STYLE RECOGNITION THROUGH STATISTICAL EVENT MODELS

Carlos Pérez-Sancho, José M. Iñesta, and Jorge Calera-Rubio
Dept. Lenguajes y Sistemas Informáticos, Universidad de Alicante, Spain
{cperez,inesta,calera}@dlsi.ua.es

ABSTRACT

The automatic classification of music fragments into styles is one challenging problem within the music information retrieval (MIR) domain and also for the understanding of music style perception. This has a number of applications, including the indexation and exploration of music databases. Some technologies employed in text classification can be applied to this problem. The key point here is to establish something in music equivalent to the words in texts. A number of works use the combination of intervals and duration ratios for this purpose. In this paper, different statistical text recognition algorithms are applied to style recognition using this kind of melody representation, exploring and comparing their performance for different word sizes.

1. INTRODUCTION

The automatic machine learning and pattern recognition techniques, successfully employed in other fields, can be also applied to music analysis. One of the tasks that can be posed is the modelization of the music style. Immediate applications are the classification, indexation, and content-based search in digital music libraries, where digitised (MP3), sequenced (MIDI) or structurally represented (XML) music can be found. For example, the computer could be trained in the user musical taste in order to look for that kind of music over large musical databases.

A number of recent papers explore the capabilities of machine learning methods to recognise music style, either using audio or symbolic sources. Among the first, for example, Pampalk et al. [8] use self-organising maps (SOM) to pose the problem of organising music digital libraries according to sound features of musical themes, in such a way that similar themes are clustered, performing a content-based classification of the sounds. Whitman et al. [11] present a system based on neural networks and support vector machines able to classify an audio fragment into a given list of sources or artists. Also Soltau et al. [10] describe a neural system to recognise music types from sound inputs.

Dealing with symbolic data, we can find a recent work by Cruz et al. [4], where the authors show the ability of grammatical inference methods for modeling musical style. A stochastic grammar for each musical style is inferred from examples, and those grammars are used to parse and classify new melodies. In [9] the authors compare the performance of different pattern recognition paradigms to recognise music style using descriptive statistics of pitches, intervals, durations, silences, etc. Other approaches like hidden Markov models [2] or multi layer feed forward neural networks [1] have been used to pose this problem.

Our aim is to explore the capabilities of text categorization algorithms to solve problems relevant to computer music. In this paper, some of those methods are applied to the recognition of musical genres from a symbolic representation of melodies. Jazz and classical music styles have been chosen as an initial benchmark for the proposed methodology due to the general agreement in the musicology community about their definition and limits.

2. METHODOLOGY

2.1. Data set

Experiments in section 3 were performed using a corpus of MIDI files collected from different web sources, without any processing. It is a quite heterogeneus corpus, not specifically created to test our system. The melodies are real-time sequenced by musicians, without quantization. The corpus is made up of a total of 110 MIDI files, 45 of them being classical music and 65 being jazz music. Each MIDI file contains a monophonic sequence written in the 4/4 meter. The length of the corpus is around 10,000 bars (40,000 beats). Classical melody samples were taken from works by Mozart, Bach, Schubert, Chopin, Grieg, Vivaldi, Schumann, Brahms, Beethoven, Dvorak, Haendel, Paganini and Mendelssohn. Jazz music samples were standard tunes from a variety of well known jazz authors including Charlie Parker, Duke Ellington, Bill Evans, Miles Davis, etc.

2.2. Encoding

Since we are trying to use text categorization approaches, there is a need to find an appropriate encoding, something like *music words*, that captures relevant information of the data and is suitable for that kind of algorithms to be applied.

One possible encoding is the one proposed by Doraisamy and Rüger [6], that make use of pitch intervals and inter onset time ratios (IOR) to build series of symbols of a given length. We will name these series n-words (we will use also just “words” for short in this document), due to the analogy to text we are trying to establish. A se-

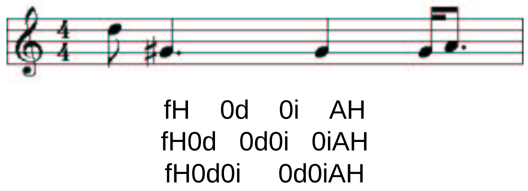

Figure 1. An example of short melody and the coding of all the possible 2- (top), 3- (middle) and 4-words (bottom) in it.

quence of n notes generates $n-1$ pitch intervals and $n-1$ IOR[1] that are represented together as a word with $2n-2$ symbols. Note that all the pitches and durations contained in the n-word are represented in a relative way (intervals and ratios) with respect to the pitch and duration of the first note, giving more generality to the coded information, which is reported to be useful for music classification [4, 2].

