MPSG. A unified view of Markov chains and decision trees[1]

José Luis Triviño-Rodriguez trivino@lcc.uma.es
Rafael Morales-Bueno morales@lcc.uma.es

Dept. Languages and Computer Sciences, University of Málaga,
Campus Teatinos, 29071, Málaga, Spain

Abstract. A variable memory length multiattribute Markov chain is described. This model is called MPSA (*Multiattribute Probabilistic Suffix Automata*). The computational complexity of learning an MPSA depends exponentially on the number of attributes; so, instead of trying to learn a complete MPSA, we describe a method of learning sequences generated by one attribute when the sequences generated by the rest of the attributes are known. The model used as the learning hypothesis is called MPSG (*Multiattribute Prediction Suffix Graph*). To show why the ability to follow just one attribute is useful, given the others, this model has been applied to automatic music generation and part-of-speech tagging. A public demonstration of this application can be found on the world wide web[2].

1 Introduction

Markov chains and *decision trees* are two of the most important models of machine learning. These two models have many practical applications and they have been applied to a wide variety of problems. However, they are very differents between them.

Decision trees have been described by Quinlan (Quinlan, 1986) and Breiman (Breiman, Friedman, Richard, & Charles, 1984). They have been applied mainly to clasification problems. In this kind of problems, classification trees are applied to determine the value of an attributed of an object from the values of the rest of atributes of the object. Like Markov models, decision trees have applied to natural language procesing such as part of speech taggin (Black, Jelinek, Lafferty, Mercer, & Roukos, 9992) and morfological analisys (Triviño, 1995).

Markov chains (Shannon, 1951) have been applied to model data sequences that exhibit the *short memory* statistical property. If we consider the (empirical) probability distribution on the next symbol given the preceding subsequence of some given length, then exists a length L (the *memory length*) such that the conditional probability distribution does not change substancially if we condition it on preceding subsequences of length greater than L. This feature can be

[1] This work has been partially supported by the MOISES project, number TIC2002-04019-C03-02, of the MCyT, Spain

[2] http://www.lcc.uma.es/˜ trivino

find in may applications related with natural language procesing such as speech recognition (Jelinek, 1985), (Nadas, 1984), and part of speech tagging (Brill, 1994), (Merialdo, 1994).

An improved model of Markov chains has been developed by Dana Ron (Ron, 1996) in 1996. This model is a subclass of PFAs (*Probabilistic Finite Automatas*) called PSAs (*Probabilistic Suffix Automatas*). A PSA is hence a variant order L Markov chain, in which the order, of equivalently, the memory, is variable. Unlike Markov chains, this model did not grow exponentially with its order, and hence longer order models can be considered. Moreover, it produces more intuitive descriptions of real problems.

However, real problems arise where several attributes must be considered simultaneously. An example of this kind of problem is the POS tagging of Spanish texts. Using a feature structure set of tags is advantageous when the available training corpus is small and the tag set large, which can be the case with morphologically rich languages such as Spanish. L. Kempe (Kempe, 1994) and L. Marquez (Márquez & Rodríguez, 1995), (Marquez, 1999) have described how to take advantage of a feature structure set of tags for POS tagging using decision trees. In (Florian & Ngai, 2001) and (Caruana, 1997) there is an excellent discussion about this problem and how Multidimensional Transformation-Based Learning could be applied to it. In this sense, MPSAs are to Markov chains as Multidimensional Transformation-Based Learning is to Transformation-Based Learning (Brill, 1994).

The only way to apply Markov chains to this kind of problem is by using the Cartesian product of alphabets of the problem's attributes as the alphabet of the Markov chain. However, the length of the Markov chain grows exponentially with the number of attributes. Moreover, the cardinal of the alphabet grows significantly. This results in a large number of states which decreases the number of samples used to compute every probability in the model; thus, it decreases the robustness of the model.

Using PSAs to model this kind of problem also has exponential complexity. On the one hand, PSAs share the same problems as Markov chains. On the other, PSAs cannot handle different memory lengths for every attribute needed to describe the model. This is because not all the attributes need the same memory length; however, if the Cartesian product of alphabets of attributes is used, all attributes must have the same memory length.

In order to consider this kind of problem, we define a variable memory length multiattribute Markov chain. This model is called MPSA (*Multiattribute Probabilistic Suffix Automata*). Thus, the PSA model is a subclass of the MPSA model. The MPSA model independently computes the memory length of every attribute needed to determine the next symbol probability distribution.

