A Comparison of Statistical Approaches to Symbolic Genre Recognition

Carlos Pérez-Sancho, Pedro J. Ponce de León and José M. Iñesta
Departamento de Lenguajes y Sistemas Informáticos
{cperez,pierre,inesta}@dlsi.ua.es
Universidad de Alicante
P.O. box 99, E-03080 Alicante, Spain

Abstract

Previous work in genre recognition and characterization from symbolic sources (melodies extracted from MIDI files) carried out by our group pointed our research to study how the different utilized approaches perform and how their different abilities can be used together in order to improve both the accuracy and robustness of their decisions. Results for a corpus of Jazz and Classical music pieces are presented and discussed.

1 Introduction

Some recent works explore the capabilities of machine learning or pattern recognition methods to recognize music genre, either using audio (Zhu, Xue, and Lu 2004; Whitman, Flake, and Lawrence 2001), or symbolic sources (Cruz, Vidal, and Pérez-Cortes 2003; McKay and Fujinaga 2004). After a period of time researching on the use of statistical models and classification paradigms for music genre (or style) characterization from symbolic data (Ponce de León and Iñesta 2003; Pérez-Sancho, Iñesta, and Calera-Rubio 2004), we show here a comparison of the performance of two different paradigms, pointing to what they share and how they can complement their work.

MIDI files have been used as the primary source of musical data. This paper presents the data, the description methods, and the classification techniques in a comparative fashion.

2 Music data

The corpus used is a set of MIDI files from *Jazz* and *Classical* music collected from different sources, without any processing before entering the system, except for manually checking the presence and correctness of key, tempo, and meter meta-events, as well as the labeling of the (monophonic) melody track since we are interested in the part of music genre that may be conveyed by melody.

The training corpus is made up of 110 files, 45 of them being classical music and 65 of jazz, with a total length around 10,000 bars (more than six hours of music). The music pieces have been selected from well-known authors from both genres, ranging a broad range of styles (see Ponce de León and Iñesta (2003)). Also, a different test set of 42 files, 21 in each style, was used for validating the performance of different classifiers trained with the previous corpus, both individually and using an ensemble of classifiers.

Two different ways of describing the content of the melody track have been used. The first one is based on melodic, harmonic, and rhythmic statistical descriptors and the second one describes melodic content in terms of strings of symbols corresponding to melody subsequences. The first approach can be seen as a global content description, while the second one focus on local statistics of music content. Both description methods are briefly described in the following sections.

3 Statistical description models

For both approaches explained below, a sliding window of width ω bars traverses the melody, shifting its position one bar each time, and provides statistical information about the music content for each window position. For music with different tempi, this results in frames of different decision time horizons. However, as information is extracted at the symbolic level, rather than at the perception level, measuring window length in bars is more likely to capture structurally meaningful content within a frame.

This way, a new dataset is constructed for each ω. Integer values $\omega \in [1, 100]$ have been used, providing one hundred datasets of different size and granularity in order to analyze

the models' behaviour.

Previous research on varying the window shifting has been carried out, showing that small shifting values are prefered for melody description and its use in genre recognition, as they provide higher number of samples from each melody. Thus, a shifting value of one bar has been selected in this work, as it provides a sufficient number of samples per song in most cases.

3.1 Shallow statistical descriptors

The first description model that has been used is based on descriptive statistics that summarise the content of a melody in terms of pitches, intervals, durations, silences, harmonicity, rhythm, etc. This kind of statistical description of musical content is sometimes referred to as *shallow structure description*.

In this model, window content is described by a vector of statistical descriptors, labeled with the genre of the original melody. 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 analyzed and the other columns indicate the kind of statistics describing the property. A bullet in the table indicates which statistics are computed for each category.

Table 1: Shallow structure descriptors.