Using this encoding, all the music words of order n are extracted from the melody track of each MIDI file. If the sequence has N notes, $N-n+1$ is the number of n-words can be extracted from it. Thus, each melody in our database is transformed into a sequence of words in the form

$$\{I_iR_iI_{i+1}R_{i+1}\dots I_{i+n-1}R_{i+n-1}\}_{i=1}^{N-n+1}$$

where i represents the position of the i^{th} note in the sequence. See Fig. 1 for an example of the coding. Using this scheme, a melody can be considered as a set of musical n-words in the same way that a text document is considered for classification as a set of words.

For obtaining the codes, a non-linear mapping from numerical values to letters is applied. Intervals are mapped into a set of 53 letters, where '0' represents the unison, and the IOR into a set of 21 letters, where 'Z' represents the IOR = 1. This is also useful to quantize the MIDI sequence and also to impose limits to the permitted ranges for intervals and IOR (see [6] for details).

In order to illustrate the distribution of codes for both styles, histograms of intervals and IORs are displayed in Fig. 2. Note the different frequencies for each style, that are in the basis of the recognition system.

Also, *stop words* are used to segmentate the melody into musical phrases. For that, a simple criterion has been considered: when a silence equal or longer than a whole note is found, no word is coded across it. This implies that $n-1$ n-words less than the amount given above are extracted for each time that a long silence is found.

2.3. Word lengths

In order to test the classification ability of different word lengths, n, a range for $n \in \{2,3,4\}$ has been established.

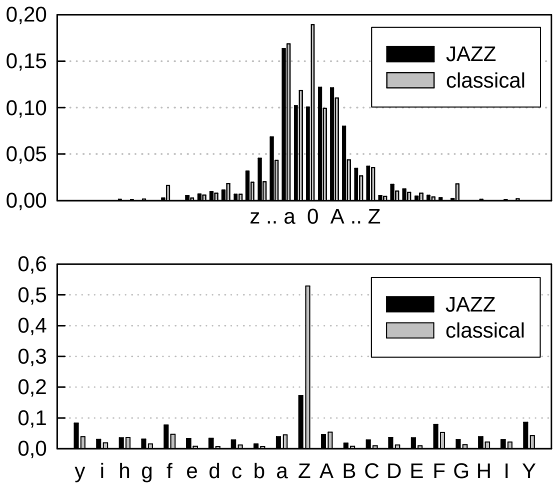

Figure 2. Histograms: (top) normalized frequencies of intervals in the training set; (bottom) frequencies of inter onset ratios. In the abcises, the coding letters are represented.

n	Jazz	Classical	Total	$\%\lvert\mathcal{V}\rvert$
2	425	485	548	49.2
3	4883	3840	7903	0.638
4	6481	6209	12501	$9.07 \cdot 10^{-4}$

Table 1. Number of words in the training sets for the different word lengths: number of different words in each style, total of different words in the corpus, and percentage on the vocabulary size.

The shorter n-words are less specific and provide more general information and, on the other hand larger n-words are maybe more informative but the models based on them will be more difficult to train. The vocabulary of each length has a size $|\mathcal{M}_n| = (53 \times 21)^{n-1}$ words.

In Table 1 the number of words that have been extracted from the training set for each length is displayed. From left to right: the total number of different words found, their percentages on the vocabulary size, and the number of different words for jazz and classical music are displayed.

2.4. Naive Bayes Classifier

The naive Bayes classifier, as described in [7], has been used. In this framework, classification is performed following the well-known *Bayes' classification rule*. In a context where we have a set of classes $\mathcal{C} = \{c_1, c_2, \dots, c_{|\mathcal{C}|}\}$, a melody x_i is assigned to the class c_j with maximum a posteriori probability (MAP), in order to minimize the probability of error:

[1] The last IOR is computed using the duration of the last note while the others use the time between note onsets. This permits to give more information to a 2-word than just one interval.

