Prediction suffix trees for supervised classification of sequences

Christine Largeron - Leténo

EURISE - Université Jean Monnet Saint-Etienne
6, rue Basse des Rives
42023 Saint-Etienne cedex 2
Tel : (33) 04 77 42 19 60
Fax : (33) 04 77 42 19 50
largeron@univ-st-etienne.fr

ABSTRACT: *This article presents a statistical test and algorithms for patterns extraction and supervised classification of sequential data. First it defines the notion of Prediction Suffix Tree - PST. This type of tree can be used to efficiently describe Variable order chain. It performs better than the Markov chain of order L and at a lower storage cost. We propose an improvement of this model, based on a statistical test. This test enables us to control the risk of encountering different patterns in the model of the sequence to classify and in the model of its class. Applications to biological sequences are presented to illustrate this procedure. We compare the results obtained with different models (Markov chain of order L, Variable order model and the statistical test, with or without smoothing). We set out to show how the choice of the parameters of the models influences performance in these applications. Obviously these algorithms can be used in other fields in which the data are naturally ordered.*

KEY WORDS *: Prediction suffix tree (PST), Patterns extraction, Supervised classification, Variable order chain, Markov model, Chronobiological and DNA sequences.*

1. Introduction

Sequences modelling can have applications in a great number of fields such as biology, robotics or even, in speech recognition or text mining. By sequence, we must understand successive observations, which might, depending on the case under study, be a protein or a DNA sequence, a series of sound signals or even the words of a sentence. Among the models proposed to represent such sequences, we can distinguish on the one hand, non probabilistic models such as deterministic and non deterministic finite automata, or even different families of grammars and, on the other hand, probabilistic models such as stochastic automata, N-grams, Markov chains of order L or Hidden Markov Models (HMM).

The concept of the Markov chain of order L, which we essentially owe to the Russian mathematician Andrej Andreevic Markov (1907), has two drawbacks. First, the number of

parameters of the model grows exponentially with the order L of the chain. This brings about computational and storage problems during implementation, including for limited memory length L. An improvement initially put forward by (Rissanen - 1983) and used particularly in compression data (Weinberger - 1992, Willems - 1995) was the Variable Length Markov chain (Buhlmann - 1999). This model can be represented by a tree, known as Prediction Suffix Tree – PST (Ron - 1996), certain branches of which are depth L and others of an inferior depth to L, whereas the Markov chain of order L corresponds to a complete tree of depth L. By reducing the storage cost, pruning the branches of the tree will enable us to increase the order of the model and, thereby improve performance.

The second drawback of the Markov chain of order L lies in the difficulty of estimating the parameters of the model in general, and in particular when there is a sub-sequence which has not been observed in sequences of the training set but is likely to appear in sequences to classify. Concretely, this implies that zero probability is affected to these events which will alter the performance of the model during the prediction phase. The proposed solution for improving generalisation performance is probability smoothing. Smoothing techniques consists in allocating reduced but non-zero probability to these events (Henikoff - 1996, Katz - 1987).

By combining probability smoothing and variable memory length L, a model is obtained which provides excellent results particularly in bio-informatic for the modelling of protein families (Bejerano - 2001, Kermorvant - 2002). However, in the context of supervised classification, there is no guarantee that the sequence to classify will not include sub-sequences (sub-chains, domains, patterns, …) which are not in the model of its predicted class and conversely. It is for this reason that we propose an improvement of the Variable order chain based on a statistical test. This test may be used as a complement to the Variable order chain to compute the decision risk that the probability of a sub-sequence in the model of a sequence to classify is equal to this probability in the model associated to its predicted class when it is not true. Moreover, in Markov chain of order L or in Variable order chain, the quality of the results depends largely upon the smoothing technique employed as well as the parameters of smoothing (Kermorvant - 2002b). The second point of interest of this test is that it can be employed directly as a decision rule, and from the experiments which we carried out, it gives us results, which generally speaking, are roughly identical with or without probability smoothing.

In this article, the problem of classification of sequences is described briefly in the second section, as well as the concept of the Prediction Suffix Tree (PST) used to represent the sequences. Examples on artificial data and on E. coli DNA data from the UCI repository of machine learning database (http://www.ics.uci.edu/~mlearn/MLRepository.html) are provided in order to illustrate this section. We will propose the statistical test, followed by algorithms of comparison and of supervised classification of sequences in the third section. An application to chronobiological sequences is then presented to illustrate these procedures. In this last section, we compare the results obtained with the aid of different models (Markov chains of order L, Variable order chain and the statistical test, with or without smoothing) and we show how the choice of parameters of the models influences performance.

2. Preliminaries

2.1 Description of the problem of supervised classification of sequences

Let A be a finite alphabet of size p. By A^*, we denote a set of sequences (or strings) defined over A, where the length of the ith sequence is l_i (with $l_i \leq l$). Let us suppose that A^* is divided into C clusters which constitute a partition of A^*, and that the sequences belonging to a same class were generated by a same unknown underlying process. In the supervised classification context, we have only a sample $E_1 \cup E_2 \ldots \cup E_C$ of A^*, which is representative of the different clusters. The

aim is to define a decision rule enabling us to automatically identify the cluster of any sequence of A^*. This rule will be constructed by using only a part E of the set $E_1 \cup E_2 \ldots \cup E_C$, known as a learning sample; the other part will serve as a test sample and will be used to estimate the error rate of the decision rule.

