Connection Science
Vol. 00, No. 00, January 2008, 1–13

RESEARCH ARTICLE

Genre classification using chords and stochastic language models

Carlos Pérez-Sancho, David Rizo and José M. Iñesta*

Pattern Recognition and Artificial Intelligence Group
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante, Ap. 99, E-03080 Alicante, Spain

(*21 February 2008*)

Music genre meta-data is of paramount importance for the organization of music repositories. People use genre in a natural way when entering a music store or looking into music collections. Automatic genre classification has become a popular topic in music information retrieval research both with digital audio and symbolic data. This work focuses on the symbolic approach, bringing to music cognition some technologies, like the stochastic language models, already successfully applied to text categorization. The representation chosen here is to model chord progressions as n-grams and strings and then apply perplexity and Naive Bayes classifiers in order to model how often those structures are found in the target genres. Some genres and sub-genres among popular, jazz, and academic music have been considered and the results at different levels of the genre hierarchy for the techniques employed are presented and discussed.

Keywords: genre classification; statistical text classification; chord progressions

1. Introduction

Organization of large music repositories is a tedious and time-intensive task for which music genre is an important meta-data. Automatic genre and style classification have become popular topics in Music Information Retrieval (MIR) research because musical genres are categorical labels created by humans to characterize pieces of music and this nature provides the genre meta-data with a high semantic and cultural information to the music items in the collection.

Traditionally, the research domain of genre classification has been divided into the audio (Tzanetakis and Cook 2002) and symbolic (McKay and Fujinaga 2004) music analysis and retrieval domains. Nevertheless, some authors have paid attention recently on making use of the best of both worlds (Lidy et al. 2007, Cataltepe et al. 2007). The work by Lidy et al. (2007) deals with audio to MIDI transcription in order to extract features from both signals and then combine the decisions of the different classifiers. On the other hand, Cataltepe and coworkers' approach (2007) is just the opposite: to synthesize audio from MIDI and then analyze both signals to integrate the classifications.

This work focuses on the symbolic approach, but keeping an eye on what can be obtained from audio data, in such a way that it can work either with symbolic

*Corresponding author. Email: inesta@dlsi.ua.es

ISSN: 0954-0091 print/ISSN 1360-0494 online

DOI: 10.1080/09540090xxxxxxxxxxxx
http://www.informaworld.com

data sources or become a back-end of an audio preprocessing and chord extraction stage.

In the book on Harmony by Walter Piston (1987) (chapter 29), the evolution of the harmony practice from the early Baroque period to the 20th century music is depicted. Some rules that were almost forbidden in a period have been accepted afterwards. Those rules include contrapunctual rules, musical form, and harmony issues like tonality modulations and valid chord progressions. Furthermore, it is well known that pop-rock tunes mainly follow the classical tonic-subdominant-dominant chord sequence, whereas jazz harmony books propose also different series chord progressions as a standard (Herrera 1998). Therefore, if in each musical period or genre there was a valid harmonic framework to compose. The works created by that time should move through those chord sequences, or at least, should not use progressions that were not valid. This is the underlying hypothesis in the proposed approach.

Our goal is bringing to music cognition some technologies, like the stochastic language models, that have been already applied successfully to text categorization. Some of those technologies, at least those working at a lexical level can be transfered to music with some adaptation work.

1.1. *Previous work*

Little attention has been paid in the genre recognition literature to how harmony can help in this task. For example, in (Tzanetakis et al. 2002) the authors use pitch histograms as feature vectors (either computed from audio signals or directly derived from MIDI data). The histograms can be folded into one single octave, which yields a chroma feature representation describing the harmonic content of the music. The histograms are compared and some patterns are found which reveal genre specific information.

In contrast to that low-level harmony features, the representation chosen in this paper is to model higher level harmonic structures, like chord progressions, as n-grams and strings in order to compute how often are those structures are found in the target genres. There are a small number of papers in the literature that poses this problem.