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

where $P(c_j)$ is the a priori probability of class c_j, $P(x_i|c_j)$ is the probability of x_i being generated by class c_j, and $P(x_i) = \sum_{j=1}^{|\mathcal{C}|} P(c_j)P(x_i|c_j)$. Since $P(x_i)$ is just a normalization factor to ensure that $\sum_{j=1}^{|\mathcal{C}|} P(c_j|x_i) = 1$ we can just ignore it and assign x_i to the class which maximizes $P(c_j)P(x_i|c_j)$.

Our classifier is based on the *naive Bayes assumption*, i.e. it assumes that all words in a melody are independent of each other, and also independent of the order they are generated. This assumption is clearly false in our problem and also in the case of text classification, but naive Bayes can obtain near optimal classification errors in spite of that [5]. To reflect this independence assumption, melodies can be represented as a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{i|\mathcal{V}|})$, where each component $x_{it} \in \{0,1\}$ represents whether the word w_t appears in the document or not, and $|\mathcal{V}|$ is the size of the vocabulary. Thus, the class-conditional probability of a document $P(x_i|c_j)$ is given by the probability distribution of words w_t in class c_j, which can be learned from a labelled training sample using a supervised learning method.

2.4.1. *Multivariate Bernoulli model*

In this model, melodies are represented by a binary vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{i|\mathcal{V}|})$, where each $x_{it} \in \{0,1\}$ represents whether the word w_t appears at least once in the melody. Using this approach, each class follows a multivariate Bernoulli distribution:

$$P(x_i|c_j) = \prod_{t=1}^{|\mathcal{V}|} x_{it}P(w_t|c_j) + (1 - x_{it})(1 - P(w_t|c_j)) \quad (2)$$

where $P(w_t|c_j)$ are the class-conditional probabilities of each word in the vocabulary, and these are the parameters to be learned from the training sample.

Given a labelled sample of melodies $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, Bayes-optimal estimates for probabilities $P(w_t|c_j)$ can be easily calculated by counting the number of occurrences of each word in the corresponding class:

$$P(w_t|c_j) = \frac{1 + N_{tj}}{2 + N_j} \quad (3)$$

where N_{tj} is the number of melodies in class c_j containing word w_t, and N_j is the total number of melodies in class c_j. Also, a Laplacean prior has been introduced in the equation above to avoid probabilities of 0 or 1. Prior probabilities for classes $P(c_j)$ can be estimated from the training sample using a maximum likelihood estimate:

$$P(c_j) = \frac{N_j}{|\mathcal{X}|} \quad (4)$$

Classification of new melodies is performed then using Eq. 1, which is expanded using Eqs. 2 and 4.

2.4.2. *Multinomial model*

This model takes into account word frequencies in each melody, rather than just the occurrence or non-occurence of words as in the multivariate Bernoulli model. In consequence, documents are represented by a vector, where each component x_{it} is the number of occurrences of word w_t in the melody. In this model, the probability that a melody has been generated by a class c_j is the multinomial distribution, assuming that the melody length in words, $|x_i|$, is class-independent [7]:

$$P(x_i|c_j) = P(|x_i|)|x_i|! \prod_{t=1}^{|\mathcal{V}|} \frac{P(w_t|c_j)^{x_{it}}}{x_{it}!} \quad (5)$$

In this case, Bayes-optimal estimates for class-conditional word probabilities are:

$$P(w_t|c_j) = \frac{1 + N_{tj}}{|\mathcal{V}| + \sum_{k=1}^{|\mathcal{V}|} N_{kj}} \quad (6)$$

where N_{tj} is the sum of occurrences of word w_t in melodies in class c_j. Class prior probabilities are also calculated using Eq. 4.

2.5. Feature selection

The methods explained above use a representation of musical pieces as a vector of symbols. A common practice in text classification is to reduce the dimensionality of those vectors by selecting the words which contribute most to discriminate the class of a document. A widely used measure to rank the words is the *average mutual information* (AMI) [3].

