Musical Style Identification Using Grammatical Inference: The Encoding Problem*

Pedro P. Cruz-Alcázar[1], Enrique Vidal-Ruiz[2], and Juan C. Pérez-Cortés[1]

[1] Departamento de Informática de Sistemas y Computadores
Universidad Politécnica de Valencia
Valencia, Spain
{ppcruz,jcperez}@disca.upv.es

[2] Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Valencia, Spain
evidal@dsic.upv.es

Abstract. A Musical Style Identification model based on Grammatical Inference (GI) is presented. Under this model, regular grammars are used for modeling Musical Style. Style Classification can be used to implement or improve content based retrieval in multimedia databases, musicology or music education. In this work, several GI Techniques are used to learn, from examples of melodies, a stochastic grammar for each of three different musical styles. Then, each of the learned grammars provides a confidence value of a composition belonging to that grammar, which can be used to classify test melodies. A very important issue in this case is the use of a proper music coding scheme, so different coding schemes are presented and compared, achieving a 3 % classification error rate.

1 Introduction

Grammatical Inference (GI) aims at learning models of languages from examples of sentences of these languages. Sentences can be any structured composition of primitive elements or symbols, though the most common type of composition is the concatenation. From this point of view, GI find applications in all those many areas in which the objects or processes of interest can be adequately represented as strings of symbols. Perhaps the most conventional application areas are Syntactic Pattern Recognition (SPR) and Language Modeling. But there are many other areas in which GI can lead to interesting applications. One of these areas is Musical Style Identification (MSI). Here, the very notion of language explicitly holds, where primitive symbols or "notes" are adequate descriptions of the acoustic space, and the concatenation of these symbols leads to strings that represent musical sentences. By adequately concatenating symbols of a given musical system, a musical event emerges. However, not any possible concatenation can be considered a "proper" event. Certain rules dictate what can or can not be considered an appropriate concatenation, leading to the concept of

* This work has been partially supported by CYCIT Project TIC2000-1703-C03-01 "TAR".

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 375–382, 2003.
© Springer-Verlag Berlin Heidelberg 2003

musical style. Main features of a musical style are rhythm and melody, which are directly related with the rules used to concatenate duration and pitch of sounds, respectively.

The interest in modeling musical style resides in the use of these models to generate (Automatic Composition) and classify music by style (MSI), which are our areas of interest. The MSI area is being recently explored, mostly in the field of multimedia databases, trying to improve content based retrieval in multimedia databases, allowing indexing by musical style in addition to other suitable indexes. But other applications can be musicology (finding authors for anonymous pieces) or music education. Some AI techniques that have been employed are Hidden Markov Models [11], Self-Organising Maps [12] and Neural Networks [17]. This paper is focused in our MSI work [4] [5] that have been extended by adding new coding schemes for music.

2 Grammatical Inference

In this section we first quickly review the field of GI and later we will explain briefly the techniques used in our study. *Grammatical Inference* is a well established discipline originated by the work of Gold about *"Language Identification in the Limit"* [8]. Perhaps the most traditional applicative field of GI has been Syntactic Pattern Recognition; but there are many other potential applications. In general, GI aims of finding the rules of a grammar G, which describes an unknown language, by means of a positive sample set $R_+ \subseteq \{ \alpha \mid \alpha \in L(G) \}$ and a negative sample set (of samples that should be rejected) $R_- \subseteq \{ \beta \mid \beta \in \Sigma^* - L(G) \}$. A more limited framework, but more usual in practice, is the use of only positive samples. The main problem lies in finding a more "*abstract*" or general grammar G' so that $R_+ \subset L(G')$ and $L(G') - R_+$ (the extra-language) contains only strings with "*similar features*" to the strings from R_+. A grammar G is said to be *identified in the limit* [8] if, for sufficiently large sample sets, L(G')=L(G). Grammars and Formal Languages can be seen from a probabilistic viewpoint too, in which different rules have different probability of being used in the derivation of strings. The corresponding extension of GI is called *Stochastic Grammatical Inference* and this is the way of work of all the techniques used in our study. Next, we concisely explain three of the techniques used in our study to infer the grammars employed for *composing* and *identifying* musical styles. Other techniques employed with less success are Regular Positive Negative Inference (RPNI) [10] and a State Merging Technique based on probabilistic criteria called ALERGIA [3]. These techniques are fairly well known in the GI community and have been proven useful in other fields.