In (Shan et al. 2002), a rule-based system is used to classify sequences of chords belonging to three categories: Enya, Beatles and Chinese folk songs. The data set used was made up of MIDI files, and chords were derived from their melody using a set of heuristic rules. A vocabulary of 60 different chords was used, including triads and 7th 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, thus leading the system to a bigger confusion.

Paiement et al. (2005) also used chord progressions to build probabilistic models. In this work a set of 52 jazz standards was used, encoded as sequences of 4-note chords, each chord with a duration of 2 beats in a 4 beat meter. 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 suggest that chord structures are a suitable source of information to represent musical genres. These models should be tested in a more general framework with more genres in order to assess that they properly characterize different styles.

As far as the authors know, there is no previous work dealing with complete

harmonic information (i.e. complete chord names with ornaments), so the goal of this paper is to test whether this information can be represented by a model that learns probabilities from the chord progressions used in each genre. To test these models, we have built a classification framework with different genres covering a wide range of the music domain, in order to see whether such a model built from a genre is able to correctly identify new chord sequences belonging to that genre.

1.2. *Audio to chord transcription systems*

Harmonic information, in spite of being a useful description of music, is a kind of information difficult to find. Although it is easy to retrieve loads of chord sequences from the Internet for many different genres, they are not usually very reliable, if we take into account that labelling chords in songs by ear is a difficult task even for trained people.

There are alternative methods to obtain chord sequences from musical information, either from symbolic melody representations, or from digital audio signals. In the first case, the Melisma Music Analyzer[1] is a choice for obtaining a harmonic analysis from a sequence of notes, including tonality and chord names. Although good results have been reported in chord recognition for some kinds of music (Lee and Slaney 2006), it is an error-prone process, but it can be taken into consideration.

Anyway, if we plan to integrate our models in a general MIR system, a more realistic scenario would be a system working with digital audio, from which many different descriptors can be extracted. In this line we can find several chord transcription systems that, given an audio signal, are able to extract the corresponding sequence of chords that are present at each time instant, either at frame or beat level. These works have reported good results using HMMs with chromagrams extracted from real recordings of the Beatles (Sheh and Ellis 2003, Bello and Pickens 2005), with chord recognition rates around 75%. Using similar techniques and characteristics, Lee and Slaney (2006) obtained a 92% chord recognition rate with classical music, but using the Melisma Music Analyzer to build the ground truth instead of using hand labelled samples. Another common feature between these works is the fact that they only use small subsets of chord names, with a vocabulary size ranging from 24 (major and minor triads) to 36 (adding diminished triads). Although in (Sheh and Ellis 2003) the original vocabulary comprised 147 possibilities, including triads and seventh chords, only 32 from the original 147 were found in the dataset.

So, it seems clear that to be able to incorporate a chord transcription system to obtain the chord sequences from real audio recordings, and then build some kind of generalization model on them, we must be ready to work with a simpler set of chords, including only triads. The experiments explained in section 4 were performed using two feature sets, one including full chord names with extensions and the other one using only triads, in order to test the loss of generalization power when using fewer information.

2. Experimental data

Music from three "domains" has been utilized: popular, jazz, and academic music. The popular music data available have been separated into two sub-genres: *pop-*

[1] `http://www.link.cs.cmu.edu/music-analysis/`

Table 1. Number of files per genre and subgenre.

Academic		Jazz		Popular	
Baroque	61	Pre-bop	195	Country	64
Classical	49	Bop	105	Pop-rock	85
Romanticism	159	Bossanova	43		
Total: 269		**Total: 343**		**Total: 149**	

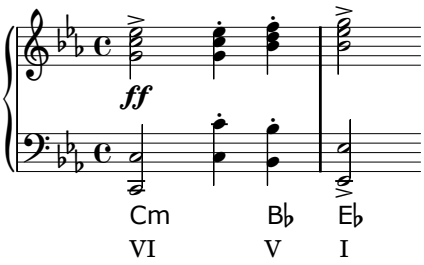

Figure 1. Fragment of "Der Freischtz" by Weber, taken from (Piston 1987).

rock and *country* (relative to the american modern folk). For jazz, three styles have been established: swing, early, and Broadway tunes grouped in a *pre-bop* class, plus *bossanova*, and *bop* standards. Finally, the academic music has been categorized according to the historic periods: *baroque*, *classicism*, and *romanticism*. All these categories have been defined with the help and advise of music experts that have also collaborated in the task of assigning meta-data tags to the files and rejecting outliers in order to have a reliable ground truth for the experiments.