Therefore, in the example of E.coli DNA sequences, four nucleotides denoted by: **a, c, g, t.** constitute the alphabet A. The database is composed of 106 instances: 53 belong to the class of sequences with biological promoter activity (positive instances) and 53 to the class without promoter activity (negative instances). So, each sequence is a suit of 57 characters nucleotides, and the aim is to predict the class (positive or negative) of any E. coli DNA sequence.

2.2 Modelling of sequence by Prediction suffix tree

If we suppose that in sequences of a same class, the probability distribution on the next symbol of A is dependent on the preceding sub-sequence of some variable length, then the sequences of the class may be modelled by a Variable order chain. This model can be described by a Prediction Suffix Tree (PST). Like this, to any sequence s_i of A^*, defined by: $s_i = (s_{it})$, $t = 0..\ l_i$, (where l_i is the length of s_i : $l_i \leq l$, $\forall\ s_i \in A^*$ and $s_{it} \in A$), is associated a PST (Prediction Suffix Tree) noted as S_i and, defined (in a slightly different form to that introduced by (Rissanen - 1983)), in the following manner:

- S_i is a tree of degree p where p is equal to the size of the alphabet A and ε is the root of the tree
- The root and each internal node is the initial extremity of p edges each corresponding to a distinct symbol of A
- Each node, excepting the root, is labelled by the pair (k , $\varphi^k{}_i$) where
  - k is the string associated with the walk starting from that node and ending in the root of the tree. For practical purposes, k will also denote a node in the tree.
  - $\varphi^k{}_i$ is a p dimensional vector where the component $\varphi^{kj}{}_i$ is the empirical conditional probability of observing the symbol j of A after the string k in the sequence s_i:

$$\varphi_i^{kj} = \frac{n_i^{kj}}{n_i^{k*}} \qquad [1]$$

  where $n^{kj}{}_i$ is the number of occurrences of j after k in the sequence s_i and $n^{k*}{}_i$ is the number of occurrences of any character of A after k in the sequence s_i.

- the root of the tree is labelled by (ε, $\varphi^\varepsilon{}_i$) where $\varphi^\varepsilon{}_i$ is a p dimensional vector and its component $\varphi^{\varepsilon j}{}_i$ is equal to the empirical probability of observing the symbol j of A in the sequence s_i of size l_i:

$$\varphi_i^{\varepsilon j} = \frac{n_i^{\varepsilon j}}{l_i}$$

- r is the maximum number of internal nodes in the tree.

For example, if we consider the alphabet $A = \{a, b\}$ and the sequence $s = (aabaabaabaab)$, then the PST S associated to s (with L = 2) is described in the Figure 1.

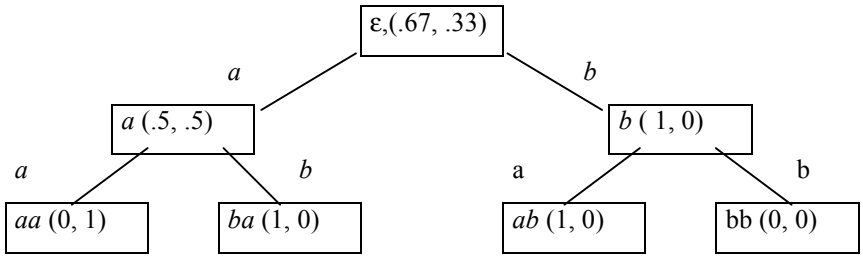

Figure 1. ***PST S associated to s = (aabaabaabaab)***

In the context of supervised classification, each class c (c = 1,..,C) is associated to a PST S_c for which the empirical probabilities are computed over the whole set of sequences of this class belonging to the training set. Classification of a new sequence $s = (s_0, s_1, \ldots, s_t \ldots s_l)$ is then computed directly with the aid of these PST. For each PST S_c , prediction probability for the sequence s is carried out. Its prediction by S_c is done letter by letter $(s_t, t = 0,..,l)$ where the probability of each letter is calculated by scanning the tree in search of the longest suffix that appears in the tree and ends just before that letter. The conditional probability of this letter given this suffix is the empirical probability associated to the node corresponding to this suffix in the PST. In this way, the probability that S_c generates s is given by:

$$P^{S_c}(s) = \prod_{i=1} \varphi_c^{suf\ \max(s_i)s_i} \qquad [2]$$

where *suf(s)* is the suffix of s. For instance, $suf(s) = (s_1, s_2 \ldots s_l)$ if $s = (s_0, s_1, \ldots s_l)$ and *sufmax(s)* is the longest suffix of s that appears in the tree. Finally, the sequence s is assigned to the class c_0 corresponding to the PST S_{c0} for which maximal probability has been obtained:

$$P^{S_{c0}}(s) = Max\left\{P^{S_c}(s), c = 1,.., \mathrm{C}\right\} \qquad [3]$$