The Error-Correcting Grammatical Inference (ECGI) technique is a GI heuristic that was explicitly designed to capture relevant regularities of concatenation and length exhibited by substructures of unidimensional patterns. It was proposed in [14] and relies on error-correcting parsing to build up a stochastic regular grammar through a single incremental pass over a positive training set. This is achieved through a Viterbi-like, error-correcting parsing procedure [6] which also yields the corresponding optimal sequence of both non-error and error rules used in the parse.

Similarly, the parsing results are also used to update frequency counts from which probabilities of both non-error and error rules are estimated. One of the most used models in Natural Language Processing are N-Grams (a class of Markov models) [9]. An N-Gram is a sequence of symbols of length N. The first N-1 of these are the context. The size of N can in theory be anything from 1 upwards. However, certain values are better than others at capturing the characteristics of the language. The larger the value of N, the more context is captured. Though it would seem useful to have a great N, it is not a case of the larger the better. As N grows, it captures more context. Eventually the sequences learned become not just characteristic of the corpus, but the exact sequences in the corpus. The parameter estimation method can be consulted in [9]. In our study, modeling with N-Grams is performed using the CMU-Cambridge Statistical Language Modeling Toolkit (SLM-Toolkit) [13]. The k-TSI technique infers *k-Testable Languages in the Strict Sense* (k-TSSL) in the limit. It has been demonstrated that they are equivalent to N-Grams with N=k [15]. The main difference between them is that it is generally assumed that an N-Gram embodies the (N-i)-Grams [i=1..N], while a k-TSSL model consists only in the model of order k. The inference of k-TSSLs was discussed in [7] where the k-TSI technique was proposed.

As each inferred automaton or N-Gram model represents a musical style, "recognizing" the style of a test melody consists in finding the automaton which best recognizes this melody. This can be best achieved by using an algorithm that performs stochastic Error-Correcting Syntactic Analysis through an extension of the Viterbi algorithm [6]. The probabilities of error rules (Insertion / Deletion / Substitution) can be estimated from data [1]. The Analysis Algorithm returns the probability that the analyzed melody is (error-correcting) generated by the automaton. By analyzing the same melody with different automata, we classified it as belonging to the musical style (language) represented by the automaton that gave the largest probability.

3 How to Code Music for Syntactic Pattern Recognition

GI, as a Syntactic Pattern Recognition (SPR) technique, works with symbol strings. Even though only duration and pitch of sounds (as main features of music) are used in this work, the way they are represented implies the inclusion of more or less musical information. The amount and/or meaning of this musical information can be key in for the success in Musical Style Recognition (MSI). So, we have to deal with the selection of pitch and duration representations and the coding into symbol strings. Many efforts have been done within the Computer Music research community in musical representation systems and it is not clear that one system is always better than the others [2] [16] [18], being very dependant on the application and the recognition paradigm. In next subsections, we present the pitch and duration representations that, in our opinion, are more suitable for a SPR technique as GI, as well as some encoding examples into symbol strings. Brief comments will be done for every representation, according to its performance for MSI and Automatic Composition (AC) with GI.

3.1 Pitch Representation

Absolute. Pitch is most often represented either by the traditional pitch naming system (e.g. F#4-G#4-A4) or as absolute pitch (e.g. in MIDI: 66, 68, 69). It is the same representation as in musical scores, but may be insufficient for applications in tonal music. The main problem is that transpositions are not accounted for (e.g. repeating the same pitch motive transposed in different samples or within the same one).

Relative. A solution for the transposition problem is the use of the *relative pitch* between notes. That is, the *interval* (number of semitones) between two notes. There exists a somewhat 'peculiar' relationship between pitch strings and pitch interval strings. If one pitch interval in a string of pitch intervals is altered then *all* the succeeding notes are altered (transposed) [2]. So a change in a string of pitches and in a string of pitch intervals is not exactly the same thing. This effect appeared in our AC experiments [4].