The number of files eventually considered for each genre is displayed in Table 1. The total amount of pieces was 761, providing around 50 hours of music data.

The corpus of song chords has been obtained from files encoded in the format of the PG Music software named Band in a Box (aka BIAB)[1], and then converted into MMA format[2] using the *biabconverter* program that can be found in the MMA webpage. The classical corpus was extracted from the *classfake* folders in the BIAB installation. The rest of the files were obtained from links found at the Internet[3], being the chords publicly available. The utilized files can be obtained upon request to the authors.

In order to focus on the progression of the chords instead of the chords themselves, before exporting to the MMA format, all the songs have been transposed to C Major / A minor using the BIAB automatic transpose option.

The chords in a progression have different tonal functions depending on the underlying tonality, and therefore, a different meaning. While using just the name, Am is always the same chord, using the degree we are able to distinguish whether it is the first degree in the Am key or the sixth in the C major key. For example, the sequence Cm – B♭ – E♭ in Fig. 1 should be read as I – VII (working as V of III) – III for the C minor tonality but as VI – V – I for E♭ major. This second approach to represent chords (using their degrees instead of their names) has been also considered. This way, we are also modeling chord degree progressions. The price the system has to pay is the need of having the tonality as an input. We will study the performance of both models comparatively.

Also, in order to test the feasibility of using state-of-the-art audio to chord transcription systems as a source of information, as those mentioned in section 1.2, two more feature sets have been generated by removing chord extensions to the previously explained representations, thus reducing every chord name/degree to its

[1] http://www.pgmusic.com
[2] http://www.mellowood.ca/mma/
[3] http://www.alisdair.com/gearsoftware/biablinks.html

Table 2. Vocabulary sizes for each data set.

	FS 1	**FS 2**	**FS 3**	**FS 4**
Baroque	82	75	40	39
Classical	53	51	36	34
Romanticism	132	120	57	52
Academic	139	127	58	54
Pre-bop	161	138	52	43
Bop	164	149	53	46
Bossanova	155	148	55	49
Jazz	221	193	64	56
Country	73	73	31	31
Pop-rock	125	120	44	41
Popular	134	128	45	42
Total	**256**	**226**	**73**	**63**

FS 1: Degrees with extensions.

FS 2: Chord names with extensions.

FS 3: Degrees without extensions.

FS 4: Chord names without extensions.

FS 1:	IVm7	VII7	III6	V7	Im11
FS 2:	Dm7	G7	C6	E7	Am11
FS 3:	IVm	VII	III	V	Im
FS 4:	Dm	G	C	E	Am

Figure 2. Chord progression taken from "In a sentimental mood" by Duke Ellington (transposed to Am), as it is encoded in each feature set.

basic triad. For every chord, there are 5 variations: *major*, *minor*, *diminished*, *augmented* and *suspended 4th*. As a result, we have four different feature sets, which are ranked in the following list by the amount of encoded information:

- Feature set 1: degrees with extensions (full chord degrees).
- Feature set 2: chord names with extensions (full chords names).
- Feature set 3: degrees without extensions (triad degrees).
- Feature set 4: chord names without extensions (triad names).

In table 2 the amounts of different chords present in each feature set are shown, and figure 2 shows a sample chord progression as it would be represented in each set.

3. Classification methods

Language modeling is a common practice in natural language tasks, such as speech recognition (Jelinek 1998), and also in text categorization (Cavnar and Trenkle 1994).

For the experiments in section 4 two statistical classification methods have been used: a naïve Bayes classifier and n-gram models. Both of them have been widely used in text classification tasks and, more recently, in classification of genres by melody (Cruz-Alcázar et al. 2003, Pérez-Sancho et al. 2005).

