasks that we are interested in are relative to the three main genres of this dataset: academic, jazz and popular music. For all our experiments we consider each time the 3-way classification problem and each of the 2-way classification problems. In addition we also study the 3-way classification problem dealing with the popular music subgenres (blues, celtic and pop music). We do not work on the academic subgenres and jazz subgenres as these two datasets contain very unbalanced subclasses,

[1] http://www.pgmusic.com/products_bb.htm

some of them being represented by only a few examples. Because of this last characteristic removing examples to get the same number of examples per class would lead to poor models built on too few examples. Finally resampling can not be used as TILDE automatically removes identical examples.

For each classification task we perform a 5-fold cross-validation. The minimal coverage of a leaf (a parameter in TILDE) is set to 5.

academic/jazz/popular	Root Int	D&C 3	D&C 7th
Accuracy (baseline = 0.40)	0.619	0.759	0.808
Stderr	0.017	0.015	0.014
# nodes in the tree	40.8	31.0	18.4
# literals in the tree	66.2	90.6	50.8
academic/jazz	Root Int	D&C 3	D&C 7th
Accuracy (baseline = 0.59)	0.861	0.872	0.933
Stderr	0.014	0.014	0.011
# nodes in the tree	11.0	16.4	10.4
# literals in the tree	19.0	46.0	30.8
academic/popular	Root Int	D&C 3	D&C 7th
Accuracy (baseline = 0.54)	0.731	0.824	0.839
Stderr	0.020	0.017	0.016
# nodes in the tree	17.0	12.4	11.0
# literals in the tree	27.6	36.4	31.8
jazz/popular	Root Int	D&C 3	D&C 7th
Accuracy (baseline = 0.55)	0.828	0.811	0.835
Stderr	0.015	0.016	0.015
# nodes in the tree	13.4	17.0	10.6
# literals in the tree	23.2	50.6	29.0
blues/celtic/pop	Root Int	D&C 3	D&C 7th
Accuracy (baseline = 0.36)	0.709	0.703	0.746
Stderr	0.027	0.028	0.026
# nodes in the tree	11.4	16.2	14.0
# literals in the tree	20.4	45.8	40.4

Table 2. Classification results on manual chord transcriptions using a 5-fold cross-validation. The number of nodes and literals present in a tree gives an estimation of its complexity. "Root Int" refers to the root intervals representation scheme. "D&C 3" and "D&C 7th" refers to the degree and chord category representation scheme respectively applied on triads only and on triads and seventh chords.

4.1.2 Chord transcription

The chord transcription algorithm based on harmonic pitch class profiles (HPCP [12]) we apply is described in [13]. It distributes spectral peak contributions to several adjacent HPCP bins and takes peak harmonics into account. In addition to using the local maxima of the spectrum, HPCPs are tuning independent (i.e. the reference frequency can be different from the standard tuning), and consider the presence of harmonic frequencies. In this paper, the resulting HPCP is a 36-bin octave independent histogram representing the relative intensity of each 1/3 of the 12 semitones of the equal tempered scale. We refer to [13] for a detailed description of the algorithm.

The algorithm can be tuned to either extract triads (limited to major and minor chords) or triads and seventh chords (limited to major seventh, minor seventh and dominant seventh). Other chords such as diminished and augmented chords are not included in the transcription (as in many transcription systems) because of the tradeoff between precision and accuracy. After pre-processing, only the chord changes (i.e. when either the root note or the chord category is modified) are kept. Notice that when dealing with the symbolic files (manual transcription) the mapping between the representations is based on the third (major or minor). Since only the chord changes were available in the symbolic files (no timing information) it was not possible to compute the transcription accuracy.

4.2 Validating our new harmony representation scheme

We first study if our new harmony representation scheme based on degrees and chord categories (D&C) can compete with our previous representation scheme based on root intervals (Root Int.). For that we test these two harmony representations on clean data, i.e. on the manual chord transcriptions. We test the degree and chord category representation scheme on both triads-only (D&C 3) and triads and seventh manual transcriptions (D&C 7th). The results (i.e. test results of the 5-fold cross-validation) of these experiments are shown in Table 2.

The D&C representation scheme obtains better results, with accuracies always as high as or higher than the root interval representation scheme classification accuracies. Furthermore the complexity of the models is not increased when using the D&C representation compared to the root interval representation. Indeed, the number of nodes and literals in the built models (trees) are comparable. Using the seventh chord categories leads to much higher accuracies, lower standard errors and lower complexity than when only using the triads.

We also tested these representation schemes when the learning examples are audio files (cf. Section 4.3 for more details on these experiments). However the root interval experiments on audio data were so slow that we were unable to complete a 5-fold cross-validation. We estimate the time needed to build one (2-class) model based on the root interval audio data to 12 hours in average, whereas only 10 to 30 minutes are needed to build a D&C 3 (2-class) model on audio data and around 1 hour and a half for a D&C 7th (2-class) model. In conclusion the degree and category representation scheme outperforms the root interval representation scheme on both classification accuracy and run times.

