TOWARDS A HUMAN-FRIENDLY MELODY CHARACTERIZATION BY AUTOMATICALLY INDUCED RULES

Pedro J. Ponce de León, David Rizo, José M. Iñesta
Universidad de Alicante, Spain
Departamento de Lenguajes y Sistemas Informáticos
{pierre,drizo,inesta}@dlsi.ua.es

ABSTRACT

There is an increasing interest in music information retrieval for reference, motive, or thumbnail extraction from a piece in order to have a compact and representative representation of the information to be retrieved. One of the main references for music is its melody. In a practical environment of symbolic format collections the information can be found in standard MIDI file format, structured as a number of tracks, usually one of them containing the melodic line, while the others contain the accompaniment. The goal of this work is to analyse how statistical rules can be used to characterize a melody in such a way that one can understand the solution of an automatic system for selecting the track containing the melody in such files.

1 INTRODUCTION

Digital music scores can be found in digital collections in a number of ways. One of the most popular formats is the standard MIDI file, where the information is structured in such a way that the melody part is stored in one or more tracks, often separately from the rest of the musical content. This is frequently the case of modern popular music.

The literature about melody voice identification in the symbolic domain is quite poor. Ghias et al. [2] built a system to process MIDI files extracting a sort of melodic line using simple heuristics. Tang et al. [7] present a work where the aim is to propose candidate melody tracks, given a MIDI file. They take decisions based on single features derived from informal assumptions about what a melody track should be. Other works in this line have been recently published at the light of recent developments [6] like that of Madsen and Widmer [4] that uses information theory measures like entropy to approach the problem.

The large-term goal of this work is to pose the general question *What is a melody?* under a statistical approach. The answer would be given as sets of rules that are both human-readable and automatically learnt from score corpora. This could be of interest for musicologists or helpful in applications such as melody matching, motif extraction, melody extraction (ringtones), etc. In this paper the first stages of this work are presented.

2 METHODOLOGY

The steps performed to obtain sets of rules that characterize melody tracks can be outlined as follows:

Feature extraction Describe a MIDI track by a set of low-level statistical descriptors.

Decision tree learning Learn a set of decision trees from a tagged corpus of MIDI-tracks.

Rule extraction Extract rules from decision tree branches.

Rule simplification Prune rule antecedents.

Rule selection Select best rules by ranking.

The details about the first two steps can be found in [6]. The low-level statistical descriptors utilized to describe musical content are based on several categories of features that assess melodic and rhythmic properties of a music sequence, such as pitch, note interval, and note duration distribution descriptors, as well as track related properties such as polyphony rate, duration, etc. Most distribution descriptors are also present in normalized form with respect to the value range for that descriptor in the whole song. This allows to know the context in which a particular descriptor value is computed.

The sets of rules that characterize melody tracks are initially extracted from an ensemble of decision trees that has shown good performance at this task [6]. The rules are simplified by pruning non useful antecedents and then, for each rule set, rules are ranked according to different metrics.

Note that there is no feature selection stage, as the Random Forest [1] method used to learn decision tree ensembles performs its own feature selection process.

Expressing a concept by rules has the advantage of being a human-readable description of the characterization process. The simplification and ranking steps focus on obtaining a small, manageable rule system. A side goal has been to test whether a small number of selected rules can perform comparably to an ensemble of decision trees, typically containing hundreds or thousands of nodes.

2.1 Rule extraction from decision trees

In a previous work [6], Random Forest classifiers were used to learn an ensemble of decision trees capable of discriminating melody tracks in a MIDI file.

The datasets used to learn the trees in a Random Forest are collections of MIDI files where tracks are labeled with a boolean tag indicating whether the track contains a melody line or not. Thus, the leaves of the decision trees are also tagged with a boolean value, so there are positive and negative leaves. For each tree, a rule set is extracted. Rules are extracted following positive branches (leading to a positive leaf). Negative branches are ignored. From each positive branch a rule is obtained:

$$if\ (X_1 \wedge X_2 \wedge ... \wedge X_n) \quad then \quad TrackIsMelody$$

where X_i are the tests found in each tree node traversed following a positive branch. As all rules have the same consequent, we will drop it from herein.

2.1.1 Track characterization by rule set ensemble

A rule set is built as the disjunction of all the rules extracted from the same tree. No particular rule ordering is imposed on a rule set. When such a rule set is applied to a sample, firing at least one rule suffices for that sample to be tagged as a melody track. From a statistical viewpoint, each rule in the set stands for a certain type of melody.