Both techniques allow the construction of a statistical model from each genre built form examples, being the main difference that while n-grams make use of context information to compute the probabilities for each word, naïve Bayes computes the probability of each word on its own.

3.1. *Naive Bayes Classifier*

The naive Bayes classifier, as described in (Mccallum and Nigam 1998), 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 $c_j \in \mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$, a melody $\mathbf{x}$ (now represented as a vector) is assigned to the class c_j with maximum a posteriori probability, in order to minimize the probability of error:

$$P(c_j|\mathbf{x}) = \frac{P(c_j)P(\mathbf{x}|c_j)}{P(\mathbf{x})} \,. \tag{1}$$

where $P(c_j)$ is the a priori probability of class c_j, $P(\mathbf{x}|c_j)$ is the probability of $\mathbf{x}$ being generated by class c_j, and $P(\mathbf{x}) = \sum_{j=1}^{|\mathcal{C}|} P(c_j)P(\mathbf{x}|c_j)$.

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 (Domingos and Pazzani 1997). To reflect this independence assumption, melodies can be represented as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|\mathcal{V}|})$, where each component $x_t \in \{0, 1\}$ 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 $P(\mathbf{x}|c_j)$ is given by the probability distribution of words w_t in class c_j, which can be learned from a labelled training set, $\mathcal{X}$, using a supervised learning method.

3.1.1. Multivariate Bernoulli model (MB)

In this model, melodies are represented by a binary vector $\mathbf{x} = (x_1, x_2, \ldots, x_{|\mathcal{V}|})$, where each $x_t \in \{0, 1\}$ represents whether the word w_t appears at least once in the melody. Using this approach, each class follows a multivariate Bernoulli distribution:

$$P(\mathbf{x}|c_j) = \prod_{t=1}^{|\mathcal{V}|} x_t P(w_t|c_j) + (1 - x_t)(1 - P(w_t|c_j)) \tag{2}$$

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 set.

Given a labelled set 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:

$$P(w_t|c_j) = \frac{1 + M_{tj}}{2 + M_j} \tag{3}$$

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

$$P(c_j) = \frac{M_j}{|\mathcal{X}|} \tag{4}$$

Classification of new melodies is performed then using Equation 1, which is

expanded using Equations 2 and 4.

3.1.2. Multinomial model (MN)

This model takes into account word frequencies in each melody, rather than just the occurrence or non-occurence of words as in the MB 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, $\mathcal{L}(\mathbf{x})$, is class-independent (Mccallum and Nigam 1998):

$$P(\mathbf{x}|c_j) = P(\mathcal{L}(\mathbf{x}))\mathcal{L}(\mathbf{x})! \prod_{t=1}^{|\mathcal{V}|} \frac{P(w_t|c_j)^{x_t}}{x_t!} \tag{5}$$

Now, Bayes-optimal estimates for class-conditional word probabilities are:

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

where N_{tj} is the sum of occurrences of word w_t in melodies in class c_j. Class prior probabilities are also calculated as for MB.

3.1.3. 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, in a process known as feature selection. This process 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. This is done by selecting the words which contribute most to discriminate the class of a document, using a ranked list of words extracted from the training set. A widely used measure to rank these words is the *average mutual information* (AMI) (Cover and Thomas 1991), which gives a measure of how much information is provided by each single word. Informally speaking, we can consider that a word is informative when it is very frequent in one class and less in the others.

For the MB 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[1]:

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

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.

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

Progression:	I – V – VI – V – I			
Bigrams:	I – V	V – VI	VI – V	V – I
Trigrams:	I – V – VI	V – VI – V	VI – V – I	

Figure 3. Decomposition of a chord progression in bigrams and trigrams.

In the MN 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, Equation 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.

3.2. *n-grams*

A language model is a probability distribution that assigns a probability to a sequence of words $P(w_1, \ldots, w_k)$, so that the probability of each word in the sequence is dependent on its *context* $P(w_i|w_1, \ldots, w_{i-1})$.