For example, using the PST S depicted in Figure 1, the prediction probability of the sequence $s' = (abaab)$ is:

$$\begin{aligned} P^{S}(abaab) &= P(a)\, P(b/a)\, P(a/ab)\, P(a/aba)\, P(b/abaa) \\ &= \varphi^{\varepsilon a}\, \varphi^{ab}\, \varphi^{aba}\, \varphi^{baa}\, \varphi^{aab} \\ &= 0.67 \times 0.5 \times 1 \times 1 \times 1 \\ &= 0.33 \end{aligned}$$

Note that $P(b/abaa)$ is given by φ^{aab} because *aa* is the longest suffix of *abaa* in the PST.

If the maximum number of characters of the string associated to a node of the tree is L then, the PST defined previously is a complete tree with a depth equal to L which corresponds to a Markov chain of order L. A first drawback of this model is that even in the case where a large training set is available and for a reduced order L, numerous conditional empirical probabilities (φ^{kj}_i) associated with nodes in the tree type, risk taking a zero value. This is due to the fact that the sub-sequence [1]

[1] [1] It must be remembered that a sub-sequence (or sub-chain) is connected whereas a sub-word is not necessarily so. Therefore, 1123 is a sub-sequence of the sequence 211231 while 2231 is simply a sub-word.

corresponding to this node was not observed in the sequences of the training set even if it is likely to appear in the sequences to classify. In prediction step, this will quite frequently lead to an indeterminate situation concerning the classification of new sequences. To avoid this, it is possible to carry out smoothing of probabilities. Several methods of smoothing have been put forward such as the position based pseudo-counts method (Henikoff - 1996) or back-off method (Katz - 1987). These consist in allocating low but not zero conditional empirical probability to sub-sequences, which have not been observed in the training set. At the same time, we reduce probabilities allocated to sub-sequences observed in the training set. A second drawback of Markov chains of order L is that the number of parameters grows exponentially with the order L and hence requires high data storage capacity even for low order. But, often in applications, the memory length depends on the context and is not fixed. This observation has led to several improvements. One of them is the Variable Length Markov chain (Buhlmann - 1999). This Markov chain with variable order can be efficiently described by a PST. The construction of the PST follows either a top-down approach in which not all the nodes are inserted in the tree or a bottom-up procedure which consists in pruning the complete tree. The two schemes are equivalent and yield the same PST. Pruning the tree enables us to reduce storage cost, and if necessary, to increase the order L and at the same time, the prediction performance.

An algorithm of PST construction combining these two improvements (smoothing and pruning) was proposed recently (Ron - 1996). In this algorithm, smoothing involves a positive ymin parameter in such a way that every conditional probability $\varphi^{kj}{}_i$ is replaced by:

$$f_i^{kj} = (1 - p \times \text{ymin})\varphi_i^{kj} + \text{ymin} \qquad [4]$$

Here, pruning consists in inserting a node k into the tree S_i only if this node corresponds to a sub-sequence $(k_0k_1k_2...k_l)$ which is sufficiently frequent in the sequence s_i and, if the conditional probabilities of occurrence of the characters of the alphabet $(\varphi^{kj}{}_i\ ,\ j \in A\ )$ following this sub-sequence are significantly different from the conditional probabilities observed in the predecessor of k corresponding to the suffix $(k_1\ k_2....k_l)$ of k. Formally, this means that the following conditions must be checked:

(1) $\varphi^{k}{}_i \geq$ pmin where $\varphi^{k}{}_i$ is the empirical probability of observing the sub-sequence $k = (k_0k_1...k_l)$ in s_i

and for at least a character j of A:

(2) $\varphi^{kj}{}_i \geq (1 + a)$ ymin

(3) $\varphi^{kj}{}_i \geq r\ \varphi_i^{suf(k)\,j}$ or $\varphi_i^{kj} \leq 1/r\ \varphi_i^{suf(k)j}$

Therefore, if we refer back to the preceding example and use as smoothing parameters ymin = 0.001 and as pruning parameters pmin = 0.001, r = 1.05 and a = 0, we obtain, with $A = \{a,b\}$ and L = 2, for the sequence $s = (aabaabaabaab)$, the PST S' of Figure 2.

