MIREX 2005: Symbolic Genre classification with an ensemble of parametric and lazy classifiers

Pedro J. Ponce de León
Computer Music Laboratory
Dept. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
E-03080 Alicante, Spain
pierre@dlsi.ua.es

José M. Iñesta
Computer Music Laboratory
Dept. de Lenguajes y Sistemas Informáticos
Universidad de Alicante
E-03080 Alicante, Spain
inesta@dlsi.ua.es

ABSTRACT

The symbolic genre classification algorithm submited to the MIREX (Music Information Retrieval Exchange) 2005 is described here. Our algorithm uses a combination of k-nearest neighbors and Bayesian classifiers trained with different sets of statistical descriptors extracted from melody tracks extracted from MIDI files. It is aimed at classifying melodies by genre. The statistical descriptors describe pitch, note duration, silence duration, and rhythmic properties of the melody. The set of descriptors is invariant to transposition or tempo scaling and deliberately contains no information based on metadata, such as instrumentation or text data. Descriptors consist mainly in counters, range, average, standard deviation of musical properties. Each track is reduced to a monophonic sequence of notes, prior to the extraction of descriptors. Classifiers are trained independently using different subsets of descriptors. The resulting models are combined using a majority vote scheme. In order to select a melody track from a MIDI file, a model of 'melody track' previously trained is applied to each track in a MIDI file. The most probable melody track is selected and used as an instance for the different classifier ensembles. Therefore each MIDI file is classified using information based on a single track.

Keywords: ISMIR, Symbolic genre classification, classifier ensemble, shallow description

1 MELODY TRACK IDENTIFICATION

The concept of a melody voice and accompaniment must be defined prior to be able to select the melodic track from a MIDI file. There are some features in a melodic track that, at first sight, seem to be definitive to identify it, like having high pitches or being monophonic[1]. Unfortunately, it is common to find a melodic line that has not the highest average pitch of the song, or that contains some chords.

To overcome these problems, a classifier that learns what is a melodic track and what is not was utilized. The WEKA (Ian H. Witten, 1999) toolkit was chosen to build the system, and it was extended to read the track descriptors proposed in section 1.1 directly from MIDI files.

1.1 Track description

The content of a track is characterized by a vector of statistical descriptors based on descriptive statistics that summarize track content information. This kind of statistical description of musical content is sometimes referred to as *shallow structure description* (Pickens, 2001). A set of 12 descriptors has been defined, based on several categories of features that assess melodic and rhythmic properties of a music sequence, as well as track properties. These descriptors showed evidence of statistical significance when comparing their distribution for melody and not-melody tracks. Other considered descriptors did not show significant difference when comparing their per-class distribution, so they have not been used in the experiments described.

For training purposes, each track is labelled as being a melody track or not. A list of the descripors used in this work is shown below.

- Track descriptors
  - Normalized track length
  - Polyphony rate
  - Occupation rate
- Pitch descriptors
  - Highest normalized pitch
  - Lowest normalized pitch
  - Average normalized pitch
- Note duration descriptors
  - Highest normalized duration
  - Lowest normalized duration
- Interval[2] descriptors
  - Highest normalized absolute interval
  - Lowest normalized absolute interval
  - Average normalized absolute interval
  - Normalized number of distinct intervals

[1] In a monophonic track there can be at most one note active at any given time

[2] Distance in pitch between two consecutives notes

Track duration and note duration descriptors are computed as the ratio between the duration value in ticks and the MIDI file resolution. This way, durations are expressed as a number of beats to make them independent from the midifile resolution. Pitch, note duration, interval and track length descriptors are normalized using the formula $(value - min)/(max - min)$, where $value$ is the descriptor to be normalized, and min and max are respectively the minimum and maximum value for this descriptor for all the tracks of the target midifile. For pitch descriptors, the maximum value is fixed at 127 (the highest possible pitch value in any MIDI file), and the minimum is set to 0 (the lowest possible pitch value).

In order to characterize the degree of polyphony in a track, the polyphony rate is defined as the ratio between the number of ticks in the track where two or more notes are active, and the track duration in ticks. The occupation rate descriptor accounts for the percentage of the track length that is occupied by notes, and is defined as the ratio between the number of ticks where at least one note is active and the track length in ticks.

Pitch descriptors are the highest, lowest and average normalized pitch in the track. Note duration properties are described by the highest and lowest normalized note durations found in a track. The interval descriptors summarize information about the difference in pitch between consecutive notes. Pitch interval values are either positive or zero, when the first pitch is lower or equal to the second pitch, or negative, when the first pitch is higher than the second one. However, pitch interval information is collected as absolute values, and it is summarized as the highest, lowest and average normalized values. Finally, the number of distinct absolute interval values is counted and normalized among tracks.