Estimating the probabilities of such a model can be an arduous task, and maybe computationally unaffordable, when dealing with long sequences. This is why language models are often approximated using n-gram models. An n-gram is a sequence of n words in which the first $n-1$ words are considered as the context. Thus, the estimated probability of a word w_i given a context is computed as $P(w_i|w_{i-n+1}, \ldots, w_{i-1})$. Typical values for n are 2 (*bigrams*) and 3 (*trigrams*).

3.2.1. Classification

In order to perform genre classification, a language model must be constructed for each genre in the data set. Each sequence (song) in the data set is decomposed in n-grams of a fixed length n (see Fig. 3). Then, the probability of each different n-gram is computed as the probability of the last word given its context. This probability can be easily calculated by dividing the number of occurrences of the n-gram by the number of occurrences of its context in the given data set:

$$P(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{\mathcal{N}(w_{i-n+1}, \ldots, w_i)}{\mathcal{N}(w_{i-n+1}, \ldots, w_{i-1})} \tag{8}$$

Once a language model is constructed for each genre, the probability that a new sample $w = w_1, \ldots, w_k$ has been generated by model c is:

$$P_c(w) = \prod_{i=1}^{k} P_c(w_i|w_{i-n+1}, \ldots, w_{i-1}) \tag{9}$$

Thus, a test sample can be classified by following the risk minimization criterion, i.e. given a set of classes $\mathcal{C} = c_1, \ldots, c_{|\mathcal{C}|}$, each test sample is assigned to the class $\tilde{c}$ of the model that holds $\tilde{c} = \arg\max_c P_c(w)$.

Another common practice to evaluate a language model is by measuring its perplexity given a test sample w. Perplexity is strongly correlated with the probability of generating a sample shown in Eq. (9), and it can be intuitively interpreted as how surprised is our model when a new sample is presented to it: the lower the

perplexity, the higher the probability that our model has generated that sample. For an n-gram model, perplexity is calculated as follows:

$$PP(w) = P(w_1, \ldots, w_k)^{-\frac{1}{k}} = \sqrt[k]{\frac{1}{\prod_{i=1}^{k} P(w_i|w_{i-n+1}, \ldots, w_{i-1})}} \tag{10}$$

Using this measure, classification can be performed by selecting the class of the model with lower perplexity $\tilde{c} = \arg\min_c PP_c(w)$. In this paper we will follow this approach, using the CMU SLM Toolkit (Clarkson and Rosenfeld 1997).

3.2.2. Parameter smoothing

Even when the training set is big enough to build a good language model, there can be situations where we can find words in a test sample that have not been seen previously. When such situation occurs, the probability of the n-grams containing that words is zero, thus causing the probability of the whole sequence being zero by the application of Eq. (9).

To avoid this problem, it is common to use a procedure known as *smoothing*, in which a small probability is substracted from the set of known words, and then shared out among all unseen words. There are several techniques to calculate the optimal amount of probability that must be taken off, and what percentage of it must receive every unseen word. In this work the *Witten-Bell* discounting method has been used.

4. Experiments

In order to test the methods explained in section 3, two different experiments were performed. In the first experiment, the data set was divided in the three music domains: academic, jazz, and popular music. The aim of this experiment is to test whether the utilized models are able to distinguish among different music categories, as a first step to a more in deep framework.

The second experiment was performed using all the sub-genres to evaluate the performance of using this harmonic information to capture more subtle differences among genres.

Both experiments were done using the training sets described in section 2: the original set of degrees and chord sequences with extensions, and the corresponding reduced sets with only triads (both with names and degrees). This way, we are able to compare the loss in the system performance associated with the loss of information. All the experiments reported here have been validated using a 10-fold validation scheme: the data set is splitted in two: 90% is used to train the models and the remaining 10% is used for test. This process is repeated 10 times, and then the results obtained in each sub-experiment are averaged.

