In Organised Sound 4(2), 115-119. 1999

An Intelligent Hybrid Model for Chord Prediction

Uraquitan Sidney Cunha & Geber Ramalho
Departamento de Informática -Universidade Federal de Pernambuco
Cx. Postal 7851, 50740-540 Recife, PE, Brazil
{usgcc, glr}@di.ufpe.br

Abstract

For a system to be able to generate real time accompaniment for previously unknown songs, it must predict their harmonic development, i.e., the chords to be played. We claim that such a system must combine long term experience, to identify typical chord sequences (e.g., II-V and II-V-I), with "on the flight" adaptation to track recurrent structures (e.g., choruses and refrains) of the particular song being played. We have implemented a prediction system using a neural network model that encompasses prior knowledge about typical chord sequences. The achieved results have been very encouraging, really better than those reported in the literature. However, our predictor could not adapt its behavior according the idiosyncrasies of each song, since on-line learning is nearly impossible in neural networks. In this paper, we propose an extension of our previous work by the inclusion a rule-based sequence tracker, which detects recurrent chord sequences while the song is being performed. We show that this hybrid model, combing a neural network predictor with a rule-based sequence tracker, improves the system's performance.

1 INTRODUCTION

This decade have witnessed the development of real time interactive accompaniment systems, which have been used as rehearsal or concert partners, as well as arrangement assistants (Ramalho 1997). To improve the interactivity of these systems, it is important to add to them the capacity of predicting the next event (chords or notes) of the song being performed. This musical task is a particular case of time series prediction, which is concerned with foreseeing a new event based on stream of events that have occurred so far (Weigend, 1993). The success of time series prediction tasks depends on various issues, including problem dimensionality reduction, context representation, as well as knowledge representation and acquisition.

A central issue here is to build a predictor that can capture typical behavior patterns of the problem class under study (prior knowledge), as well as the specific behavior patterns of the particular problem instance (in-the-flight knowledge). This dichotomy between prior and in-the-flight knowledge holds particularly in the problem of musical chord prediction. Prior knowledge on typical chord sequences (such as II-V-I, V-I, etc.) is of great help in prediction. However, it is also necessary to acquire in-the-flight the knowledge concerning each song-specific recurrent structures (such as choruses, sections, etc.). Even for senior musicians, it can be very hard to perform a real-time accompaniment of an unknown song, mainly during the initial measures of the song. However, as times goes by, the musicians detect some repetitive chord sequences and re-use them to improve their capacity to predict the future chords.

From a computation standpoint, the problem is to determine how to bring together the prior knowledge, learned earlier by many song examples, and the experience being acquired during the song execution.

We have implemented a prediction system using an MLP-backpropagation neural network (Rumelhart & McClelland 1986) encompassing prior knowledge about typical chord sequences (Cunha & Ramalho 1998). This knowledge is obtained by supervised learning (Mitchell 1997) on a corpus composed by more than 30 jazz standards. In our experiments, we have achieved good results (about 15% of error rate). These are significantly better results than those presented in the literature for tonal music real time chord prediction. However, the difficulties of doing on-line learning in neural networks are well known. Without on-line learning, it would be impossible to adapt the behaviour of our neural network predictor according the idiosyncrasies of each song. For this reason, we have proposed an extension of this neural network-based work with the inclusion of a rule-based sequence tracker, which is capable of detecting recurrent chord sequences while the song is being performed. In this paper we presents this hybrid model, showing that it improves the prediction rates with respect to the single neural network performance.

Sections 2 and 3, present, respectively, the chord prediction problem and the research efforts made to solve it. In Section 4, it will be explained how our hybrid model was designed and developed. The results are presented in Section 5. Finally, we draw some final considerations and point out future research.

2 THE PROBLEM

How can a computer predict, in real time, chords of previously unknown song? A chord predictor task is to determine which would be the “next chord” of a song based only on the previously played chords and melody (Cf. Figure 1).