To summarize, information about track content and pitch distribution, note duration distribution, and absolute interval distribution in a track is provided to describe the content of a MIDI track as a vector of real numbers, normalized between 0 and 1. This is the representation used to train the random forests classifier, as described in the next section.

1.2 Training corpora

A set consisting of 600 MIDI files was created due to the lack of existing databases for this task. There are a lot of MIDI files available on internet, but it is difficult to find tracks within them labelled as 'the melody'. One subset with jazz files, another one with classical music pieces where there is an evident melodic line, and one more for sung popular music in karaoke (.kar) format were utilized for training the random forest classifier. Each subset contains 200 MIDI files.

These files were downloaded from various internet sources. From thousands of available files, only those with some track whose name in lowercase is in the set {*melody, melodie, melodia, vocal, chant, voice, lead, lead vocal, canto*} were selected. These tracks were labelled as melodic lines. Remaining tracks were labelled as non-melodic tracks. The MIDI percussion tracks (channel 10) were removed.

The melody selection system was trained prior to the MIREX submission. In a 4-fold cross-validation experiment we did, it scored a 93% of average success in identifying melody tracks.

1.3 The random forest classifier

Random forests are a combination of tree predictors that use a random selection of features to split each node. This classifier yields error rates that compare favorably to techniques like Adaboost, but are more robust with respect to noise. The forest consists of K trees. Each tree is built using CART (Duda et al., 2000) methodology to maximum size and do not prune. The number F of randomly selected features to split on at each node is fixed for all trees. After growing the trees, new samples are classified by each tree and their results are combined, giving as a result a membership probability for each class. In our case, this is simply the probability of being a melodic line track. For MIREX, $K = 10$ trees are used, and $F = 5$ features are randomly selected to split each tree node.

Therefore, a set of descriptors is extracted from each track of a target melody, and these descriptors are the input to a classifier that assigns a melodic line probability for each track. The tracks with the highest probability is selected as the melodic line for that melody.

2 STATISTICAL DESCRIPTION FOR MUSIC GENRE CLASSIFICATION

Once a melody track is selected it must be assigned a genre. For genre classification purposes, tracks are described by a different set of statistical descriptors. A set of 28 descriptors has been defined, based on several categories of features that assess melodic, harmonic, and rhythmic properties of a melody. These descriptors are summarized in Table 1. The first column indicates the musical property analysed and the other columns indicate the kind of statistics describing the property. A blank entry in the table means that a particular statistic has not been computed.

Four different description models have been defined. The model containing all the descriptors is called the F (full) model. From this one, three reduced models have been derived. This has been achieved using a per-feature separability test described in (Ponce de León and Iñesta, 2003) to rank the features. Subsets of features are incrementally built by choosing the best ranked features. These models are called here A, B, and C for simplicity. Model A includes the six best ranked features, model B adds four features to model A, and model C adds two features to model B, so that $A \subset B \subset C \subset F$. Each entry in Table 1 indicates the smallest feature subset where the particular statistical descriptor has been included.

For the descriptor computations, the melodies are quantized to a resolution of $Q = 48$ ticks per bar. Durations are measured in ticks. For pitch and interval categories, the range descriptors are computed as the maximum minus the minimum value in the melody, and the average-relative descriptors are computed as the average value minus the minimum value. For durations (note and

Table 1: Shallow structure descriptors

Category	Counter	Range	Average-relative	Deviation	Normality
Notes	A				
Significant silences	B				
Non significant silences	F				
Pitches		A	A	A	F
Note durations		F	F	C	F
Silence durations		F	F	F	F
Inter-onset intervals		F	F	B	F
Pitch intervals		A	F	B	B
Non-diatonic notes	F		F	C	F
Syncopations	A				

silence durations, and inter-onset intervals) the range descriptors are computed as the ratio between the maximum and the minimum values, and the average-relative descriptors are computed as the ratio between the average and the minimum value. Finally, normality descriptors are computed using the D'Agostino statistic (D'Agostino and Stephens, 1986) for assessing the normality of the distribution of each property.

3 CLASSIFIER ENSEMBLE

Two different classification paradigms have been used with the four description models presented in section 2: the k-nearest-neighbour classifier, and the bayesian classifier assuming non-diagonal covariance matrices (Duda et al., 2000). For the first one, given a sample $\mathbf{x}_i$, the distances to the prototypes in the training set are computed, and the class labels of the closest k are taken into account to take the decision by a majority. A value $k = 7$ has been establish for this classifier after some trials.