Category	Count	Range	Avg.-rel.	Dev.	Norm.
Notes	•				
Significant silences	•				
Non significant silences	•				
Pitches		•	•	•	•
Note durations		•	•	•	•
Silence durations		•	•	•	•
Inter-onset intervals		•	•	•	•
Pitch intervals		•	•	•	•
Non-diatonic notes	•		•	•	•
Syncopations	•				

Durations are measured in ticks. For pitch and interval categories, the range values are the difference between the maximum and the minimum, and average-relative descriptors are computed as the average value minus the minimum value. For durations (note durations, silence durations, and inter-onset intervals), the ranges are computed as the ratio between maximum and minimum values, and the average-relative descriptors are computed as the ratio between the average and the minimum value. Non-significant silences are those whose duration is less than a sixteenth note. Non-diatonic note statistics are computed considering key information encoded at the beginning of each MIDI file [1]. The syncopation counter is an estimate of the actual number of syncopes present in a melody. A note is considered a syncopation if it starts near the middle of a beat and extends at least near the middle of the following beat [2]. Finally, normality descriptors are computed using the D'Agostino (D'Agostino and Stephens 1986) statistic for assessing the distribution normality of the n values v_i in the window content for pitches, durations, intervals, etc. The statistic [3] is computed using Eq. 1:

$$D = \frac{\sum_i (i - \frac{n+1}{2}) v_i}{\sqrt{n^3 (\sum_i v_i^2 - \frac{1}{n}(\sum_i v_i)^2)}} \quad (1)$$

3.2 n-word based descriptors

The n-word based models make use of text categorization methods. The technique encodes note sequences as character strings, therefore converting a melody in a text to be categorized. Such a sequence of n consecutive notes is called a n-word. All possible n-words are extracted from a melody, except those containing a silence lasting four or more beats. The encoding for n-words used in this work has been derived from the method proposed in (Doraisamy and Rüger 2003). This method generates n-words by encoding pitch interval and duration information. For each n-note window, all possible intervals and duration ratios are obtained, respectively, by the equations:

$$I_i = Pitch_{i+1} - Pitch_i \qquad (i = 1, \ldots, n-1) \quad (2)$$

$$R_i = \frac{Onset_{i+2} - Onset_{i+1}}{Onset_{i+1} - Onset_i} \qquad (i = 1, \ldots, n-2) \quad (3)$$

and each n-word is defined as a string of symbols:

$$[\ I_1\ R_1\ \ldots\ I_{n-2}\ R_{n-2}\ I_{n-1}\ R_{n-1}\ ] \quad (4)$$

where the intervals and duration ratios have been mapped into alphanumeric characters (see Pérez-Sancho, Iñesta, and Calera-Rubio (2004) for details). In order to compute R_{n-1}, the duration of the last note in the n-word substitutes for the numerator in Eq. 3.

4 Classification techniques

4.1 Classifiers for shallow statistical features

Four conceptually different classification paradigms have been used with the description model presented in section 3.1:

[1] The presence of key metaevents has been verified by hand for all MIDI files used in this work.

[2] Here 'near' means a few ticks around the middle of the beat

[3] The D statistic is a small number typically between 0.25 and 0.3.

nearest-neighbour classifier (NN), bayesian classifier (Bayes), multilayer perceptron (MLP) and support vector machines (SVM) (Duda, Hart, and Stork 2000). These are standard machine learning techniques used for classification.

The multilayer perceptron and support vector machine implementation from the WEKA toolkit (Witten and Frank 2005) has been used. The NN and Bayes classifiers used are the author's own implementation. A summary of the parameters used for those classifiers is presented in Table 2.

Table 2: Classifier parameters for shallow statistical features.

Classifier	Parameters	
NN	euclidean distance	
Bayes	uniform priors	
MLP	learning coefficient:	0.25
	momentum:	0.12
	epochs:	500
	no. of hidden units:	$\frac{features+classes}{2}$
	normalized descriptors	
SVM	polinomial kernel degree:	2
	C:	1
	ϵ:	0.001
	normalized descriptors	

Thus, given a window of length ω (i.e. a dataset), four different classifiers have been trained. In order to estimate the accuracy of such classifiers, a 10-fold cross-validation scheme was used on each training dataset.