Figure 1 - Chord prediction task

The implementation of such a prediction system involves complex issues. First of all, chord prediction must be performed on line, under hard time constraints. The real time constraints have a great impact on the kind of algorithm that can be employed, since the computational resources are limited.

Another issue is the fact that there are no universal rules for chord prediction. If any “chord chaining rules” exist, they depend on various factors: the song, composer or arranger style, and the musicians current intentions (since they can change song harmony - e.g., by chord substitution). Moreover, each song has its particularities, no matter the style, composer, arranger, etc. Despite the lack of universal prediction rules to explain the chord chaining of the whole song, some short chord sequence are recurrent (e.g., II-V-I, II-V). This fact motivated the current approaches to employ different machine learning algorithms (Mitchell 1997), which can infer, from a given set of songs, rules for identifying these recurrent chord sequences (Thom & Dannemberg 1995).

The next issue to consider is the representation of musical context. In fact, even using a learning algorithm, it is necessary to determine how to represent the learning examples. How many previously played chords must be considered to predict the current one? How to represent the chord? Which attributes of a chord (root pitch, structure -major, minor, etc-, interval with previous chord, position within the song, etc.) are actually relevant to its definition? The success of the learning process strongly depends on adequate choice of a representation of the chord (Cunha & Ramalho, 1998).

The final issue is how to combine prior and on-the-flight knowledge, as discussed in section 1. Many peculiarities and recurrent structures of a song, such as refrain, chorus and stanzas, must be extracted in real time, since some important information is hidden within the global structure of each song. On the other hand, the general aspects of songs structure identified by the machine learning process on the song examples cannot be ignored. There are many algorithms that can be used to learn prior knowledge from examples (Mitchell 1997). However, most of them cannot adapt the previously acquired knowledge to fit to the structure of the song being played. For instance, in the chord sequence showed in the figure 2, it is quite unlikely to predict that the chord Bbm7 would follow the G6, since it is an unusual modulation. Nevertheless, when this passage will be repeated over and over, one can expect that musicians will play Bbmin7 at the right moment.

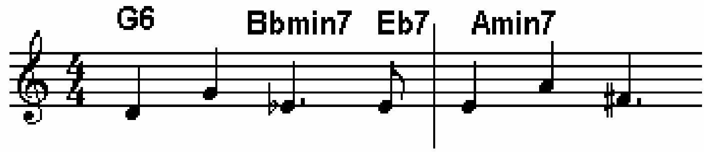

Figure 2 - Passage of "Darn That Dream"

3 State of the Art

Some research efforts have been devoted to prediction of musical parameters (Dirst & Weigend 1994; Hörnel & Ragg 1996; Rowe 1993). In the case of tonal music chord prediction, one of the most relevant works was developed at the Carnegie Mellon University (CMU) (Thom, 1995). They used a learning algorithm based on the *n-gram* models (Bell *et al.*, 1990) to perform real time chord prediction in jazz songs context. The basic idea is to estimate, by training, the probability of the occurrence of a chord, given its antecessors. Although the reported results were not satisfactory, Thom and Dannemberg discuss this problem and propose an elegant model for combining theses two sources of knowledge.

Since *n-gram* models technique provides both off-line and incremental (on-line) learning, the implemented system can employ three functioning modes. In the off-line mode, the system uses the knowledge acquired by training on a set of about 30 songs to predict chords of an unknown song. In the on-line mode, the system starts from scratch and acquires knowledge while trying to predict chords of an unknown song. In the last one, the mix mode, both on-line and off-line learning are combined. In this mode, the systems starts with a certain knowledge and refines/extends it according to the particularities of the song whose chords are being predicted.