In the bayesian classifier the classification is performed following the well-known *Bayes' classification rule*. In a context where there is a set of classes $c_j \in \mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$, a sample $\mathbf{x}_i$ is assigned to class c_j with maximum a posteriori probability, in order to minimize the probability of error:

$$P(c_j|x_i) = \frac{P(c_j)P(x_i|c_j)}{P(x_i)} \quad . \tag{1}$$

Using these two different classification techniques, eight different classifiers have been defined using the four shallow structure description models presented in section 2. Each classifier has been trained separately on the musical corpus and its accuracy estimated through leave-one-out cross-validation.

After analysing the performance of the different classifiers studied, we have found a diversity of errors among the decisions taken by the different classifiers. This diversity has been suggested by some authors (Kuncheva, 2003; Cunningham and Carney, 2000) as an argument for using classifier ensembles with good results. These ensembles could be regarded as committees of 'experts' (Blum, 1997) in which the decisions of individual classifiers are considered as opinions supported by a measure of confidence usually related to the accuracy of each classifier. The final classification decision is taken by majority vote. Ties are resolved picking one of the more voted genres randomly.

4 RESULTS

The *MIREX 2005 Symbolic Genre Classification* contest consisted of two classification tasks. The first was to classify by genre 950 MIDI files distributed in a genre taxonomy with 38 leaf genres. The other task consisted in classifying files distributed in a smaller genre taxonomy, with three root genres and 9 subgenres, with 25 files per subgenre. Our algorithm ranked last in the evaluation results. This was not surprising at all, as the algorithm makes use of one single track from a MIDI file to classify the whole file, therefore missing a lot of information contained in other tracks. Due to the difficulties that arose while integrating the algorithm into the M2K framework used for the contest, other versions of the algorithm, that take information from all tracks in a MIDI file to classify it, couldn't be submitted to the contest. Also, an internal cross-validation to measure the accuracy of the individual classifiers in order to weigh them and to apply a weighted voting scheme was not implemented within the M2K framework, due to time constraints. Despite the low rate results (see table 4), the accuracy of the system is several times better than random classification. However, we agree this is in no way a ready-to-go genre classification system, but a prototype for a more elaborated system based on statistical description of MIDI data, classifier ensembles with weighted voting schemes, and a windowing system to extract fragments from tracks and classify tracks by fragment voting.

ACKNOWLEDGEMENTS

We want to thanks the people at IMIRSEL for the great effort they have made to set up and run this big thing called MIREX 2005.

This work was supported by the projects Spanish CICyT TIC2003–08496–C04, partially supported by EU ERDF, and Generalitat Valenciana GV043–541.

References

A. Blum. Empirical support for winnow and weighted-majority based algorithms: Results on a calendar

OVERALL	
	accuracy
Mean hierarchical classification	37.76%
Mean raw classification	26.52%

38 classes taxonomy		
	acc.	std.
Hierarchical classification	24.84%	1.40
Raw classification	15.26%	1.13
Runtime	821 secs.	

9 classes taxonomy		
	acc.	std.
Hierarchical classification	50.67%	1.26
Raw classification	37.78%	2.30
Runtime	197 secs.	

Table 2: MIREX 2005 Symbolic genre classification. Evaluation results.

scheduling domain. *Machine Learning*, 26(1):5–23, 1997.

Leo Breiman. Random forests. *Machine Learning*, 45(1): 5–32, October 2001.

Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *Machine Learning: ECML 2000, 11th European Conference on Machine Learning*, volume 1810 of *Lecture Notes in Computer Science*, pages 109–116. 2000.

R. B. D'Agostino and M. A. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, Inc., New York, 1986.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons, 2000.

Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, 1999.

L. I. Kuncheva. That elusive diversity in classifier ensembles. In *Proc. 1st Iberian Conf. on Pattern Recognition and Image Analysis (IbPRIA'03)*, volume 2652 of *Lecture Notes in Computer Science*, pages 1126–1138. 2003.

Jeremy Pickens. A survey of feature selection techniques for music information retrieval. Technical report, Center for Intelligent Information Retrieval, Departament of Computer Science, University of Massachussetts, 2001.

P. J. Ponce de León and J. M. Iñesta. Feature-driven recognition of music styles. In *1st Iberian Conference on Pattern Recognition and Image Analysis. LNCS, 2652*, pages 773–781, 2003.