The computational complexity of learning an MPSA depends exponentially on the number of attributes, so the MPSA model could be considered hard to learn. However, let us consider a independence relation with respect to the values of attributes of an MPSA such that the value of an attribute only depends on the previous values of all attributes. This independence relation is reasonable. With this independence relation the MPSA can be modeled as a set of MPSGs, an MPSG for every attribute of the MPSA. An MPSG computes the next symbol probability distribution of a target attribute given the previous values of all the attributes of an MPSA. In figure 1, the difference between a Markov chain and a MPSG is shown.

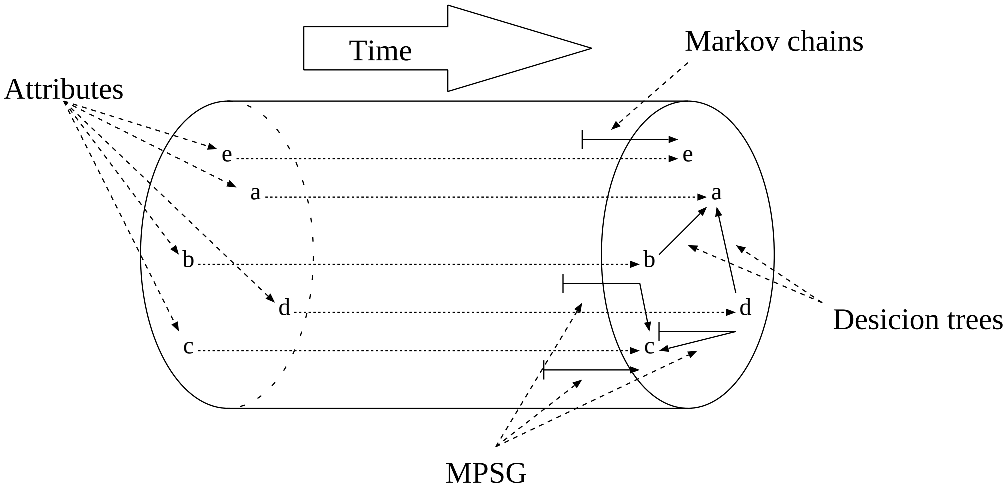

Fig. 1. The state of a system defined by means of several attributes. Markov chains model the evolution of an attribute; decision trees usually model the relation between different attributes at the same time; and MPSGs model the evolution of an attribute from several attributes of the system.

The advantage of using the MPSG model as a learning hypothesis is that the MPSG model can be learned in polynomial time over the number of attributes. Moreover, if state transitions of the MPSG are restricted after learning the model, then at the same time the independence relation between the attributes of the MPSG could be considered in many practical applications of the MPSA model.

In this paper, a unified model of Markov chains and decision tress is described. This model is called MPSG (Multiattribute Prediciton Suffix Graph) and it is equivalent to a variable memory multiattribute Markov chain. This model allows to analize independtly the memory length in every attribute needed to know the next symbol of a target attribute. Due to this model computed every attribute in an independent manner, the number of samples used to computed next symbol probabilities do not decrease very much. So the confidence of the model is bigger.

On the one hand, when this model is cosidered over a set of one attribute, the model is the same that the Ron's PST model (Ron, Singer, & Tishby, 1996). It is different from a PST in the learning algorithm approach. The Ron's learning algorithm follows a top-down approach since it start with a tree consisting in a single root node and incrementally grow the tree in contrast to the MPSG learning algorithm that follows a bottom-up approach. On the other hand, when the memory length of the model is 1, the hypothesis model is equivalent to a decision tree.

The MPSA model and its learning hypothesis, called MPSG, have been successfully applied to POS tagging (Triviño & Morales, 2001a) and music prediction and generation (Triviño & Morales, 2001b), (Triviño, 2000).

Section 2 describes the MPSA model. Later, in section 3, the model used as the learning hypothesis is described. This model is a directed graph (with several restrictions in its layout) and a set of labels. This model is called MPSG (Multiattribute Prediction Suffix Graph), and the PST model (Prediction Suffix Tree model) (Ron, 1996) is a subclass of this model. Finally, two practical applications of the MPSG model are shown.

2 MPSA: Multiattribute Probabilistic Suffix Automata

This section describes the MPSA model. Like a Markov chain, an MPSA is basically a set of states and relations between them. These relations between states, or equivalently, nodes, are the transition function of the MPSA. That is, if the evolution of a system is modeled by means of an MPSA, then given the state of the system computed from the last symbols of the sequence of every attribute and the next symbol for every attribute of the system, the transition function computes the next state of the system.