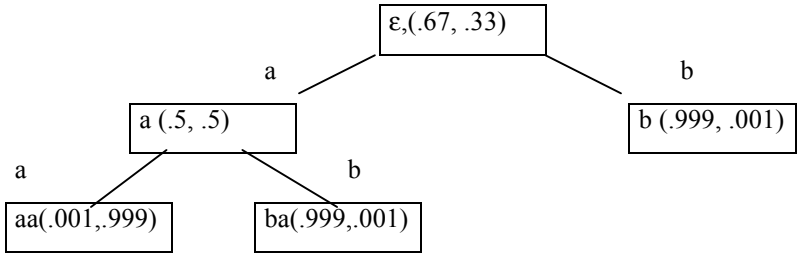

Figure 2. ***PST S' associated to s = (aabaabaabaab)***

From the experiments carried out recently in bio-computing (Bejerano - 2001, Kermorvant - 2002), probability smoothing and pruning would seem to enhance this model with a performance which is equivalent to that of HMM at lower storage costs. For example, the size (number of nodes) of the PST built (without smoothing: ymin = 0) for each class of E. coli DNA sequences (PST_+ for positive instances and PST_- for negative instances) varies with the values of r and pmin as shown in Table 1; L and a remaining constant respectively equal 3 and 0. In comparison to the Markov model of order L (pmin =0 and r= 1), pruning induces a reduction of the number of the nodes.

pmin	r	size of PST_+	Size of PST_-
0	**1**	**84**	**84**
0	**1,2**	78	83
0	**2**	21	48
0,01	1	76	75
0,1	1	5	4
0,01	**1,2**	70	74

Table 1. ***Storage costs***

From this benchmark, we tested the performance of several models (Markov chain of order L and Variable order chain, with or without probability smoothing). As smoothing and pruning techniques, we applied those described in (Bejerano – 2001with a =0). Results are presented in Table 2 where each configuration corresponds to a set of different values of parameters. For each configuration, error rate (MC) was obtained using the leaving one out method: MC corresponds to the number of elements wrongly classified over the 106 experiments for the same values of parameters. The result obtained by combining probability smoothing and pruning (III.4) is equivalent to those obtained with Markov chain (III.1) but with a lower storage cost.

configurations	L	pmin	r	ymin	MC (%)
I	**1**	0	1	0	19.81
III.1	**3**	**0**	**1**	**0**	**7.55**
III.2	3	**0.01**	**1.2**	0	9.43
III.3	3	0	1	**0.001**	5.66
III.4	3	**0.01**	**1.2**	**0.001**	7.55

Table 2. ***Comparison of models: error rates***

However, in the context of supervised classification of a new sequence, by introducing probability smoothing and pruning in Markov chain, there is no guarantee that there is no difference between the probability of a sub-sequence in the model of the sequence to classify and in the model of its class. For this reason, in the following section we propose a statistical test, which serves as a complement to the Variable order chain. The aim is to calculate the decision risk that these probabilities are equal whereas there is a significant difference between them. This test may also be used directly as a decision rule for the classification of a new sequence.

3. Comparison of PST and supervised classification of sequences

3.1. Hypothesis.

In this section, we propose a statistical test of comparison of PST and an algorithm of supervised classification of sequences. Our approach assumes that the sequences of a same cluster are generated by the same process. To translate this hypothesis, we consider that in sequences of a same cluster c (c varying from 1 to C), the probability of observing a character j of A after observing the sub-sequence k is equal, for any k varying from 1 to r, whereas in sequences belonging to distinct classes c and c', this probability differs for at least one character of A in at least one of the sub-sequences k. So, for two distinct sequences s_i and $s_{i'}$ belonging to the class c and, for k and k' two sub-sequences respectively of s_i and $s_{i'}$, we have $p^{kj}_{\ i} = p^{k'j}_{\ i'}$ if $k=k'$, for any k varying from 1 to r and for any j belonging to A. While for s_i belonging to c and $s_{i'}$ belonging to c', there is k belonging to $\{1,..,r\}$ and j belonging to A such that $p^{kj}_{\ i} \neq p^{kj}_{\ i'}$ where $p^{kj}_{\ i}$ denotes the conditional probability of observing a character j after the sub-sequence k in the model used for generating the sequence s_i.

3.2. *Comparison of PST*

Given two sequences s_i and $s_{i'}$ of E ($i \neq i'$) and k a sub-sequence of s_i and $s_{i'}$ ($k = 1,..,r$), the aim is to test for any j belonging to A, the null hypothesis of equality of the distributions:

$$\text{H0} : p^{kj}_{\ i} = p^{kj}_{\ i'}$$
$$\text{against} \quad \text{H1} : p^{kj}_{\ i} \neq p^{kj}_{\ i'}$$

H0 is the null hypothesis according to which the characters are identically distributed in the models corresponding to these sequences. A primary solution consists in using the Chi-squared test, frequently employed to compare two percentages in the case of large samples, or Fisher's exact tests for small samples (Fisher R. - 1925, Metha C. - 1983).

However in the Chi-Squared test, it is necessary to estimate $p^{kj}_{\ i}$. In the following paragraph, we will therefore set out another test, equivalent to the Chi-Squared test, which overcomes this drawback.