The tests done by the CMU team reached 42% to 53% of right answers on a corpus composed by about 30 songs, all in the same tonality. These unsatisfactory results may be perhaps explained by the fact that a poor representation of chords song context was used and that no information about melody was considered (to a detailed discussion on this, see Cunha & Ramalho 1998). Despite the low prediction rate, the CMU contributed by showing that the best results were achieved always in the mix (on-line + off-line) mode.

The model we proposed is based on a neural network model and a richer representation of chords and songs (Cunha & Ramalho, 1998). We used a Multi-layered Perceptron (MLP) Model, with backpropagation learning algorithm. The results we obtained, on a *corpus* of more than 60 jazz' songs in different tonalities, were quite good: 85% of right predictions in average.

These is really a great improvement in results with respect to CMUs work, but the tests showed that results could be even better if our model could adapt its behavior on the flight. In fact, the nature of the neural network used was totally deterministic, in the sense that the same entry would always produce the same output answers, and obviously, the same errors. In other words, it was not possible to undergo on-line adaptations.

This is due to the fact that it is hard to implement on line adaptation the learned connections weight. As we describe in the following section, we decided to include in our system a sequence tracker that could work in competition with the neural network. When the tracker is sure about what will be the next chord, it assumes the control of the prediction system. In the other cases, the neural network performs the prediction.

4 THE CHORD PREDICTION MODEL

In this section we describe a hybrid model composed by a neural net predictor (MLP-backpropagation) which learns and uses prior knowledge, and a sequence tracker, which analyses the structure of each song in real time to pursue the recurrent structures.

4.1 The sequence tracker

Jazz' songs are formed by well-defined blocks, or chords sequences, that, in many cases, repeat themselves along the song, sometimes completely, sometimes with slight differences. Each of these chord blocks forms the structure of the jazz songs and are informally called sections "A", "B", "C" or "D" (the most common structure being the 32AABA, composed by sections of 8 measures). For example, a song can begin with a chords' section called "A", followed by one section called "B". Next, it can be found another section with the same harmonic structure of section "A" followed by another section "B". A system capable of identifying when a block "AB" will be repeated, for example, would not need to ask to the machine learning module to try to reach the right answer. We could pass the control of the prediction to a rule system which would identify the block repeating the sequence "AB" already played in the song. Besides section repetitions, the song as a whole is usually repeated many times, according, in general, to the following scheme: one theme exposition chorus, "n" improvisation choruses and one final theme exposition chorus. These repetitions also should be capture by a chord predictor algorithm.

From these principles, we have tried to define rules for real time identification of these blocks or sequences of chords within a song (including all of its repetitions). We have analyzed the structure of about 30 jazz standards, in order to extract sequence tracking rules, i.e., rules that determine when a given chord sequence is in fact a repetition of a previously played one. The rules guide the process of pattern matching, which is continually done in order to detect relevant repetitions.

The sequence tracking process takes into account the melody, and not only the chords. This is an essential point for the tracker success and this is also an innovation with respect to previous works, including ours. Considering the melody is actually a natural design decision, since the melodic information, as well as the associated lyrics (when it exists), is extremely useful for musicians in a prediction task.

The main sequence tracking rules and strategies are the following

1. One sequence is a block of a song including its melody and chords;
2. The whole song is a sequence;
3. Each part of a song that is not repeated in any other part of the same song is considered a sequence;
4. One can only guarantee that a chord sequence is starting to be repeated after the co-incidence of at least three consecutive measures;
5. Two consecutive chords or measures cannot be tested as belonging to different sequences because this can generate a loop;
6. Normally, the sequences' blocks have a quantity of measures that is a multiple of eight;
7. In many cases, repeated sequences inside of a song are not completely repeated, presenting some differences in the measures 8n or 8n+1, where n=1,2,3.... In this case the sequence tracker pass the control to the neural network ;
8. Each sequence must be tested until the beginning of the next sequence is found, to avoid loops.