Melodic Contour. Pitch interval encodings readily lend themselves to the construction of a number of more abstract representations of musical strings such as *contour strings*. Intervals can be categorised in a number of classes according to the signs of intervals. Instead of taking the absolute or relative pitch, it is coded if the next pitch goes "up" (U), "down" (D) or it is "equal" (E) to the last one. So, melodic contour can be represented as a string from the alphabet {U, D, E}, leading to a very small alphabet which provides less musical information than previous representations.

Relative to Tonal Centre. This coding scheme arose along with our experiments in AC [5] in order to correct the *relative* representation 'peculiar' effect mentioned before. Pitch is coded as the distance to the *tonal centre* or *tonic* in semitones. It includes more musical information than the others, as it allows characterizing relationships between pitches and *tonality*.

3.2 Duration Representation

In terms of the rhythmic component of musical strings, almost the same representations as pitch ones can be applied. It should be noted, however, that the problems that arise with pitch representations (highlighted in the previous section) apply also for duration representations.

Absolute. It is a direct translation of the representation used in musical scores (e.g. whole note, half note, quarter note, and so on). It is the most commonly applied in musical string processing algorithms [2] and, to this end, is the only one we have tested.

Relative. It is well known that listeners usually remember a rhythmic pattern as a relative sequence of durations that is independent of an absolute tempo. So, with *absolute* duration encoding, the same rhythmic pattern written with two different *metrics* will be considered as two different patterns. Representing rhythm as *duration ratios* can overcome augmentations or diminutions of a rhythmic pattern (Fig. 1).

Rhythmic Contour. Like melodic contour, durations can be coded in terms of *contour* strings. Instead of taking the absolute or relative duration, it is coded if the next duration is "shorter" (S), "longer" (L) or "equal" (E).

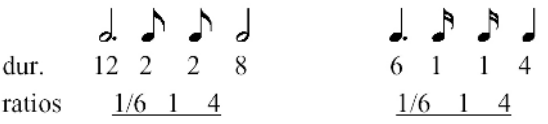
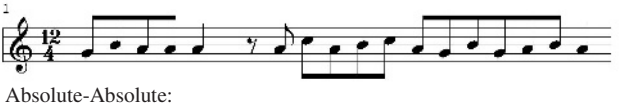


Fig. 1. Two rhythmic patterns that match at the level of duration ratios are in fact the same

3.3 Musical String Representation

Notes can be coded as one symbol (what we call *not-splitted* encodings), or with two symbols (what we call *splitted* encodings). In the beginning of our study, only *not-splitted* encodings where used, but we realized that these coding schemes could be establishing a relationship between the pitch and length stronger than the one generally existing in music. Therefore, *splitted* encodings were introduced as a modification from previous ones, just separating with a space character the pitch and duration symbols (Fig. 2), obtaining better results in MSI (see section 5).

Combining pitch and duration representations exposed in the previous subsections, 12 *not-splitted* encodings can be defined. *Splitting* them, other 12 arise and, if pitches and durations are used stand-alone, 7 more emerge. As a result, we have 31 different coding schemes that can be tested. For naming conventions, the format "pitch representation name - duration representation name" is used (Fig. 2). The "*splitted*" term is not used with *not-splitted* encodings. To this end, 13 coding schemes have been tested and results are presented in next section. In order to obtain the *musical string*, for these coding schemes, we have used numbers for pitch representations (except for *contour* representation) and letters for duration representation (only *absolute* representation has been tested). Thus, musical strings are easier to understand, as can be seen in Fig. 2.



Absolute-Absolute:
22c 26c 24c 24c 24n 0c 24c 27c 24c 26c 27c 24c 22c 26c 22c 24c 26c 24n
Relative-Absolute:
Sc 4c -2c 0c 0n 99c 0c 3c -3c 2c 1c -3c -2c 4c -4c 2c 2c -2n
Splitted Contour-Absolute:
S c U c D c E c E n R c E c U c D c U c U c D c D c U c D c U c U c D n

Fig. 2. A Gregorian style score coded with different representations

4 Experiments

We chose 3 occidental musical styles from different epochs and we took 100 sample melodies from each one. These samples were 10 to 40 seconds long. The first style was *Gregorian Chant*. As a second style, we used passages from the sacred music of *J. S. Bach*. The third style consisted of passages from *Scott Joplin*'s Ragtimes. Experiments in Style Identification were performed using ECGI, k-TSI, N-Grams and other GI techniques (mentioned in section 3). Three automata (one per style) are inferred with each GI technique, trying different values of k (with k-TSI) and N (with N-Grams). Test melodies are analyzed to see which of the learned automaton can generate them with the greatest probability. Given the small size of the available corpus, 10-fold *Cross-Validation* was used to measure the identification accuracy of the different techniques and coding schemes. *Average Classifying Error* in identification for each style was obtained and the best results are presented here.