We will note $\varphi^{kj}_{\ i}$ and $\varphi^{kj}_{\ i'}$ the respective conditional empirical probabilities of observing a symbol j of A right after a sub-sequence k in s_i and in $s_{i'}$ as defined in the paragraph 2.2 (equation [1]). With a view to deciding whether the sequences s_i and $s_{i'}$ belong to the same class (hypothesis H0) or whether there is a significant difference between them (hypothesis H1) for the sub-sequence kj, we test:

$$\text{H0} : p^{kj}_{\ i} - p^{kj}_{\ i'} = 0$$
$$\text{against} \quad \text{H1} : p^{kj}_{\ i} - p^{kj}_{\ i'} = \delta > 0$$

For any sequence s_i it can easily be demonstrated that $n^k_i \times f^{kj}_i$ is the observed value of a variable $n^k_i \times F^{kj}_i$ which has a binomial distribution $B(n^k_i, p^{kj}_i)$ and, which can be approached by a normal distribution if $n^k i \times p^{kj}_i$ and $n^k i \times (1- p^{kj}_i)$ are high enough:

$$F_i^{kj} \to N(p_i^{kj}, \sqrt{\frac{p_i^{kj}(1-p_i^{kj})}{n_i^k}})$$

If we want to control the type II error, a solution to avoid estimating the unknown parameter p^{kj}_i consists in changing the variable (Lehmann - 1986, Kendall - 1987). In fact, if n^k_i is high enough (in practice for $n^k_i > 20$) then,

$$Arc\sin(\sqrt{F_i^{kj}}) \to N(Arc\sin(\sqrt{p_i^{kj}}), \sqrt{(\frac{1}{4n_i^k})})$$

The statistic of the test is the variable $D = Arcsin(\sqrt{F^{kj}_i}) - Arcsin(\sqrt{F^{kj}_{i'}})$ normally distributed:

$$D \to N(Arc\sin(\sqrt{p_i^{kj}}) - Arc\sin(\sqrt{p_{i'}^{kj}}), \sqrt{(\frac{1}{4n_i^k} + \frac{1}{4n_{i'}^k})})$$

It can be deduced that under H0, as by hypothesis $p^{kj}_i - p^{kj}_{i'} = 0$, we have:

$$D \to N(0, \sqrt{(\frac{1}{4n_i^k} + \frac{1}{4n_{i'}^k})})$$

For a given alpha value α, the decision rule is:
If $D_{kj}(i,i') <$ Lim, H0 can not be refused; otherwise, H0 is refused where:

$D_{kj}(i,i')$ is the value of D observed over s_i and s_i' for the sub-sequence kj (with smoothing):

$$D_{kj}(i,i') = Arc\sin(\sqrt{f_i^{kj}}) - Arc\sin(\sqrt{f_{i'}^{kj}})$$

$$Lim = z_\alpha \sqrt{(\frac{1}{4n_i^k} + \frac{1}{4n_{i'}^k})}$$

z_α is the percentile of the standard normal distribution with zero mean and unity standard deviation, corresponding to (1-α).

For a given alpha value α, we can therefore calculate the type II error noted β. β is equal to the probability to decide H0 if H1 is true:

$$\beta = P_{H1}(D < \mathrm{Lim})$$

$$\Leftrightarrow \beta = P(N(0,1) < (\frac{\mathrm{Lim} - \delta'}{\sqrt{(\frac{1}{4n_i^k} + \frac{1}{4n_{i'}^k})}}))$$

$$\delta' = Arc\sin(\sqrt{p_i^{kj}}) - Arc\sin(\sqrt{p_{i'}^{kj}})$$

In this test, with a given alpha value α, there is a risk equal to β of not noticing a difference which is equal to δ' between $Arcsin(\sqrt{p^{kj}_i})$ and $Arcsin(\sqrt{p^{kj}_{i'}})$. For a given difference δ' between $Arcsin(\sqrt{p^{kj}_i})$ and $Arcsin(\sqrt{p^{kj}_{i'}})$, it is possible to compute the difference δ between p^{kj}_i and $p^{kj}_{i'}$. In fact, we demonstrated that:

$$\delta = Sin^2(\delta')(1 - 2p_{i'}^{kj}) + 2\sqrt{(p_{i'}^{kj}(1 - p_{i'}^{kj}))}Cos(\delta')Sin(\delta')$$

3.3. *Algorithms of comparison of PST and of clustering of sequences*

In the supervised classification of sequences, this test may be used as a complement to the Variable order chain. In the first step, a PST S_c is constructed for each class c (c = 1,..,C). In S_c , the empirical probabilities φ^{kj}_c are computed over all sequences of this class belonging to the training set:

$$\varphi_c^{kj} = \frac{n_c^{kj}}{n_c^{k*}}$$

where n^{kj}_c is the number of occurrences of j after k in all the sequences of the class c belonging to the training set and n^{k*}_c is the number of occurrences of any character of A after k in these sequences. In a model with smoothing, these conditional probabilities are replaced by f^{kj}_c (as defined by equation [4] in paragraph 2.2) . In a second step, given a new sequence s to classify, we compute a prediction probability for s with each PST S_c (c = 1,..,C) as mentioned in the paragraph 2.2. (Equation 2). The sequence can then be assigned to the PST type S_{c0} class for which the highest probability has been obtained (as indicated in paragraph 2.2, equation 3). Moreover, a PST S is constructed from the sequence s. Then, each node k belonging to one of the two PST (S or S_{c0}) is compared to the corresponding node k in the other PST using the preceding test for any character j of A.