4.2 Naive Bayes Classifier for n-words

For n-word based classification the naive Bayes classifier (McCallum and Nigam 1998) has been used. Here, the classifier is based on the Bayes rule, but applying the *naive Bayes assumption*: all the n-words extracted from a melody sample are independent of each other, and also independent of the order they were generated. This assumption is clearly false, but naive Bayes can obtain near optimal classification errors in spite of that (Domingos and Pazzani 1997).

The class-conditional probability of a melody is given by the probability distribution of note sequences (n-words) for each genre, which can be learned from a labeled training set.

Two different distribution models have been used for the class-conditional probability: a Multivariate Bernoulli (MB) model, which just reflects the fact of words appearing or not in a melody, and a Multinomial (MN) model, which reflects the frequency of apparition of the words.

In the MB model, each class follows a multivariate Bernoulli distribution where the parameters to be learned from the training set are the class-conditional probabilities for the words in the vocabulary, while in the MN model, the probability that a melody has been generated from a genre is a multivariate multinomial distribution, where the melody length is assumed to be class-independent.

This method represents a musical piece as a vector $\mathbf{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{i|\mathcal{V}|}\right)$, where each component represents the presence of the word w_t in the melody, and $|\mathcal{V}|$ is the size of the vocabulary extracted from the dataset.

A common practice in text classification is to reduce the dimensionality of the vocabulary (usually very large) by selecting the words that contribute most to discriminate the class of a document (a melody here). This is useful to avoid overfitting to the training data when there are limited data samples and a large number of features, and also to increase the speed of the system. The *average mutual information* (AMI) (Cover and Thomas 1991) has been used in this work to rank the words. This method gives a high value to those words that appear often in the melodies of one genre and are seldom found in the melodies of the other genres. The n-words extracted from the training set are ranked using this value, and only information about the most informative words are provided to the classifier, thus limiting the size of the vocabulary ($|\mathcal{V}|$). This is a parameter that must be set at training time, and several values were tested as will be explained later in Section 5.

4.3 Classifier ensembles

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 and Whitaker 2003) as an argument for using classifier ensembles with good results. These ensembles could be regarded as committees of 'experts' in which the decisions of individual classifiers are considered as opinions supported by a measure of confidence usually related to the accuracy of that particular classifier. The final classification is taken either by majority vote or by a weighting system.

4.3.1 Voting schemes

In this paper, two different possibilities that are presented below have been proposed and compared. In the discussion that follows, N stands for the number of samples contained in the training set $\mathcal{X} = \{\mathbf{x}\}_{i=1}^{N}$, M is the number of classes in a set $\mathcal{C} = \{c_j\}_{j=1}^{M}$, and K classifiers, C_k, are utilized.

1. Best-worst weighted majority. In this ensemble, the best and worst classifiers in the ensemble are identified using their estimated accuracy. A maximum authority, $a_k = 1$,

is assigned to the former and a null one, $a_k = 0$, to the latter, being equivalent to remove this classifier from the ensemble. The rest of classifiers are rated linearly between these extremes. The values for a_k are calculated as follows:

$$a_k = 1 - \frac{e_k - e_B}{e_W - e_B} \quad , \tag{5}$$

where

$$e_B = \min_k\{e_k\} \quad \text{and} \quad e_W = \max_k\{e_k\} \tag{6}$$

and e_k is the number of errors made by C_k.

2. Quadratic best-worst weighted majority. In order to give more authority to the opinions given by the most accurate classifiers, the values obtained by the former approach are squared. This way,

$$a_k = (\frac{e_W - e_k}{e_W - e_B})^2 \quad . \tag{7}$$

4.3.2 Classification

Once the weights for each classifier decision have been computed, the class receiving the highest score in the votation is the final class prediction. If $\hat{c}_k(\mathbf{x}_i)$ is the prediction of C_k for the sample $\mathbf{x}_i$, then the prediction of the ensemble can be computed as

$$\hat{c}(\mathbf{x_i}) = \arg\max_{c_j \in \mathcal{C}} \sum_k w_k \delta(\hat{c}_k(\mathbf{x}_i), c_j) \quad , \tag{8}$$

being $\delta(a, b) = 1$ if $a = b$ and 0 otherwise, and w_k the normalized authority a_k of each classifier.