4.1 Results

For the sake of conciseness, results will be summarized in Table 1, which shows the best *Average Classifying Error* for each GI technique with some of the tested encodings. Due to N-Gram's results, the *stand-alone* and *tonal centre* encodings have only been tested with them. Results are worst than the obtained with the other coding schemes, being the best one a 6.66 % classifying error when using *tonal centre* pitch representation.

Table 1. Classifying Error in Musical Style Identification experiments with the different GI techniques and coding schemes employed

	ECGI	**K-TSI**	**N-GRAM**
Absolute-Absolute	34.33 %	10 %	4.66 %
Relative-Absolute	13.66 %	8.66 %	**4.33 %**
Contour-Absolute	8.66 %	7 %	5.33 %
Tonal Centre-Absolute			5.33 %
Splitted Absolute-Absolute	9.66 %	5.66 %	3.33 %
Splitted Relative-Absolute	**7.66 %**	**5 %**	**3 %**
Splitted Contour-Absolute	10 %	5.66 %	5.33 %
Splitted Tonal Centre-Absolute			4.66 %

4.2 Discussion

Analyzing these results by GI techniques, the best of them is clearly the N-Gram technique. A 3% error in Style Identification is obtained, comparing with a 5% using k-TSI and a 7.66 % using ECGI techniques. Although comparisons with the success rates of other style identification models is not very meaningful unless the same datasets are used, if we look to other similar studies [11] [12] [17], these average rates of success can be considered as quite good. It is worth noting that GI techniques tend

to need a larger quantity of training samples to get good results. So, it is expected that our results will be improved if the amount of training samples is increased. The advantage of the N-Gram technique, over k-TSI and ECGI, relays on using the *Back-off smoothing* procedure within the analysis algorithm [9]. It is well known in the Pattern Recognition community that N-Gram inference with back-off smoothing outperforms many other Syntactic Pattern Recognition techniques and, in our study, this has been a fact.

Although the GI technique used for modeling musical style is very important, as seen before, our study has shown that the musical string representation is determinant in the results. It is clear that with the *relative* pitch representation are obtained the best results, consequently, it is expected that *relative* duration representation will be a good option to try in the future, specially *splitted relative-relative* encoding. The *stand-alone* encodings results are bad, showing the necessity of more musical information within the coding scheme. Thus, we do not consider the future use of the remainder (*relative* duration and *contour* duration). Although with *tonal centre* representation, Automatic Composition results were improved, in MSI it has been very different. It can be due to this approach does not accounts for transpositions of a pattern within the same piece. An important fact is that *contour* pitch encodings, specially the *splitted* one, have not achieved as good results as the others. It is due to the small size of the alphabet (symbols) for these coding schemes (e.g. only 9 symbols for Gregorian style in the splitted encoding).

Another conclusion of the study is that *splitted* encodings are clearly better than joined pitch and duration representations. As a result of this discussion, for future studies, we can discard *contour* and *stand-alone* representations, remaining 6 combinations with *relative* duration representations to be tested. Of them, the *splitted* encodings are expected to be the best.

5 Conclusions and Future Works

A Musical Style Identification (MSI) model based on Grammatical Inference is presented. Different coding schemes for music have been proposed and compared according to their suitability for working with Syntactic Pattern Recognition techniques and the results obtained in our experiments. Result from this work shows the need of proper music coding schemes, being the most important the use of two separated symbols (pitch and duration) for the encoding of each musical note. The best results in MSI have been obtained with the N-Gram technique, achieving a 3 % classifying error. Several lines of study can be followed to attempt improving results. First, the amount of data used so far is insufficient and better performance is expected by increasing the number of training samples. From the coding schemes presented in this work there are still 6 to be tested. Of course, other coding schemes must be explored, as trees or strings labelled with information about modulations or harmony. Once these tasks are dealt with, we could employ entire musical pieces as samples, and not just small fragments as was done in this study.