Over the total number of the tests carried out, several indicators are computed to compare the Variable order chain represented by PST S and associated to s on the one hand and, the model associated to $S_{c0,}$ the PST corresponding to its assigned class c_0:

- the number of tests carried out
- the number (or percentage) of tests where H0 can not be refused.
- the average of the P-values calculated over the tests.
- the minimum of the P-values calculated over the tests.
- The average of the type II errors calculated over the tests.
- The maximum of type II errors calculated over the tests.

The lower the maximum of type II errors, the lower the risk of deciding that the subjacent model to s corresponds to the model associated to the class c_0 when in fact s does not belong to this class. By pruning, a node k can not appear in one of the two PST. In this case, the test may include smoothing and be carried out after associating to k a previously chosen minimum number of occurrences (nmin) as well as a minimal value of empirical conditional probability (ymin for example).

This test can also be used directly as decision rule for a sequence s represented by a PST S. The general principle of the algorithm consists in comparing each PST S_c ($c \in \{1,\ldots C\}$) to S in each node k with the test if the conditions of application are respected. Over the whole of the tests carried out between S and S_c, critical probabilities are computed. The sequence is then allocated to the class c_0 corresponding to the PST S_{c0} for which the average critical probability is maximal. The classification algorithm of sequences classSEQ is given in appendix 1.

From a computational point of view, it is obvious that the cost of the statistical test is higher than the cost of the decision rule used in Variable order chain since the former requires the construction of a PST for each sequence to classify.

But this is not so dissuasive in that sense that Variable order chain can be learned more efficiently than order L Markov chain. The estimation of parameters in the latter requires data length and time exponential in L whereas Ron's model can be learned efficiently in a PAC-like sense. It has been proved (Ron - 1996) that for every variable order chain M[n,n] and for every given security parameter $0 < d < 1$ and approximation parameter $0 < e < 1$ the learning algorithm outputs a PST T such that with probability at least $1 - d$, T is an e-good hypothesis with respect to M.

4. Application in the comparison of chronobiological sequences [2]

As illustration, we present in this section an application (Largeron – 2001) to chronobiological sequences describing the activity of fish which may be swimming (N), resting (R), submerged and immobile at the bottom of the aquarium (B) or lying in wait of food (F). The activity of each fish is noted each minute over a period of time of up to 24 hours, and so is described by a sequence of at most 1440 characters belonging to the alphabet A = {N, R, B, F}. The fish are kept in differing light conditions, some alternating between night and day, others under constant light, and, it is expected that depending on the conditions of light, they will adopt different attitudes. Among the 53 available sequences, 15 correspond to constant light conditions while 38 were observed in alternating conditions of light and dark. The aim is to define a model which will enable us to forecast the conditions of observation (daylight / darkness or artificial lighting) in which any sequence has been observed

[2] We wish to thank all the members of the Animal Biology team of the University of Saint-Etienne who provided the chronobiological data and their help in the processing.

4.1. Experimental Procedures

In this comparative study, we used several models: Markov chains of order L, variable order chains and statistical test, with or without probability smoothing. As smoothing and pruning techniques we applied those used in (Bejerano-2001). The aim was not only to compare these models but also to study the impact of the choice of smoothing and pruning parameters on the results overall. To achieve this, we carried out several experiments varying the values of these parameters. The results of these experiments are presented in Table 4 where each configuration corresponds to a set of different values of parameters.

Considering the small size of the data available, we used the leaving one out method to calculate the error rate for each configuration: the test sample is thus reduced to one element, the other elements belong to the learning sample. The experiment is renewed as often as there are available elements (53) and the error rate for a configuration corresponds to the number of elements, which are wrongly classified over the 53 experiments.

4.2 Comparison of models

Pruning implies considerable reduction in the size of PST as indicated in Table 3 showing the number of nodes of PST S_1 and S_2 associated with each of the two classes with different values of r and pmin; L, ymin and a remaining constant, respectively equal 15, 0.001 et 0.

r	pmin	Size S_1	Size S_2
1	0.001	968	844
1.05	0.001	903	671
1.2	0.001	634	462
2	0.001	218	159
3	0.001	111	53
1	**0.01**	100	54
1	**0.1**	37	31

Table 3. ***Computational costs***

The interest in reducing the computational cost is to allow the increase in the order L of the model, when this is justified. Generally speaking, this is the case in this application in which a slight decrease in the rate of misclassification can be observed (MC = 9%) in relation to the order L, as shown in Table 4. (configuration 6 with L = 15 in relation to the configuration 5 with L = 5 and MC = 11.3)

In Table 4, the first configuration corresponds to a Markov chain of order L (L=5) without probability smoothing (ymin = 0) and without pruning (pmin = 0, r= 1 and a =0); the error rate is quite high (MC = 69.81%). This results from the impossibility of assigning a large number of sequences to one class.

An improvement in the result is obtained (MC = 51 %) by proceeding with tree pruning (pmin = 0.001 et r= 1.05) or to put it another way, by using the variable memory length Markov model (configuration 2).