4.3 Performances on audio data

We now test if our first-order logic classification framework can build good classification models when the learning examples are automatic chord transcriptions from audio files (i.e. noisy data). This is essential for the many applications in which no symbolic representation of the harmony is available. The results of this framework when using the degree and chord category representation scheme on audio data are shown in Table 3.

academic/jazz/popular	D&C 3	D&C 7th
Accuracy (baseline = 0.39)	0.582	0.575
Stderr	0.017	0.017
# nodes in the tree	59.2	66.8
# literals in the tree	171.2	198.4
academic/jazz	D&C 3	D&C 7th
Accuracy (baseline = 0.59)	0.759	0.743
Stderr	0.018	0.018
# nodes in the tree	26.4	31.8
# literals in the tree	76.0	93.8
academic/popular	D&C 3	D&C 7th
Accuracy (baseline = 0.55)	0.685	0.674
Stderr	0.020	0.021
# nodes in the tree	25.8	26.4
# literals in the tree	72.2	74.0
jazz/popular	D&C 3	D&C 7th
Accuracy (baseline = 0.54)	0.789	0.773
Stderr	0.016	0.017
# nodes in the tree	22.4	28.8
# literals in the tree	66.0	86.0
blues/celtic/pop	D&C 3	D&C 7th
Accuracy (baseline = 0.35)	0.724	0.668
Stderr	0.027	0.028
# nodes in the tree	13.2	14.8
# literals in the tree	38.8	43.2

Table 3. Classification results on audio data using a 5-fold cross-validation.

Although the accuracies are still good (significantly above the baseline), it is not surprising that they are lower than the results obtained for clean data (i.e. manual transcriptions). The noise introduced by the automatic chord transcription also leads to a higher complexity of the models derived from audio data. Also using the seventh chords leads to slightly less accurate models than when using triads only. The opposite result was obtained with the manual transcription data, where the seventh chord representation scheme outperformed the triads representation scheme. We surmise that the reason for this difference is the fact that the automatic chord transcription algorithm we use is much less accurate when asked to use seventh chords than when asked to use triads only.

Concerning the classification tasks, all the 2 and 3-class problems are solved with accuracies well above chance level. The 3-class popular music subgenres classification problem seems particularly well handled by our framework with 72% and 67% accuracy when using respectively triads and seventh chords. The best 2-class classification results (between 74% and 79% accuracy) are obtained when trying to distinguish jazz from another genre (academic or popular). Indeed the harmony of classical and popular music can be very similar, whereas jazz music is known for its characteristic chord sequences, very different from other genres harmonic progressions.

4.4 Transparent classification models

To illustrate the transparency of the classification models built using our framework we present here some interesting rules with high coverage extracted from classification models generated from symbolic data. Notice that the classification models are trees (or ordered sets of rules), so a rule in itself can not perform classification both because of having a lower accuracy than the full model and because the ordering of rules in the model is important to the classification (i.e. some rule might never be used on some example because one of the preceding rules in the model covers this example). To illustrate this for each of the following example rules we provide its absolute coverage (i.e. if the order was not taken into account) on each genre.

The following rule was found in the popular subgenres classification models:
[*coverage: blues=42/84; celtic=0/99; pop=2/100*]
genre(blues,A,B,Key) :-
gap(A,C),degreeAndCategory(1_,7,C,D,Key),
degreeAndCategory(4_,7,D,E,Key),gap(E,B).
"Some blues music pieces contain a dominant seventh chord on the tonic directly followed by a dominant seventh chord on the subdominant (IV)."

The following rules were found in the academic/jazz/popular classification models:
[*cov.: jazz=273/338; academic=42/235; popular=52/283*]
genre(jazz,A,B,Key) :-
gap(A,C),degreeAndCategory(2_,min7,C,D,Key),
degreeAndCategory(5_,7,D,E,Key),gap(E,B).
"Some jazz music pieces contain a minor seventh chord on the supertonic (II) directly followed by a dominant seventh chord on the dominant."
[*cov.: jazz=173/338; academic=1/235; popular=17/283*]
genre(jazz,A,B,Key) :-
gap(A,C),degreeAndCategory(6_,7,C,D,Key),
degreeAndCategory(2_,min7,D,E,Key),gap(E,B)
"Some jazz music pieces contain a dominant seventh chord on the submediant (VI) directly followed by a minor seventh chord on the supertonic (II)."