An ensemble of K rule sets is obtained from the learnt decision trees. As each rule set outputs a decision, the whole ensemble is applied to a sample calculating the ratio between positive decisions and K. This value is taken as the probability for a track to be a melody track.

2.2 Rule set simplification

The number of conditions in the antecedent part of a rule can be too large to be easily read. Moreover, complex rules are often very specific, overfitting the training set. A rule can be generalized by dropping some of the conditions from its antecedent. This process is explained below.

On the other hand, the decision trees learnt by the Random Forest classifier from large training sets are usually big, leading also to huge rule sets [1] , so it is desirable to remove rules that do not characterize a lot of samples. Furthermore, a given sample in a dataset could fire more than one rule, thus making the presence of some rules in the set not necessary. The method used here to reduce the size of a rule set consists of ranking rules according to a measure based on how many samples from a validation dataset fire them. The more samples (MIDI tracks) fire a rule, the better the rule. After the ranking is made, the best N rules are selected from each rule set in order to classify new samples, discarding the rest.

2.3 Antecedent pruning

The method used here is to perform a test for consequent's independence from each condition X_i for each rule [5]. A χ^2 test with a 95% confidence interval is performed considering condition relevance as independent from other conditions in the same rule. This makes the test to be very conservative, dropping only conditions that do not satisfy the test's hypothesis for all validation samples. A validation dataset different from the initial training set is used to test single rule conditions.

2.4 Rule selection

Once rules are simplified using the method from the previous section, a rule ranking is established in order to select the best rules in each rule set. The procedure used in this work to rank the rules in a rule set R using a validation dataset D is as follows:

1. Sort rules in R decreasingly, according to the number of samples in the training set firing each rule.
2. For each rule r^i in R: [2]
   (a) $r^i_{score} = Score(r^i, D)$
   (b) $D = D - r^i_{\oplus} - r^i_{\ominus}$
3. Sort R according to r_{score}, in decreasing order.
4. Select the first N rules of R and discard the rest.

There are several choices for the scoring function. In this work, two functions have been tested:

- $Score_1(r^i, D) = r^i_{\oplus}$
- $Score_2(r^i, D) = r^i_{\oplus}/(r^i_{\oplus} + r^i_{\ominus})$

Additionally, a variant of the ranking procedure that does not remove samples from D (step 2b) has been tested along with the $Score_1$ function. We denote this variant as $Score_0$.

The same validation dataset is used for all rule sets in the rule system derived from the decision trees.

3 EXPERIMENTS AND RESULTS

For the experiments presented here, rule systems have been derived from an ensemble of decision trees built using a Random Forest classifier with $F = 5$ features and $K = 10$ trees. Therefore rule systems consist of ten rule sets. There is one rule system per training corpus.

3.1 Datasets

Four corpora have been used: *SMALL*, *LARGE*, *RWC-G* and *RWC-P*. The corpora *SMALL* and *LARGE* are described in [6] [3] . They contain songs of classical, jazz and popular music downloaded from a number of freely accessible Internet sites. These two corpora have been used to train and validate the system. The *RWC* corpora, used to test the system, are MIDI files from the well-known *RWC Popular (RWC-P)* and *Genre (RWC-G) Music Databases* [3]. Tracks in these corpora (detailed in table 1) have been manually tagged as melody or non-melody tracks.

[1] In our experiments, trees with more than 500 leaves (and thousands of nodes) have been obtained.

[2] $r^i_{\oplus}$ is the number of melody tracks in D that fire rule r^i and $r^i_{\ominus}$ is the number of non-melody tracks in D that fire rule r^i.

[3] Where *SMALL* is named as *ALL200* and *LARGE* is the union of the *CLA*, *JAZ* and *KAR* corpora in that paper.

Corpus	Tracks	Melodies	Songs	Validation set
SMALL	2775	554	600	LARGE
LARGE	15168	2337	2513	SMALL
RWC-G	311	44	48	–
RWC-P	801	74	75	–

Table 1. Structure of the corpora

3.2 Antecedent pruning results

The results of the antecedent pruning process are summarized in table 2.

Rule system	SMALL	LARGE
Uniq. cond.	779 (816)	1878 (2297)
% pruning uniq.	4.5%	18.2%
Cond.	4370 (4692)	22481 (25581)
Avg. cond./rule	8.3 (9)	10.8 (12.3)
% pruning	6.9%	12.1%

Table 2. Rule antecedent pruning results. Numbers in parentheses indicate number of conditions before pruning.

As the figures in the table are not that impressive, recall from section 2.3 that the testing procedure is very conservative, removing only conditions that are not relevant for the whole validation dataset. Another fact that contribute to the small pruning percentage achieved is that, for a given descriptor, there are a lot of similar but not equal conditions, as we are using two digit precision in floating point numbers.