Figure 3 shows an example of "Basin Street Blues" chord grid. Because of rule 5, the tracker will not try to interpret the measure 2 (or measure 10) as the repetition of measure 1 (or measure 9). On the other hand, the tracker will suspect that the sequence beginning at measure 9 is the repetition of that starting at the first measure (if the respective melodies also matches - rule 1). From the measure 12 (rule 4), the tracker is sure that there is such a repetition and, then, assumes the control of the whole prediction system repeating the chord of the sequence 1-8. However, according to rule 7, the tracker will not try to play the chords (Bb F7) of the final measure of the tracked sequence. Then, at measure 16, the control goes back to the neural network predictor.

1	Bb	Cm7	C#dim	Bb	Bb	Cm7	C#dim	Bb

3	Bb	Dbm6	Cm7	F9	Bb	Dbm6	Cm7	F9
5	Bb		Bb7		Eb6		Ebm6	
7	Bb				Bb		F7	
9	Bb	Cm7	C#dim	Bb	Bb	Cm7	C#dim	Bb
11	Bb	Dbm6	Cm7	F9	Bb	Dbm6	Cm7	F9
13	Bb		Bb7		Eb6		Ebm6	
15	Bb				Bb			

Figure 3: Basin Street Blues chord grid (two measures per line)

4.2 The hybrid model

The hybrid model we propose works as a competitive system between a neural network predictor and a rule-based sequence tracker (Cf. Figure 4). The sequence tracker monitors the chord stream and when a sequence repetition is detected, the tracker assumes the control of the whole prediction system, indicating which will be the next chord. When the tracker is not sure that a sequence is being repeated, or when the tracked sequence finishes, then the tracker returns the control of the execution to the neural network. The tracker is, thus, only used when the prediction is 100% guaranteed. Otherwise, it is preferable to rely on the neural network predictions, which are near to 90% of right answers.

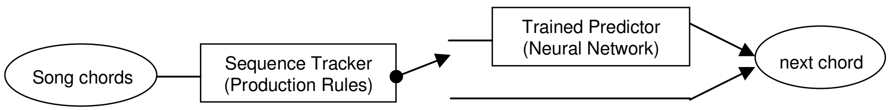

Figure 4. Hybrid Model Scheme

5 IMPLEMENTATION AND RESULTS

The system was developed in Borland Delphi 3.0 (Delphi 1997). The sequence tracking rules have been translated into commands in procedural form. The neural network have been tested with the Qnet97 software (Qnet97 1997), and implemented in Delphi. When the tests were concluded we created an integrated environment for chords prediction that runs on Windows 95/98 platform. With a user-friendly interface, the software provides visualization of the whole process of prediction in real time (cf. Figure 5).