5 Results

The classifiers described in section 4 have been applied to the training datasets with different parameter values both for the feature extraction and classifier tuning. A study of their performance as a function of the window length is presented in Figure 1. In both plots the estimated accuracies for classification approaches based on the same representation scheme are presented together in the same plot.

For the Naive Bayes classifier, different values of vocabulary size were tested, ranging from 10 n-words to the whole vocabulary. The results presented here were computed averaging over the size ranges where best results can be obtained, with a standard deviation of 1.5% in accuracy.

The estimated accuracy for the shallow description based classifiers is shown in Figure 1 (top). The classifiers perform comparably for medium to large window lengths, except for the bayesian classifier showing poorer performance. Multilayer perceptron performance also degrades slightly for large windows. This is probably due to overfitting the training data, since less data are avalaible for larger window lengths. The best results are obtained by NN with $\omega \in [55, 60]$, giving an estimated accuracy about 95%.

In the case of n-word based classifiers, a similar behaviour can be observed. Their performance tends to improve as the window length increases, obtaining their best results using short n-words and a multivariate Bernoulli distribution.

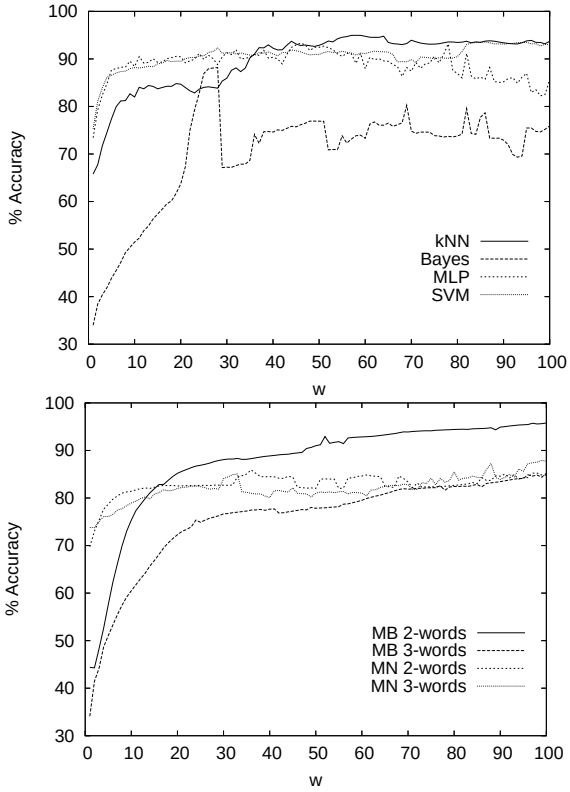

Figure 1: Performance of the classifiers for the shallow (top) and n-word (bottom) approaches.

5.1 Classifier ensemble results

In order to evaluate how well these different statistical approaches combine, two ensembles have been constructed using the votation methods described above (represented as V1 and V2). One classifier per classification technique was trained on training sets constructed using window lengths of 30, 60 and 90 bars, considered representative of short, average and large lengths. This produced a total of 24 different classifiers, whose decisions on a test set can be combined in a voting ensemble.

A summary of results from single classifiers is presented in Table 3. The third and fourth column in the table are the authority values a_k for voting methods V1 and V2, respectively. The last two columns are the number of errors made by the classifiers and their accuracy on the test set. This values are computed on a per-song basis. For each test song, several samples are extracted using the sliding window method and classified into a genre. Then the song is assigned to the most voted genre. The best results are shown in boldface.

A brief summary of the decisions of the ensembles on the test set is displayed in Table 4. These results show the number of errors made by the ensembles and their accuracy on a per-song basis. Note that the ensemble's performance using the quadratic best-worst strategy slightly improves the behaviour of the best of the individual classifiers. Although this is not a significant result, it shows how the ensembles performed comparatively well to the best single classifier used. When no single classifier is known to be best suited for a particular task, classifier ensembles can help obtaining reasonable good results with no-so-good classifiers. Also it shows how a multiresolution statistical analisys capturing both local and global properties of a melody can be used to extract certain information from it.

Table 3: Classifiers used in the ensembles. From left to right: classification technique, window length, authority for both voting methods and number of errors and accuracy with the test set.