For the experiments that follow, the resulting pruned rule systems are used.

3.3 Rule ranking results

Table 3 summarizes rule coverage of the validation dataset when considering all rules in a rule set. Figures in the second and third rows are average percentages of the validation dataset covered by a rule set from the corresponding rule system. Rule coverage percentages are a rough estimation of a rule system accuracy. In this case, both rule systems perform comparably on their respective validation datasets, with more than 85% of the melody tracks firing at least one rule in a set. However, the coverage of non-melody tracks is somewhat high, especially for the *LARGE* rule system.

Rule system	SMALL	LARGE
melody tracks	86%	87%
non-melody tracks	17%	32%
zero scoring	23%	67%

Table 3. Rule coverage summary.

The last row shows the percentage of rules in a rule set, in average, that scored zero, i.e. rules not fired by any melody track. Note that these percentages are the same for the two scoring functions used. These rules are dropped from the rule sets, thus reducing their size in the *LARGE* rule system to one third. This means keeping about 70 rules out of more than 200 for each rule set, in average.

Figure 1 shows the average rule set coverage percentages when selecting the N best rules from each rule set in a rule system. Note that when N is small, the $Score_1$ function gives a better coverage ratio for melodies. On the other hand, the $Score_2$ function gives better coverage ratios for large N. In particular, the *LARGE* rule system achieves more than 75% melody track coverage in average while maintaining the non-melody track coverage very low (about 2%).

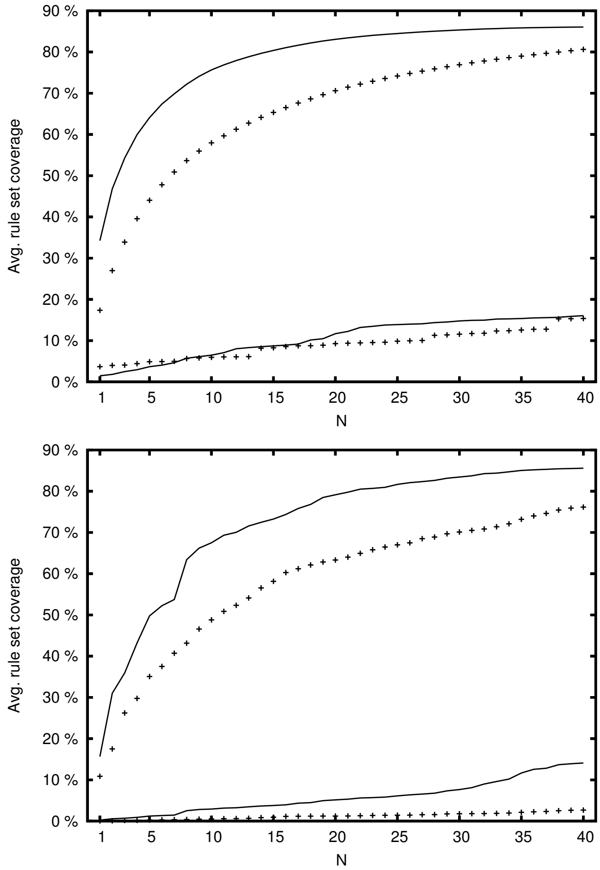

Figure 1. Average rule set coverage when the number of rules selected (N) varies from 1 to 40, for both rule systems and score functions $Score_1$ (top) and $Score_2$ (bottom). Lines indicate coverage for *SMALL* and (+) indicate coverage for *LARGE*. Plots over 20% are for melody tracks. Plots under 20% are for non-melody tracks.

3.4 Track selection procedure

Two experiments are presented. In the first one, all tracks from test MIDI files are classified as melody or non-melody track. In the second one, a single track is selected from a MIDI file as its melody track. Therefore, given a file, all its non-empty tracks are classified and their probabilities of being a melody are obtained. Then the track with the highest probability is selected as the melody track. If all tracks have zero probability, no melody track is selected.

For each experiment, several possibilities are explored and the rule system results are compared to those obtained by the Random Forest from which the rules are derived.

3.5 Experiments

3.5.1 Melody versus non-melody classification

The *RWC-G* and *RWC-P* datasets have been used as test sets. The rule systems have three free parameters: the number of best rules N to be selected from each rule set, the minimum number θ of rule sets that must agree to tag a track as a melody, and the scoring function used to rank the rules in the sets. Each combination of N, θ and the scoring function results in a distinct rule system classifier. Recall that there are 10 rule sets in a rule system. The range of values used in these experiments is $N \in 1..20$, $\theta \in 1..10$ and three different scoring functions. A summary of the best results obtained applying these classifiers to the *RWC* datasets is summarized in tables 4 and 5. The F-measure is used as the accuracy measure.