4.1. *Experiment 1: three-genres classification*

Since the Naive Bayes classifier has one more parameter to evaluate than the n-gram models, namely the vocabulary size used to build the feature vectors, we must first determine the optimal number of chords that should be used in order to achieve a good classification rate, and also to be able to compare its results with those obtained using the n-gram models. In Fig. 4 the evolution of the success rate as a function of the utilized vocabulary size can be observed. From the results we can draw the following conclusion: the higher the number of chords used, the

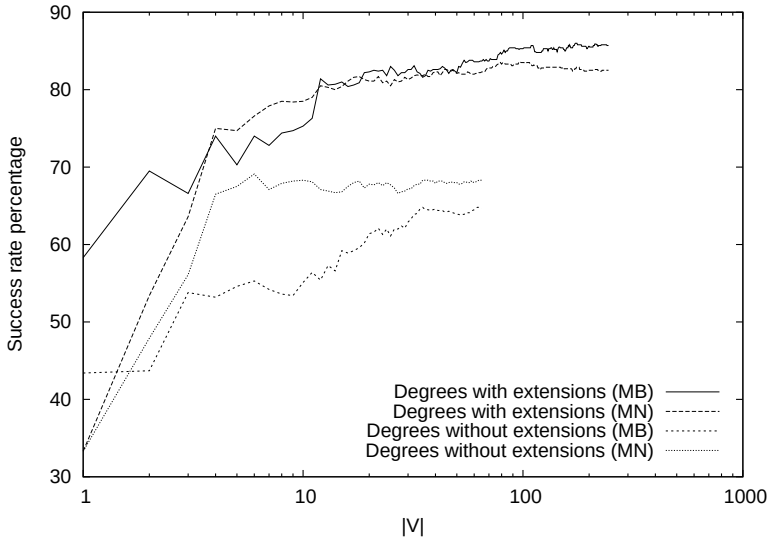

Figure 4. Results obtained with the Naive Bayes classifier using the Multivariate Bernoulli (MB) and Multinomial (MN) statistical models.

better are the results obtained. In the case of classification using chord degrees without extensions, it is necessary to use the whole set of chords, whereas when using chord degrees with extensions, it seems that good results are achieved with a vocabulary size of around 100 chords. From now on, we will use these vocabulary size to compare this method with the n-gram models. Also, it doesn't seem to be a significant difference between using Multivariate Bernoulli and Multinomial statistical models, so we have chosen MB model since it is the one who obtained (slightly) higher classification rates.

In Fig. 5 the success rates using n-gram models with $n \in \{2, 3, 4\}$ are shown. In this plot the results using the four feature sets are displayed in order to emphasize the difference in performance when using different amounts of information. As expected, the feature set containing chords with their extensions (all information available) provided much better results than using just triad chords. These results support our hypothesis that better models can be built based on more precise information. On the other hand, the major/minor key information that allows to use degrees instead of chord names only contributes with a small improvement of the performance.

Note that the use of longer n-grams seems to compensate the lack of key information (both success rates for degrees and chords are closer as n increases, specially for $n = 4$), since longer chord progressions can help to disambiguate possible confusions: whereas the sequence C – G – Em could be interpreted as I – V – III in C key or III – VII – V in Am key, it is more likely that the longer sequence C – G – Em – Am appears as III – VII – V – I in Am.

Finally, the best success rates for each classification method are compared in table 3. As it can be seen, the best results were obtained by the naïve Bayes classifier using full chord names (feature sets FS1 and FS2) although they were not much better than those obtained with n-grams using the same feature set.

On the contrary, n-gram models outperformed naïve Bayes when using the triad sets of chords (feature sets FS3 and FS4). In these cases the use of context information is essential in order to make better decisions. This is specially relevant for feature set FS4 where just triad names were utilized.

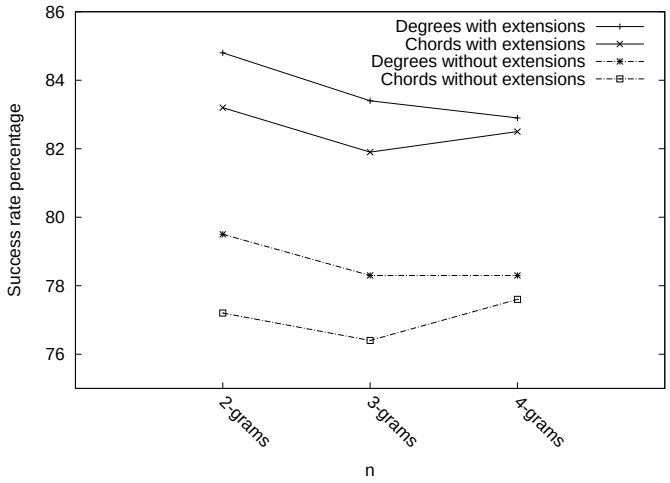

Figure 5. Performances with n-grams as a function of the context size n.