By introducing probability smoothing (ymin = 0.001) in the Markov chain of order L (L = 5), there is significant improvement (MC = 9.00 % - configuration 3) in relation to certain values of parameters. However, beyond a certain threshold this tendency is reduced and the results become poorer (MC = 57,00% - configuration 4 when ymin = 0.1) in particular for class 1 in which the underlying hypothesis of the model appears to be verified (MC1 = 76% and MC2 = 7%).

Finally, by combining smoothing probability and pruning (ymin = 0.001, pmin = 0.001, r= 1.05 , a= 0) a quite satisfactory result is obtained (MC = 11.3 % - configuration 5) at an inferior storage cost.

Configurations	L	pmin	r	ymin	a	MC (in %)
1	5	0	1	0	0	69.81
2	5	**0.001**	**1.05**	0	0	51.00
3	5	0	1	**0.001**	0	9.00
4	5	0	1	**0.1**	0	57.00
5	5	**0.001**	**1.05**	**0.001**	0	11.3
6	**15**	**0.001**	**1.05**	**0.001**	0	9.00
7	**15**	**0.001**	**1.05**	**0**	0	87 %
I	**15**	0.001	1.05	0.001	0	11.00
II	**5**	0.001	1.05	**0**	0	19.00
III	**15**	0.001	1.05	**0**	0	13.00

Table 4. Comparison of models: error rate.

Configurations I and II in Table 4 give the error rate (ie percentage of misclassification) obtained by using the statistical test as decision rule (with nmin = 20 to respect the conditions in which the test must be applied). The results (MC = 11 % for L = 15 configuration I) are equivalent to those obtained by the variable memory length Markov model with probability smoothing (configuration 6 in Table 4).

Finally by using the test as decision rule without probability smoothing during the PST construction and the prediction step, we observe relative stability in the results (MC = 19 % configuration II and MC = 13% configuration III Table 4). On the other hand, results become poorer without probability smoothing with Variable memory length Markov models (MC = 51% configuration 2 and MC = 87% configuration 7- Table 4). From these experiments, the statistical test used as decision rule, enables us to do without probability smoothing and so, to avoid choosing a smoothing technique and smoothing parameters.

These results are highly encouraging. They confirm the significance of this model which has already been demonstrated using larger data set. It is important to point out that in biology, the comparison and the classification of sequences have been the subject of highly interesting papers (Day - 94, Kruskal - 83). The advantage of this approach is that it does not require the prior alignment of sequences; which in itself is a difficult operation made all the more delicate in that it conditions the quality of the results.

5. Conclusion

Trees provide a natural representation for sequential data since they respect the order of the characters in the sequence. In this article, we chose a PST type tree, equivalent to a variable memory length Markov model. The advantage of this model compared to the Markov chain of order L is in the pruning of the tree branches. This pruning enables us to reduce computational costs and thereby increase the order of the model. However, in supervised classification, nothing can guarantee that there will not be different sub-sequences in the subjacent model of the sequence to classify and in the model of its class. This is the reason why we set out an improvement of the

variable memory length Markov model based on a statistical test enabling us to calculate this risk. We compared the results of different models on chronobiological sequences describing the activity of fish. The results obtained are highly satisfying and confirm the interest of these models, even when the training set is relatively small. They also demonstrate the interest of probability smoothing both for Markov chains of order L as well as Variable order chains on condition that the smoothing parameters have been correctly chosen. Employed as a rule of classification the statistical test enables us to avoid choosing smoothing technique and smoothing parameters.

6. Bibliography

AGRESTI A. (1990; 2ND EDITION 2002), Categorical data analysis sing substitution probabilities to improve position-specific scoring matrices*, John Wiley and sons.*

BEJERANO G. , YONA G. (2001) Variations on probabilistic suffix trees: statistical modelling and prediction of protein families. *Bioinformatics,* 17(1), p. 23-41.

BUHLMANN P. , WYNER A. (1999) Variable order chains *The annals of Statistics*, *Vol 27, N° 2, p. 480-513.*

DAY W.H., MCMORRIS F. R.(1994), Alignment, comparison and consensus of molecular sequences*, New approaches in classification and data analysis* Diday E. Editors, P. 327 -346.

FISHER R.A. (1925 - 1970), Statistical Methods for Research Workers*, Edinburgh : Oliver and Boyd*.

GREENWOOD P.E., NIKULIN M.S. (1996), A guide to Chi-Squared testing*, John Wiley and sons*.

HENIKOFF J.G., HENIKOFF S. (1996), Using substitution probabilities to improve position-specific scoring matrices*, Com. App. Biosci*. 12:2, p. 135-143.

KATZ S.M. (1987), Estimation of probabilities from sparse data for the language model component of a speech recognizer*, IEEE Trans. on Acoustics, Speech and Signal Processing*, ASSP-35, n°3 , p. 400-401.

KENDALL M., STUART A(1987), Kendall's advanced theory of statistics, vol 2, Griffin.