Classifier	ω	a_k (V1)	a_k (V2)	# errors	% accu.
Bayes	30	0	0	8	81.0
	60	0.22	0.05	7	83.3
	90	0.21	0.04	6	85.7
NN	30	0.68	0.46	3	92.9
	60	**1**	**1**	**2**	**95.2**
	90	0.95	0.90	3	92.9
MLP	**30**	**0.87**	**0.76**	**2**	**95.2**
	60	0.75	0.57	3	92.7
	90	0.65	0.43	3	92.7
SVM	30	0.87	0.76	3	92.7
	60	**0.87**	**0.75**	**2**	**95.2**
	90	0.94	0.88	4	90.5
MB ($n = 2$)	30	0.76	0.58	8	81.0
	60	0.86	0.74	6	85.7
	90	0.89	0.80	6	85.7
MB ($n = 3$)	30	0.33	0.11	13	69.0
	60	0.40	0.16	12	71.4
	90	0.33	0.11	12	71.4
MN ($n = 2$)	30	0.56	0.32	4	90.5
	60	0.53	0.28	3	92.9
	90	0.30	0.09	3	92.9
MN ($n = 3$)	30	0.53	0.28	9	78.6
	60	0.63	0.40	11	73.8
	90	0.63	0.40	11	73.8

Table 4: Ensemble's performance.

Voting method	# errors	% accu.
V1	2	95.2
V2	1	97.6

6 Conclusions

Two different statistical approaches to melody description and classification into a set of genres have been presented in this paper. The first one is based on global (or shallow) statistical descriptors and the second one on local (n-word based) statistical descriptors. Furthermore, the classifiers have been tested based on varying the length of the melodic fragment analized. Also, the performance of ensembles of those classifiers for classifying a symbolically represented melody into a given music genre has been shown. One ensemble slightly improved the performance of the best single classifier used. In previous works we have shown the feasibility of using these kind of data and representations to approach this problem, but by constructing an ensemble using different classifiers, their votes are "averaged" and this reduces the risk of choosing the wrong classifier.

Further work is needed to test the robustness of the capabilities of these approaches to classify larger datasets and extend the problem to more music genres.

7 Acknowledgments

This work was supported by the projects Spanish CICyT TIC2003–08496–C04, partially supported by EU ERDF.

References

Cover, T. M. and J. A. Thomas (1991). *Elements of Information Theory*. John Wiley.

Cruz, P. P., E. Vidal, and J. C. Pérez-Cortes (2003). Musical style identification using grammatical inference: The encoding problem. *LNCS 2905*, 375–382.

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

Domingos, P. and M. Pazzani (1997). Beyond independence: conditions for the optimality of simple bayesian classifier. *Machine Learning 29*, 103–130.

Doraisamy, S. and S. Rüger (2003). Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems 21*(1), 53–70.

Duda, R. O., P. E. Hart, and D. G. Stork (2000). *Pattern Classifi-*

cation. John Wiley and Sons.

Kuncheva, L. I. and C. J. Whitaker (2003). Measures of diversity in classifier ensembles. *Machine Learning 51*, 181–207.

McCallum, A. and K. Nigam (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48.

McKay, C. and I. Fujinaga (2004). Automatic genre classification using large high-level musical feature sets. In *Int. Conf. on Music Information Retrieval, ISMIR 2004*, pp. 525–530.

Pérez-Sancho, C., J. M. Iñesta, and J. Calera-Rubio (2004). Style recognition through statistical event models. In *Proceedings of the Sound and Music Computing Conference, SMC '04*.

Ponce de León, P. J. and J. M. Iñesta (2003). Feature-driven recognition of music styles. *LNCS 2652*, 773–781.

Whitman, B., G. Flake, and S. Lawrence (2001). Artist detection in music with minnowmatch. In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, pp. 559–568.

Witten, I. H. and E. Frank (2005). *Data Mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.

Zhu, J., X. Xue, and H. Lu (2004). Musical genre classification by instrumental features. In *Int. Computer Music Conference, ICMC 2004*, pp. 580–583.