If we consider the attributes of an MPSA as a set of random variables, then a state in an MPSA denotes the last values of these variables and the edges of an MPSA allows us to compute the next state of an MPSA given the previous state and the next symbol of every attribute. Thus, unlike a belief network (Saul, T.Jaakkola, & Jordan, 1996), the edges of an MPSA represent the transition function of a multiattribute Markov chain instead of causal inferences between random variables. In figure 2 an example of an MPSA is shown.

An MPSA could be described basically as a PSA where every state is defined as a set of strings each one over a different alphabet (every string denotes the last values of an attribute). Hence, a transition in an MPSA is defined by a tuple of symbols (a symbol for every attribute). Thus, the number of transitions from a state in an MPSA depends exponentially on the number of attributes. However, if an independence relation between the values of the attributes is considered, then the probability distribution of the next symbol of every attribute in an MPSA can be computed separately.

Moreover, an MPSA allows us to efficiently compute the next state given the previous state and the next symbol for every attribute. However, it does not show the relation between the memory length of every attribute and the next symbol probability function needed in the learning task. Hence, an MPSA is difficult to learn directly. Instead, section 3 describes a model, called an MPSG,

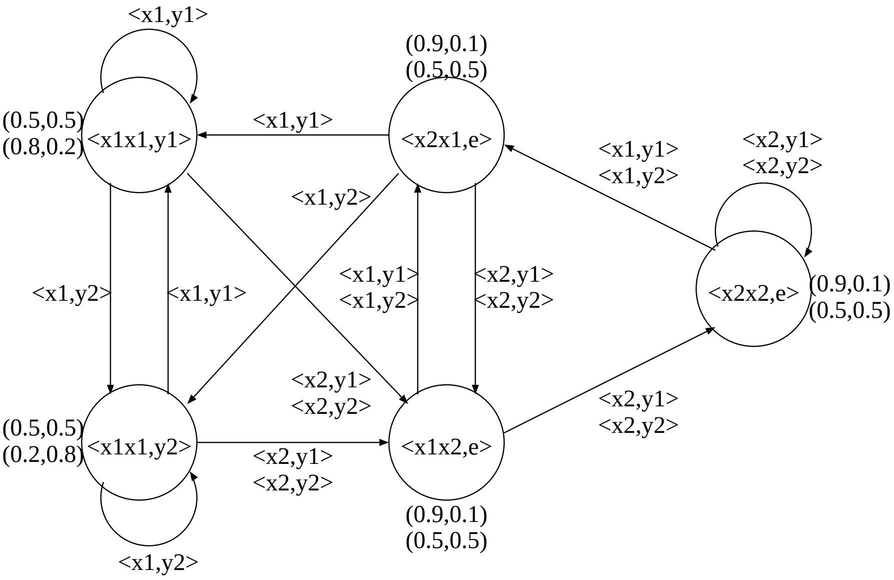

Fig. 2. Example of an MPSA. The next symbol probability functions of attribute X (upper) and attribute Y (lower) are depicted in parenthesis beside each node.

that allows for the computation of the memory length needed for every attribute that determines the next symbol probability distribution.

3 MPSG: Multiattribute Prediction Suffix Graph model

The MPSG model constitutes the learning hypothesis of the MPSA model. Informally, an MPSG is a set of states and relations between them with the structure of a directed graph. The structure of an MPSG shows how the memory length of every attribute has been expanded progressively along the learning task to compute the minimum memory length of every attribute needed to determine the next symbol probability distribution. Hence, the relations between the states of an MPSG are the edges of the MPSG. Every edge shows how the memory length of an attribute has been expanded in the learning task.

A multiattribute prediction suffix graph (MPSG) G is a directed graph where each node in the graph is labeled with a state G. These states represent the memory that determines the next symbol probability function. The node labeled with n empty strings is always in the graph. This node expresses the next symbol probability distribution independently of any context. Nodes are joined by edges labeled by pairs (σ, i) where $\sigma \in A$ and $i \in \mathbf{N}$. These pairs allow us to efficiently find every state in the graph.

The MPSG model could be considered as a multiattribute extension of the PST model of Ron (Ron, 1996). Thus, an MPSG represents the next symbol probability distribution of an attribute. Hence, there exists a unique state in an MPSG that represents the next symbol probability distribution given previous sequences of symbols for every attribute. This state is called the *most representative node* of the sequence in the MPSG. In a PST, this state if the longest suffix of the sequence in the PST.

4 Experimental results

This section describes two practical application of the MPSG model. Section 4.1, describes how MPSGs can be applied to Spanish part-of-speech Tagging (Triviño & Morales, 2001a). Next, in section 4.2, MPSGs are applied to music generation (Triviño & Morales, 2001b).