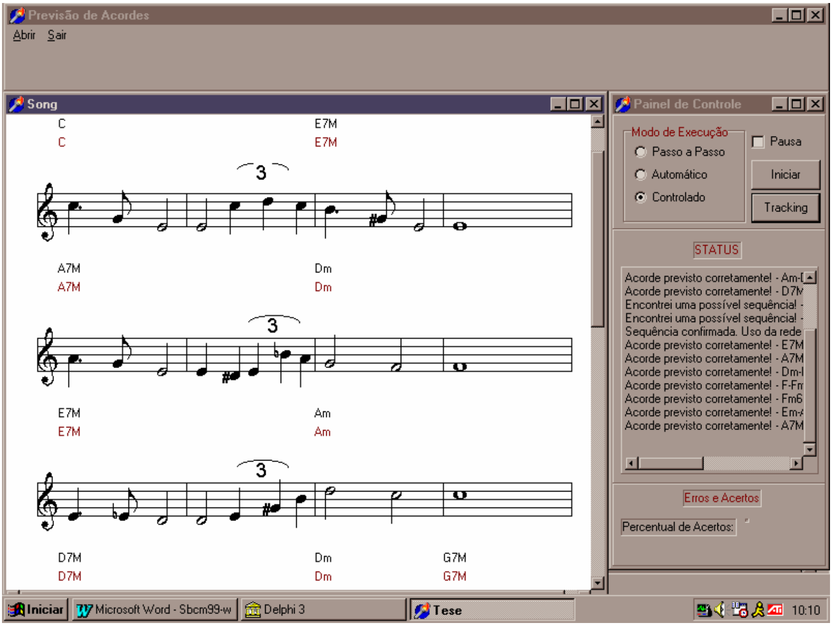

Figure 5. Main window of our prediction system

The experimental evaluation have been undergone with about 60 jazz standards in different tonalities. The results obtained with the hybrid model are better than those using just a neural network to solve the problem (Table 1). Furthermore, the tests have shown that the more the song is repeated, the lower is the error rate of the hybrid model. These results show that this model can capture, on the flight, new knowledge along the time.

Rate Error with a Neural Network only	Rate error of the hybrid system (Song played once)	Rate error of the hybrid system (Song played twice)	Rate error of the hybrid system (Song played three times)
12%	8.8%	5.1%	2.2%

Table 1- comparison of results using the hybrid model

6 CONCLUSION

We proposed an original model of chord predictor, which can combines prior knowledge with on-line adaptation. The reached results motivate us to develop chord prediction systems beyond the jazz style, that will be the basis of projects of real-time accompaniment systems

In spite of the good results, we are aware of the complexity of this problem, and of the necessities of new and more detailed analysis of the best ways to follow from now. We intend to continue our research trying to extend our model to the prediction of other musical parameters.

7 REFERENCES

Bell, T. Cleary, J. & Witten, I. (1990). *Text Compression*. Prentice Hall Advanced References Series. New Jersey: Prentice Hall.

Cunha, U. & Ramalho, G (1998) Previsão de Acordes em Músicas Tonais. In proceedings of *V Brazilian Symposium on Computer Music* - SBC&M'98, pp. 45-54. Belo Horizonte: SBC ed.

Delphi (1997) Delphi 3.0. Inprise Corporation,100 Enterprise Way Scotts Valley, CA 95066-3249, available in http://www.borland.com/delphi

Dirst, M. & Weigend, A.S. (1994) Baroque forecasting: On completing J.S. Bach's last fugue. In A.S. Weigend & N.A. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past* (pp. 151–172). Reading, MA: Addison-Wesley.

Hörnel, D. & Ragg, T. (1996) Learning musical structure and style by recognition, prediction and evolution. In D. Rossiter (Ed.), *Proceedings of the 1996 International Computer Music Conference* (pp. 59–62). San Francisco: International Computer Music Association.

Mitchell, T. (1997) *Machine Learning*. New York: McGraw-Hill

Mozer, M. C. (1994). Neural network music composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing. *Connection Science*, 6, 247-280.

Qnet97 (1997). Qnet97. Vesta Services, Inc. 1001 Green Bay Rd., STE196 Winnetka, IL 60093 - USA, available in http://www.qnet97.com

Ramalho, G. (1997) Pattern Reuse in Tonal Music Improvisation and Accompaniment Systems. In proceedings of *IV Brazilian Symposium on Computer Music* - SBC&M'97, pp 47-58, Brasilia: SBC ed.

Rowe, R. (1993). Interactive Music Systems. Massachusetts: The MIT Press.

Rumelhart, D. & McClelland, L. (1986). *Parallel Distributed Processing*, volume 1. Massachusetts: The MIT Press

Russel, S. & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach.* Englewood Cliffs: Prentice Hall.

Thom, B. & Dannemberg, R. (1995). *Predicting Chordal Transitions in Jazz: The Good, the Bad, and the Ugly.* VI Workshop on Music and AI. 14th International Joint Conference on Artificial Intelligence (IJCAI-95) Montréal

Weigend, A. (1993). *Time Series Prediction; Forecasting The Future And Understanding The Past.* London: Addison Wesley Pub Co.