Finally the following rules were found in the academic/jazz classification models:
[*cov.: academic=124/235; jazz=6/338; popular=78/283*]
genre(academic,A,B,Key) :-
gap(A,C),degreeAndCategory(1_,maj,C,D,Key),
degreeAndCategory(5_,maj,D,E,Key),gap(E,B).
"Some academic music pieces contain a major chord on the tonic directly followed by a major chord on the dominant."
[*cov.: academic=133/235; jazz=10/338; popular=68/283*]
genre(academic,A,B,Key) :-
gap(A,C),degreeAndCategory(5_,maj,C,D,Key),
degreeAndCategory(1_,maj,D,E,Key),gap(E,B).
"Some academic music pieces contain a major chord on the dominant directly followed by a major chord on the tonic."
Note that the lack of sevenths distinguishes this last common chord change from its jazz counterparts. Indeed the following rule has a high coverage on jazz:
[*cov.: jazz=146/338; academic=0/235; popular=15/283*]
genre(jazz,A,B,Key) :-
gap(A,C),degreeAndCategory(5_,7,C,D,Key),
degreeAndCategory(1_,maj7,D,E,Key),gap(E,B).

5. CONCLUSION AND FUTURE WORK

In this paper we showed that our genre classification framework based on harmony and first-order logic and previously tested on symbolic data in [3] can also directly learn classification models from audio data that obtain a classification accuracy well above chance level. The use of a chord transcription algorithm allows us to adopt a high-level representation of harmony even when working on audio data. In turn this high-level representation of harmony based on first-order logic allows for human-readable, i.e. transparent, classification models. We increased this transparency by introducing a new harmony representation scheme, based on the western representation of harmony which describes the chords in terms of degrees and chord categories. This representation is not only musically more meaningful than a previous representation we adopted, it also got better classification results and the classification models using it were built faster. Testing our model on manual transcriptions we observed that using seventh chords in the transcription task could considerably increase the classification accuracy. However the automatic transcription algorithm we used for these experiments was not enough accurate when using seventh chords and we could not observe such improvements when using audio data.

Future work includes testing several other chord transcription algorithms to see if they would lead to better classification models when using seventh chords. We also plan to use these chord transcription algorithms to study how the accuracy of classification models built on transcriptions evolves with the accuracy of these transcriptions. In addition the audio data used in these experiments was generated with MIDI synthesis. This is generally cleaner than CD recordings, so we expect a further degradation in results if we were to use audio recordings. Unfortunately we do not possess the corresponding audio tracks that would allow us to make this comparison. We intend to look for such recordings and extend our audio tests to audio files that are not generated from MIDI. Finally with these experiments we showed that a classification system based only on chord progressions can obtain classification results well above chance level. If such a model based only on one dimension of music (harmony) can not compete on its own with state-of-the-art classification models, we believe – and intend to test this hypothesis in future experiments – that if such an approach is combined with classification models based on other dimensions (assumed orthogonal) such as rhythm and timbre we will improve on state-of-the-art classification accuracy.

6. ACKNOWLEDGMENTS

We would like to thank the Pattern Recognition and Artificial Intelligence Group of the University of Alicante for providing the data. This work is supported by the EPSRC project OMRAS2 (EP/ E017614/1) and the Spanish TIN project ProSeMus (TIN2006-14932-C02-01). During her internship at the Music Technology Group the first author was supported by the EPSRC Platform grant EP/E045235/1.

7. REFERENCES

[1] G. Tzanetakis, A. Ermolinskiy, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. In *Proceedings of ISMIR 2002*, Paris, France, 2002.

[2] W. Piston. *Harmony*. Norton, W. W. & Company, Inc., 5th edition, 1987.

[3] Amélie Anglade, Rafael Ramirez, and Simon Dixon. First-order logic classification models of musical genres based on harmony. In *Proceedings of the 6th Sound and Music Computing Conference*, Porto, Portugal, 2009.

[4] T. Yoshioka, T. Kitahara, K. Komatani, T. Ogata, and H. G. Okuno. Automatic chord transcription with concurrent recognition of chord symbols and boundaries. In *Proceedings of ISMIR 2004*, pages 100–105, Barcelona, Spain, 2004.

[5] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):291–301, 2008.

[6] M.-K. Shan, F.-F. Kuo, and M.-F. Chen. Music style mining and classification by melody. In *Proceedings of 2002 IEEE International Conference on Multimedia and Expo*, volume 1, pages 97–100, 2002.

[7] J.-F. Paiement, D. Eck, and S. Bengio. A probabilistic model for chord progressions. In *Proceedings of ISMIR 2005*, pages 312–319, London, UK, 2005.

[8] K. Lee. A system for automatic chord transcription using genre-specific hidden markov models. In *Proceedings of the International Workshop on Adaptive Multimedia Retrieval*, Paris, France, 2007.

[9] C. Perez-Sancho, D. Rizo, S. Kersten, and R. Ramirez. Genre classification of music by tonal harmony. In *International Workshop on Machine Learning and Music*, Helsinki, Finland, 2008.

[10] Hendrik Blockeel and Luc De Readt. Top down induction of logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, 1998.

[11] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[12] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the 1999 International Computer Music Conference*, pages 464–467, Beijing, China, 1999.

[13] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, MTG, Universitat Pompeu Fabra, 2006.