4.1 Using MPSGs for POS Tagging

Many words in Spanish function as different parts of speech (POS). Part-of-speech tagging is the problem of determining the syntactic part of speech of an occurrence of a word in context. Because most of the high-frequency Spanish words function as several parts of speech, an automatic system for POS tagging is very important for most other high-level natural language text processing.

The MPSG model has been applied to part-of-speech tagging. This tagger uses a feature structure set of tags like the model described by Kempe (Kempe, 1994) for POS tagging. The order L of the MPSG tagger has been set to 4. The tagger has been trained on a Spanish 45000-word tagged corpus.

The accuracy obtained is 97.32%. This is not the best accuracy achieved by a Spanish tagger (Triviño & Morales, 2000). However, it must be taken into account that the tagger has been trained with a small sample. This tagger is more complex than the VMM tagger, so it needs a larger number of words to perform the training task. Moreover, many of the mistakes are due to incorrect tags in the training sample. In Table 1 a comparison between the accuracy of several taggers is shown.

Tagger	Language	Corpus (words)	Accuracy (%)
Triviño MPSG tagger	Spanish	45000	97.32
Triviño VMM tagger	Spanish	45000	98.58
Padró	Spanish/English	10^6	97.45
Charniak	English	10^6	96.45
Kempe	French	$2 * 10^6$	96.16
Brill	English	350000	96.0
Singer	English	10^6	95.81
Kempe	French	10000	88.89

Table 1. Comparison between several taggers

Our results (97.32%) are better than Singer's tagger (95.81%) based on VMM because the single word tagger has added lexical information to our tagger. This can decrease the error rate when errors due to bad tags for rare words are avoided by the single word tagger. However, it is difficult to compare these results with other works, since the accuracy varies greatly depending on the corpus, tag set, etc. Performance could be measured more precisely by training the system on a larger corpus.

4.2 Using MPSGs to generate music

Let us consider the following goal: learn musical pieces in a specific style and generate new musical pieces in the same style. To reach this goal, an initial step was to define the attributes of the model. We used the multiple viewpoint system described by Witten (Conklin & Witten, 1995). Witten's terminology denotes as a viewpoint every attribute of the model. Every viewpoint models a type t (or abstract property) of events in the sequence. Thus, a viewpoint comprises a partial function $\Psi_\tau : \xi^* \rightarrow [\tau]$ and a model to predict the next symbol in sequences in $[\tau]^*$, where ξ is the set of valid events, and $[\tau]$ denotes the set of all syntactically valid elements of type τ.

An MPSG could be used as the prediction model of sequences in an attribute. In this way, the set of all types t in the multiple viewpoint system corresponds to the set of attributes A of the MPSG and, for every type τ, $[\tau]$ defines the set of symbols of the equivalent attribute. Thus, an event could be described as a vector with n components, each one formed by a symbol of the alphabet of a different attribute.

For example, pitch and duration are two attributes that could be considered in order to model musical events. *Pitch* is the pitch of the event represented as an integer value in the range from 60 (C4 or middle C) to 75 (G5, 19 semitones above middle C), according to the MIDI standard; and *duration* is the duration of a event measured in sixteenth notes. We need two MPSGs to model these attributes: an MPSG for every attribute. Both MPSGs must have two attributes (pitch, duration) because every MPSG computes the next symbol probability distribution of one attribute conditioned by all the attributes of the problem.

To apply MPSGs to music generation, two MPSGs were trained with one hundred Bach chorales from The 371 Four-Part Chorales (Mainous and Ottman, 1966). These chorales were used by Witten (Conklin & Witten, 1995) for music prediction/generation. The data files used by Witten consist of only one voice (the melody) of Bach's original chorales, so this system has not been tested with polyphonic music. However, the model could represent several voices in polyphonic music modeling by adding an attribute for every viewpoint of every voice (for example, an attribute for the pitch of the first voice and another attribute for the pitch of the second voice). The main problem in learning polyphonic music is that events in this kind of music are asynchronous (i.e., overlapping in time) and MPSGs can only represent synchronous events (i.e., all attributes change their values at the same time). This could be solved by considering the time elapsed between events as described by Assayag (Assayag, Dubnov, & Delerue, 1999).

These MPSGs have been used in the implementation of a random generator of sequences. The sequences of events are converted to MIDI format and can be played on a computer. An online demo of this task is available in the URL http://www.lcc.uma.es/˜ trivino.

A random sample of 100 pieces of 70 notes each has been generated with this program. Although it is difficult to prove that sequences generated by the MPSG follow the same rules as Bach's pieces, use of the Kolmogorov-Smirnov test (Hollander & Wolfe, 1973) has proved that both distributions follow the same short context distribution (the relation modeled by MPSGs).