Table 3. Best classification rates obtained with n-grams and Naive Bayes for the three-classes problem.

	2-grams	3-grams	4-grams	MB	MN
Degrees with extensions	84.8	83.4	82.9	**85.3**	83.5
Chord names with extensions	**79.5**	78.3	78.3	64.9	68.3
Degrees without extensions	83.2	81.9	82.5	**84.5**	81.4
Chord names without extensions	77.2	76.4	**77.6**	62.5	69.0

Table 4. Best classification rates obtained with n-grams and Naive Bayes for the eight-classes problem.

	2-grams	3-grams	4-grams	MB	MN
Degrees with extensions	45.3	45.0	43.0	**48.4**	47.5
Chord names with extensions	45.5	45.9	46.0	48.5	**49.8**
Degrees without extensions	46.9	**48.1**	48.0	33.6	38.3
Chord names without extensions	47.0	**47.9**	46.5	33.9	40.4

4.2. *Experiment 2: eight-genre classification*

Once the capabilities of the system to distinguish among the broad music domains have been tested, the second experiment tries to investigate how these models performed in a more complex task. In this experiment the data set consists of eight different classes, corresponding to the eight sub-genres described in section 2.

Table 4 shows the average success rate for each method. The performance is poorer now, around 50% in the best cases, but note that the baseline of the recognition rate is now 12.5%. When these results are studied in more detail (see Fig. 6) one sees that the errors are mainly caused within the broad domains. For example, it is particularly difficult for the system to properly distinguish between baroque and classical music, or between pop-rock and country, or among jazz sub-genres. On the other hand, misclassifications among different domains are much less frequent. For example, none of the academic pieces was classified as a jazz sub-genre.

These results are in keeping with those presented for the three-genre case: the model is able to capture well the harmonic differences among music domains but it performs poorer for closer music genres. Anyway, the existence of a diagonal in the confusion matrix suggests that these results could be improved with a more sophisticated classification scheme.

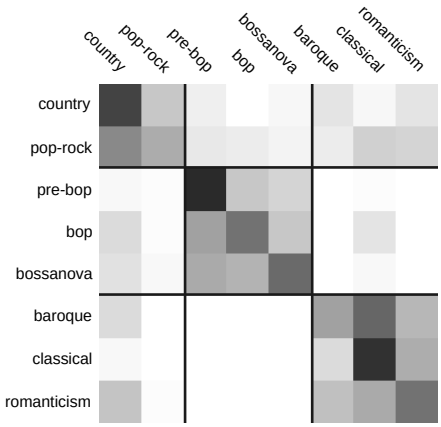

Figure 6. Confusion matrix for the 8-class problem using classification by 2-grams. Grey levels represent the classification percentages. Rows are the ground truth and columns the actual system output. The parts of the matrix corresponding to the different music domains have been highlighted.

5. Conclusions and future work

In this paper, the feasibility of classifying music in genres using harmonic information has been tested. For this, we applied n-gram models and a naïve Bayes classifier on a corpus of chord progressions from three root genres (academic, jazz and popular music), and eight leaf sub-genres. In order to test the amount of information required in this task, four different feature sets were extracted from the corpus using different levels of information, from triads to full chords with extensions including tonality information (encoded as degrees).