For the multivariate Bernoulli model, the AMI is calculated between (1) the class of a document and (2) the absence or presence of a word in the document. We define C as a random variable over all classes, and F_t as a random variable over the absence or presence of word w_t in a melody, F_t taking on values in $f_t \in \{0,1\}$, where $f_t = 0$ indicates the absence of word w_t and $f_t = 1$ indicates the presence of word w_t. The AMI is calculated for each w_t as[2]:

$$I(C;F_t) = \sum_{j=1}^{|C|} \sum_{f_t \in \{0,1\}} P(c_j, f_t) \log \frac{P(c_j, f_t)}{P(c_j)P(f_t)} \quad (7)$$

where $P(c_j)$ is the number of melodies for class c_j divided by the total number of melodies; $P(f_t)$ is the number of melodies containing the word w_t divided by the total number of melodies; and $P(c_j, f_t)$ is the number of melodies in class c_j having a value f_t for word w_t divided by the total number of melodies.

In the case of the multinomial model, the AMI is calculated between (1) the class of the melody from which a word occurrence is drawn and (2) a random variable over all the word occurrences, instead of melodies. In this case,

[2] The convention $0 \log 0 = 0$ was used, since $x \log x \to 0$ as $x \to 0$.

Eq. 7 is also used, but $P(c_j)$ is the number of word occurrences appearing in melodies in class c_j divided by the total number of word occurrences, $P(f_t)$ is the number of occurrences of the word w_t divided by the total number of word occurrences, and $P(c_j, f_t)$ is the number of occurrences of word w_t in melodies with class label c_j, divided by the total number of word occurrences.

3. RESULTS

The style recognition ability of the different word sizes has been tested. For each model, the naive Bayes classifier has been applied to the words extracted from the melodies in our training set. The experiments have been made following a leave-one-out scheme: the training has been constructed with all the melodies but one, kept for test. After training the model, the words in the test melody are extracted and used to classify it. The presented results are the percentage of successfully classified melodies.

The evolution of the classification as a function of the significance of the used information is presented in the graphs in figure 3. For this, the words in the training set have been ordered according to two different criteria: (1) their frequencies in the training set, and (2) their AMI value. After that, experiments using only the best situated words ($|\mathcal{V}|$ in the graphs) have been performed.

Note that the results were not conclusive in terms of different statistical distributions or word order, since all the methods performed comparatively. There is a tendency of the Bernoullis to classify better for small values of $|\mathcal{V}|$ while multinomials seem to provide better results for larger $|\mathcal{V}|$.

Table 2 shows the best results obtained in the experiments. The best accuracy was obtained for the word size $n = 2$, reaching a 93.25% of successful style identification. Large n-words only perform well (above 80%) for very small $|\mathcal{V}|$ values, and get worse rapidly for larger values. This preference for little specific information points to the fact that the method is indeed able to classify but maybe the training set is small, and the results can be improved for larger models with more training melodies.

Also the values for precision and recall have been studied. Note that the recall figures get very low as n increases, being the cause of the lower classification rates obtained for large words. In fact, the tendency of lengths $n = 2, 3$ is to get low percentage rates when $|\mathcal{V}|$ increases that are due to low recall and very high precision values: there are a lot of unclassified melodies, but the decisions taken by the classifier are usually very precise. It can be said that the classifier learns very well but little. This fact also reflects the need of a larger training set.

Finally, we have compared our results to those obtained by our research group with the same training set, but using melodic, harmonic and rhythmic statistical descriptors. They are fed into well-known supervised classifiers like, k-nearest neighbours (k-NN) or a standard Bayes rule (see [9] for details). In those experiments, the best recognition rates obtained when extracting the descriptors

n	Best classification %	$\|\mathcal{V}\|$	Jazz		Classical	
			Prec.	Recall	Prec.	Recall
2	93.25	300	94	95	93	91
3	86.78	50	79	57	97	71
4	90.62	20	50	9	100	58

Table 2. Best results in classification percentages obtained in the experiments. For each word length value, n, the table shows, from left to right: best classification, size of vocabulary used for it, and precision and recall figures for both styles, also in percentage.