References

1. Amengual, J.C.; Vidal, E. 1996. Two Different Approaches for Cost-efficient Viterbi Parsing with Error Correction. Advances in Structural and Syntactic Pattern Recognition, pp. 30-39. P. Perner, P. Wang and A. Rosenfeld (eds.). LNCS 1121. Springer-Verlag.
2. Cambouropoulos, E. et al. 2001. Pattern Processing in Melodic Sequences. Computers and the Humanities 35(1):9–21
3. Carrasco, R.C.; Oncina, J. 1994. Learning Stochastic Regular Grammars by means of a State Merging Method. "Grammatical Inference and Applications". Carrasco, R.C.; Oncina, J. eds. Springer-Verlag, (Lecture notes in Artificial Intelligence (862)).
4. Cruz, P.; Vidal, E. 1998. Learning Regular Grammars to Model Musical Style: Comparing Different Coding Schemes. Proceedings of the 4th International Colloquium on Grammatical Inference ICGI-98, Vasant Honavar, Giora Slutzki (Eds.) Lecture Notes in Artificial Intelligence 1433, Subseries of Lecture Notes in Computer Science.
5. Cruz, P.; Vidal, E. 2003. Modeling Musical Style Using Grammatical Inference Techniques: a Tool for Classifying and Generating Melodies. Proceedings of the 3rd International Conference on Web Delivering of Music (WEDELMUSIC 2003), Leeds, UK. To appear in Septermber 2003. IEEE Computer Society.
6. Forney, G. D. 1973. The Viterbi algorithm. IEEE Proc. 3, pp. 268–278.
7. García, P.; Vidal, E. 1990. Inference of K-Testable Languages In the Strict Sense and Application to Syntactic Pattern Recognition. IEEE Trans. on PAMI, 12, 9, pp. 920–925.
8. Gold, E. M. 1967. Language Identification in the Limit. Inf. and Control, Vol. 10, pp. 447–474.
9. Jelinek, F. 1998. Statistical Methods for Speech Recognition. MIT Press.
10. Oncina, J.; García, P. "Inferring regular languages in polynomial update time. Pattern Recognition and Image Analysis". N. Pérez de la Blanca, A. Sanfeliú and E. Vidal eds. World Scientific, 1992.
11. Pollastri, E.; Simoncelli, G. 2001. Classification of Melodies by Composer with Hidden Markov Models. Proceedings of the First International Conference on WEB Delivering of Music WEDELMUSIC'01. IEEE Computer Press, pp. 88–95.
12. Ponce de León, P. J.; Iñesta, J. M. 2002. Musical Style Identification Using Self-Organising Maps. Proceedings of the Second International Conference on WEB Delivering of Music WEDELMUSIC'02. IEEE Computer Press, pp. 82–92.
13. Rosenfeld, R.; Clarkson, P. 1997. Statistical Language Modeling using the CMU-Cambridge Toolkit. Proceedings of the Eurospeech'97. Volume 5, pages 2707–2710.
14. Rulot, H.; Vidal, E. 1987. Modelling (sub)string-length based constraints through a Gramatical Inference method. NATO ASI Series, Vol. F30 Pattern Recognition Theory and Applications, pp. 451–459. Springer-Verlag.
15. Segarra, E. 1993. Una Aproximación Inductiva a la Comprensión del Discurso Continuo. PhD diss. Facultad de Informática. Universidad Politécnica de Valencia.
16. Selfridge-Field, E. 1998. Representing Musical Information for Retrieval. Invited talk, Association for Computing Machinery: SIGIR: Exploratory Workshop on Music Information Retrieval, Berkeley, CA; 19 August 1998.
17. Soltau, H. et al. 1998. Recognition of Music Types. Proceedings of the International Conference on Acoustics, Speech and Signal Processing ICASSP 98, pp. 1137–1140. Seattle, Washington.
18. Wiggins, G. et al. 1993. A Framework for the Evaluation of Music Representation Systems. Computer Music Journal 17(3):31–42