In the first experiment, when classifying between the three root genres, and using chord degrees with extensions, the naïve Bayes classifier achieved a recognition rate of 85.3%, not being a significant difference between this method and the n-gram models. But when using fewer information it is necessary the use of context (n-grams) to achieve good results. Using the simpler feature set, i.e. progressions of triads, the system reached a 77.6% success rate using 4-grams. This is a promising result, as it suggest that state-of-the-art audio to chord transcription systems could be used to obtain this kind of chord progressions in the absence of real harmonic information. These features could be used in conjuntion with the original audio signal as the input of a classifier, in the same way that similar systems using melodic representations extracted from audio already do.

The experiment using the decomposition of the data set in eight classes obtained poorer results, around 50%, but classification errors were mainly made between sub-genres inside each of the three domains. Further experimentation should be done with more sophisticated methods, as hierarchical classification, to improve these results. Also, we are considering to incorporate melodic information which, combined with harmony, could help to improve system performance.

Acknowedgements

The authors want to thank Pedro J. Ponce de León and José L. Heredia for their help and advice. This work is supported by the Spanish CICyT PROSEMUS project (TIN2006-14932-C02) an the research programme Consolider Ingenio 2010 (MIPRCV, CSD2007-00018).

References

Tzanetakis, G., and Cook, P. (2002), "Musical Genre Classification of Audio Signals," *IEEE Transactions on Speech and Audio Processing*, 10, 205–210.

McKay, C., and Fujinaga, I. (2004), "Automatic genre classification using large high-level musical feature sets," in *Proceedings of the ISMIR*, pp. 525–530.

Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. (2007), "Improving genre classification by combination of audio and symbolic descriptors using a transcription system," in *Proceedings of the ISMIR*, Vienna, Austria, pp. 61–66.

Cataltepe, Z., Yaslan, Y., and Sonmez, A. (2007), "Music Genre Classification Using MIDI and Audio Features," *EURASIP Journal on Advances in Signal Processing*, 2007.

Piston, W., *Harmony*, 5th. edition ed., Norton, W. W. & Company, Inc. (1987).

Herrera, E., *Teoría musical y armonía moderna*, Vol. II, Antoni Bosch Editor (in spanish) (1998).

Tzanetakis, G., Ermolinskiy, A., and Cook, P. (2002), "Pitch Histograms in Audio and Symbolic Music Information Retrieval," in *Proceedings of the ISMIR*, Paris, France.

Shan, M.K., Kuo, F.F., and Chen, M.F. (2002), "Music style mining and classification by melody," *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, 1, 97–100 vol.1.

Paiement, J.F., Eck, D., and Bengio, S. (2005), "A Probabilistic Model for Chord Progressions," in *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*, pp. 312–319.

Lee, K., and Slaney, M. (2006), "Automatic chord recognition from audio using a supervised HMM trained with audio-from-symbolic data," in *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, Santa Barbara, California, USA, New York, NY, USA: ACM, pp. 11–20.

Sheh, A., and Ellis, D.P.W. (2003), "Chord segmentation and recognition using EM-trained hidden markov models," in *Proceedings of the 4th International Conference on Music Information Retrieval, ISMIR 2003*.

Bello, J.P., and Pickens, J. (2005), "A Robust Mid-Level Representation for Harmonic Content in Music Signals," in *Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR 2005*, pp. 304–311.

Jelinek, F., *Statistical Methods for Speech Recognition*, MIT Press (1998).

Cavnar, W.B., and Trenkle, J.M. (1994), "N-Gram-Based Text Categorization," in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, US, pp. 161–175.

Cruz-Alcázar, P.P., Vidal, E., and Pérez-Cortes, J.C. (2003), "Musical Style Identification Using Grammatical Inference: The Encoding Problem," in *Proceedings of the 8th Iberoamerican Congress on Pattern Recognition, CIARP 2003*, pp. 375–382.

Pérez-Sancho, C., Iñesta, J.M., and Calera-Rubio, J. (2005), "Style recognition through statistical event models," *Journal of New Music Research*, 34(4), 331–340.

Mccallum, A., and Nigam, K., "A comparison of event models for Naive Bayes text classification," (1998).

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

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

Clarkson, P.R., and Rosenfeld, R. (1997), "Statistical Language Modeling Using the CMU-Cambridge Toolkit," in *Proceedings of ESCA Eurospeech*.