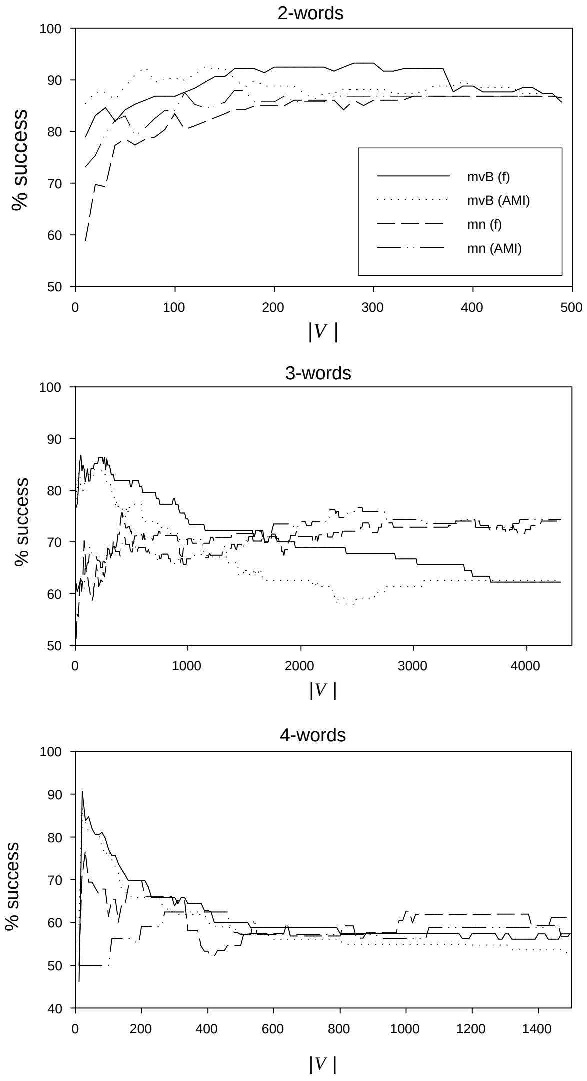

Figure 3. Evolution of style recognition percentage in average for both classes and different word sizes. The four plots in each graph represent: **mvB** multi-variate Bernoulli, **mn** multinomial, **(f)** words sorted by frequencies, **(AMI)** words sorted by AMI.

from the whole melody were 91.0% for Bayes and 93.0% for k-NN, after a long study of the parameter space and the descriptor selection procedures. Thus, the first results obtained under this new approach are very encouraging.

4. CONCLUSIONS

In this paper, the feasibility of using text classification technologies for music style recognition has been tested. The first results of our research in this particular application have been presented and discussed. The models based on 2-words had the best performance, reaching a 93.25% of successful style recognition. Larger word lengths have provided also good results using small vocabulary sizes. In these cases, the precision of the classifiers are good or even perfect but the recall figures are very low, due to a lot of unclassified melodies. This fact points to a lack of training data. It is very likely that longer words would improve their performance with larger corpora. The various statistical distributions tested did not present significant differences in classification.

The results have been compared to those obtained by other description and classification techniques, providing similar or even better results. We are convinced that an increment of the data available for training will improve the results clearly, specially for larger n-word sizes, were the method has proved to be very accurate, but lacks retrieval power.

In the further work, more data and styles will be included in our experimental framework and other classifiers, based on the symbolic representation of music, will be investigated.

Acknowledgment

This work was supported by the Spanish CICYT project TIRIG, code TIC2003–08496–C04.

5. REFERENCES

[1] G. Buzzanca. A supervised learning approach to musical style recognition. In *Music and Artificial Intelligence. Additional Proceedings of the Second International Conference, ICMAI 2002*, Edinburgh, Scotland, 2002.

[2] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proc. of the Int. Conf. on Artificial Intelligence*, Las Vegas, USA, 2001.

[3] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[4] P. P. Cruz, E. Vidal, and J. C. Pérez-Cortes. Musical style identification using grammatical inference: The encoding problem. In A. Sanfeliu and J. Ruiz-Shulcloper, editors, *Proc. of CIARP 2003*, pages 375–382, 2003.

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

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

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

[8] E. Pampalk, S. Dixon, and G. Widmer. Exploring music collections by browsing different views. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, pages 201–208, Baltimore, USA, 2003.

[9] 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. Lecture Notes in Computer Science, 2652*, pages 773–781, Majorca, Spain, 2003.

[10] H. Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-1998)*. Seattle, Washington, May 1998.

[11] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568. Falmouth, Massachusetts, September 10–12 2001.