The Kolmogorov-Smirnov test can prove that the MPSGs can generate sequences with the same probability distributions and context properties as Bach's chorales. However, it cannot show how a human listener perceives the music. In order to evaluate the distance between the generated music and Bach's chorales, we performed an auditory test where the subjects classified fragments coming from the original chorale, and other ones coming from MPSG simulation. This test has shown that the melodies generated by the MPSG model are hard to recognize from the original Bach's melodies.

5 Conclusions and future work

A multiattribute variable memory length Markov chain model has been described in this paper. This model is called MPSA (Multiattribute Probabilistic Suffix Automata) and has been developed as a generalization of the PSA model (Probabilistic Suffix Automata) described by Ron (Ron, 1996). Thus, Dana Ron's model can be viewed as an MPSG with only one attribute.

Moreover, the equivalence between Markov chains, decision tree and MPSGs has been studied. This analisys has prooved that the MPSG model is a unified view of Markov chains and decision trees. So, the MPSG model combines the learning of time sequences with the inductive learning from several attributes.

The main development line of this work is to applied several techniques used in desicion tree learning to MPSG learning. These techniques include learning from unknown values, attributes with continous values, domain information and adaptative learning sampling.

Bibliography

Assayag, G., Dubnov, S., & Delerue, O. (1999). Guessing the composer's mind: Applying universal prediction to musical style. In *Proceedings of the ICMC*.

Black, E., Jelinek, F., Lafferty, J., Mercer, R. L., & Roukos, S. (9992). Decision tree models applied to the labeling of text with parts-of-speech. In *Darpa Workshop on Speech and Natural Language*, Harriman, N.Y.

Breiman, L., Friedman, J., Richard, O., & Charles, S. (1984). *Classification and regression trees*. Wadsworth and Brooks.

Brill, E. (1994). Some advances in transformation-based part of speech tagging. In *Proceedings of AAAI94*, p. 6.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*(1), 41–75.

Conklin, D., & Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*(1), 51–73.

Florian, R., & Ngai, G. (2001). Multidimensional transformation-based learning. In *Proceedings of CoNLL'01*, pp. 1–8.

Hollander, M., & Wolfe, D. (1973). *Nonparametric Statistical Methods*. Wiley, New York.

Jelinek, F. (1985). Self-organized language modeling for speech recognition. Tech. rep., IBM T.J. Watson Research Center.

Kempe, A. (1994). Probabilistic tagging with feature structures. In *Coling-94*, Vol. 1, pp. 161–165.

Marquez, L. (1999). *POS Tagging: A Machine Learning Approach based on Decision Trees*. Ph.D. thesis, Dept. LSI. Universitat Politècnica de Catalunya (UPC), Barcelona.

Merialdo, B. (1994). Tagging English text with a probabilistic model. *Computational Linguistics*, *20*(2), 155–171.

Márquez, L., & Rodríguez, H. (1995). Towards learning a constraint grammar from annotated corpora using decision trees. ESPRIT BRA-7315 Acquilez II, Working Paper.

Nadas, A. (1984). Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Trans. on ASSP*, *32*(4), 859–861.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Ron, D. (1996). *Automata Learning and its Applications*. Ph.D. thesis, MIT.

Ron, D., Singer, Y., & Tishby, N. (1996). The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, *1*, 34.

Saul, L., T.Jaakkola, & Jordan, M. (1996). Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, *4*, 61–76.

Shannon, C. (1951). Prediction and entropy of printed English. *Bell Sys. Tech Jour.*, *30*(1), 50–64.

Triviño, J. (1995). SEAM. Sistema experto para análisis morfológico. Master's thesis, Universidad de Málaga.

Triviño, J. (2000). Modelado y generación de música mediante autómatas sufijos probabilísticos multiatributo (MPSA). I premio de la III convocatoria de los premios Severo Ochoa de Ciencia y Tecnología del Ateneo de Málaga.

Triviño, J., & Morales, R. (2000). A Spanish POS tagger with variable memory length. In *(IWPT 2000) Sixth International Workshop on Parsing Technologies*, Trento (Italia).

Triviño, J., & Morales, R. (2001a). Using multiattribute prediction suffix graphs for part-of-speech tagging. In *International Symposium on Intelligent Data Analysis*, Cascais (Portugal). Lecture Notes in Computer Science.

Triviño, J., & Morales, R. (2001b). Using multiattribute prediction suffix graphs to predict or generate music. *Computational Music Journal*, *1*.