	Sco	N	θ	%OK	Prec	Rec	F
SMALL	0	2	2	91.3	0.67	0.75	**0.71**
(rules)	1	9	2	78.1	0.39	0.93	0.70
	2	17	4	89.4	0.60	0.80	0.68
SMALL(RF)	–	–	*6*	*90.4*	*0.65*	*0.68*	*0.67*
LARGE	0	17	3	87.5	0.54	0.75	0.63
(rules)	1	14	3	88.7	0.57	0.84	0.68
	2	8	3	92.3	0.76	0.66	0.71
LARGE(RF)	–	–	*6*	*92.0*	*0.71*	*0.72*	***0.72***

Table 4. Best results for the *RWC-G* dataset. (Sco = score; RF = Random Forest; reference results in italics)

	Sco	N	θ	%OK	Prec	Rec	F
SMALL	0	1	3	98.1	0.89	0.91	0.90
(rules)	1	5	6	98.1	0.90	0.89	0.90
	2	2	2	98.4	0.88	0.96	**0.91**
SMALL(RF)	–	–	*6*	*97.8*	*0.84*	*0.93*	*0.89*
LARGE	0	3	3	98.6	0.91	0.95	0.93
(rules)	1	9	4	98.8	0.90	0.97	**0.94**
	2	5	2	98.8	0.90	0.97	**0.94**
LARGE(RF)	–	–	*6*	*98.6*	*0.90*	*0.97*	***0.94***

Table 5. Best results for the *RWC-P* dataset.

3.5.2 Per-song melody track selection

Now, the goal is to know how many times the method selects as melody track a proper one among those in a file. For each MIDI file, the classifiers output the track with the highest probability of being a melody, except when all these probabilities are zero, in which case the system says that the file has no melody track.

An answer is considered as a success if:

1. The file has at least one track tagged as *melody* and the selected track is one of them.
2. The file has no melody tracks and the answer is that there is no melody track.

Table 6 shows the best rule system results and those using Random Forest classifiers. The results for the *RWC-P* dataset show us a hint that rule systems with a few tens of rules can perform comparably to ensemble of decision trees with hundreds of leaves.

	RWC-G			*RWC-P*		
	Sco	N	%OK	Sco	N	%OK
SMALL	1	3	70.5	0	2	**97.3**
SMALL (RF)	–	–	***75.0***	–	*6*	*94.7*
LARGE	1	5	68.2	0	6	**97.3**
LARGE (RF)	–	–	***72.9***	–	*6*	*96.0*

Table 6. Best melody selection results for *RWC-G* (left) and *RWC-P* (right).

4 CONCLUSIONS AND FUTURE WORK

In this work a method to obtain reduced rule systems from previously learnt random forests that characterize melody tracks has been exposed. Such rule systems perform comparably to the original decision tree ensembles.

To be able to give a more human-readable description a fuzzyfication process is required on the sets of rules. Currently we are working on this process.

The study of these rules can lead also to the description of other track categories, such as *solo* or *chorus* tracks.

5 ACKNOWLEDGMENTS

Supported by the projects: GV06/166 and CICyT TIN2006–14932–C02, partially supported by EU ERDF. The authors want to thank Andrés Pérez for his tagging help.

6 REFERENCES

[1] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[2] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: Musical information retrieval in an audio database," in *Proc. of 3rd ACM Int. Conf. Multimedia*, 1995, pp. 231–236.

[3] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "Rwc music database: Music genre database and musical instrument sound database." in *ISMIR*, 2003.

[4] S. T. Madsen and G. Widmer, "Towards a computational model of melody identification in polyphonic music." in *IJCAI*, 2007, pp. 459–464.

[5] J. R. Quinlan, "Simplifying decision trees." *Int. J. Hum.-Comput. Stud.*, **51**(2), pp. 497–510, 1999.

[6] D. Rizo, P. J. Ponce de León, C. Pérez-Sancho, A. Pertusa, and J. M. Iñesta, "A pattern recognition approach for melody track selection in midi files," in *Proc. of ISMIR 2006*, Victoria, Canada, 2006, pp. 61–66.

[7] M. Tang, C. L. Yip, and B. Kao, "Selection of melody lines for music databases." in *Proceedings of Annual Int. Computer Software and Applications Conf. COMPSAC*, 2000, pp. 243–248.