KERMORVANT C, DUPONT P. (2002) Improved smoothing for variable order Markov chains*, European Conference on Machine Learning (ECML),* Helsinki, Finland, 2002.

KERMORVANT C, DUPONT P. (2002b) C. Chaînes de Markov d'ordre variable pour la détection de domaines dans les protéines, *(JOBIM),* Saint-Malo France

KRUSKAL J.B.(1983), *An overview of sequence comparison*, Time Warps, String Edits, and Macromolecules: the theory and practice of sequence comparison, Addison-Wesley Publishing Company. p1-44.

LARGERON C (2001)*,* Algorithme de comparaison d'arbres: application au classement de séquences. *SFC'01 7eme conférence de la Société Francophone de Classification Pointe-à-Pitre (available from Internet, visited 19/05/03)*

http://www.univ-st-etienne.fr/creuset/puvwp/Largeron_sfc2001rev.ps

http://www.univ-st-etienne.fr/creuset/pubwp/Largeron_sfc2001rev.pdf)

LEHMANN L.(1986), *Testing statistical hypotheses,* John Wiley Sons

MEHTA C.R., PATEL N.R. (1983), A network algorithm for performing Fisher's exact test in r × c contingency tables*, Journ. of American Statistical Associ*. 78:382, p. 427-434.

REINGOLG E.M., NIEVERGELT J., DEO N.(1977) *Combinatorial algorithm theory and practice*, Englewood Cliffs.

RISSANEN J.(1983), A universal data compression system *, IEEE Trans Infor Theory,* vol 29, n° 5, p. 656-664

RON D, SINGER Y., TISHBY N.(1996) The power of amnesia: learning probabilistic automata with variable memory length*, Machine learning* N°25, p. 117-149

WEINBERGER M. J . LEMPEL A., ZIV J. (1992) A sequential algorithm for the universal coding of finite memory sources*, IEEE Trans Infor. Theory*, vol 38, p. 1002-1014.

WILLEMS F. M. J. , SHTARKOV Y.M., TJALKENS T. J.(1995) The context tree weighting method: Basic properties*, IEEE Trans Infor. Theory ,* vol 41,. 53-664.

7. Appendix

The algorithm comparPST($\alpha, \varepsilon', n\min, y\min$)

// input

{S_1,...,S_c ,...S_C}	set of PST type
S_i	PST to classify

// output

c_0	cluster of S_i

Begin
For all c $\in \{1, .., C\}$ do
Begin // Comparison of PST S_c and S_i
// Initialization
NT[c]:= 0; // Number of tests carried out between S_c and S_i
PC[c]:=0; // Sum of the P-values calculated over the tests

For all k $\in S_c \smile S_i$ do

Begin // Analysis of node k belonging to S_c or S_i
If k $\in S_c$ and k $\notin S_i$ then
Begin
For all j $\in$ A do
Begin
if ($\frac{f_i^{kj}}{f_i^{suf(k)j}} \leq r$) AND ($\frac{f_i^{kj}}{f_i^{suf(k)j}} \geq \frac{1}{r}$) then $f_i^{kj} = f_i^{suf(k)j}$
else $f_i^{kj} = y\min$;
EnFor_j
Endif
If k $\notin S_c$ and k $\in S_i$ then
Begin
For all j $\in$ A do
Begin
if ($\frac{f_c^{kj}}{f_c^{suf(k)j}} \leq r$) AND ($\frac{f_c^{kj}}{f_c^{suf(k)j}} \geq \frac{1}{r}$) then $f_c^{kj} = f_c^{suf(k)j}$
else $f_c^{kj} = y\min$;
EndFor_j
Endif;

For all j $\in$ A do
Begin // Comparison of p_c^{kj} and p_i^{kj}
if ($n^k i \prec$ nmin) then $n^k i$:=nmin;
if ($n^k c \prec$ nmin) then $n^k c$:=nmin;
NT[c]:=NT[c]+1;
D_{kj}(i,c):= Arcsin($\sqrt{f_i^{kj}}$) − $Arc\sin(\sqrt{f_c^{kj}})$;
L:= $t_\alpha \times \sqrt{\frac{1}{4n^k i} + \frac{1}{4n^k_c}}$;
PC := 1−P(N(0,1)< $\frac{D_{kj}(i,c)}{\sqrt{\frac{1}{4n^k i} + \frac{1}{4n^k c}}}$);

PC[c]:= PC[c]+PC ;
β :=P(N(0,1) $\prec$ $\frac{L-\varepsilon'}{\sqrt{\frac{1}{4n^{k_i}}+\frac{1}{4n^{k_c}}}}$);
EndFor_j
EndFor_k
EndFor_c

// End of comparison of PST S_c and S_i

// Decision rule: maximum of the average critical probabilities
Maxpc:= 0;
For all c $\in \{1, ..C\}$do
Begin
If $\frac{PC[c]}{NT[c]}$ > Maxpc then
Begin
Maxpc := $\frac{PC[c]}{NT[c]}$;
c_0 := c;
EndIf
EndFor_c