Markov models for identification of significant episodes

Robert Gwadera *
Purdue University
Department of Computer Sciences
gwadera@cs.purdue.edu

Mikhail Atallah †
Purdue University
Department of Computer Sciences
mja@cs.purdue.edu

Wojciech Szpankowski ‡
Purdue University
Department of Computer Sciences
spa@cs.purdue.edu

October 4, 2004

Abstract

We propose a new method for a reliable identification of statistically significant sequential episodes occurring *within a window of size* w in an event sequence modeled by a Markov source. As a measure of significance we use $\Omega^{\exists}(n,w)$, the number of windows containing the episode as a subsequence. We prove that $\Omega^{\exists}(n,w)$ is a sum of a φ-mixing sequence of random variables and obeys the central limit theorem. This leads us to a computational formula for a threshold to identify significant episodes. The novelty of our method for Markov source stems from the fact that, instead of scoring the whole sequence using a Markov model, we compute the expected value of $\Omega^{\exists}(n,w)$ and its variance in order to compute the threshold and compare it to the observed $\Omega^{\exists}(n,w)$. Since performance of the method critically depends on the model structure and parameters, we argue that variable order Markov models of event streams are superior to fixed order Markov models. We chose DNA sequences as event sources in experiments, and compared the performance of fixed order Markov models with interpolated Markov models. This paper is an extension of our previous work in [9, 2] where we considered the problem of the reliable detection of significant episodes for memoryless sources.

Keywords: frequent episode mining, probabilistic models

1 Introduction

Mining episodes was introduced in [11], where Mannila et al. defined the problem of finding frequent episodes in event sequences, subject to the observation-window constraint. An episode was defined as a partially ordered collection of events, that can be represented as a directed acyclic graph. We distinguish three types of episodes. A *serial episode* is a sequence of events that occur in a window of size w in a specified order. As a graph the serial episode is a single directed path from the first episode to the last one. A *parallel episode* is an unordered collection of events that occur in a window of size w. A *composite episode* can be build recursively from an event, a serial episode and a parallel episode. Given a window length w and an event sequence T, an episode α was defined as *frequent* if its frequency, defined as the fraction of windows in which it occurred at least once, was more that a given frequency threshold τ. Note that the same episode may be present in several consecutive windows but within a particular window we count one occurrence even if there are may instances of it in that particular window.

In our previous papers [9, 2] we assumed that the event sequence T was generated by the Bernoulli (memoryless) source and showed how to compute the threshold τ as well as how to design the window size w such that the discovered frequent episodes are statistically significant. Observe that, for an appropriately large

*The work of this author was supported by the NSF Grant CCR-0208709, and NIH grant R01 GM068959-01.

†Portions of this author's work were supported by Grants EIA-9903545, IIS-0219560, IIS-0312357, and IIS-0242421 from the National Science Foundation, Contract N00014-02-1-0364 from the Office of Naval Research, by sponsors of the Center for Education and Research in Information Assurance and Security, and by Purdue Discovery Park's e-enterprise Center.

‡The work of this author was supported by the NSF Grant CCR-0208709, NIH grant R01 GM068959-01 and AFOSR Grant FA 8655-04-1-3074.

window size w and/or low frequency τ any episode will almost surely occur in every window because the probability of its existence in the window of size w $P^{\exists}(w)$, estimated as a fraction of windows in which the episode occurred, will be close to one. In particular, paper [9] considered serial episodes and [2] considered sets of episodes including the special case of the parallel episode. We used $\Omega^{\exists}(n, w)$, the number of windows of length w, that contain an episode α as a subsequence in event sequence T after n shifts of the sliding window. Using the fact that $\Omega^{\exists}(n, w)$ is a sum of w dependent random variables we proved that appropriately normalized $\Omega^{\exists}(n, w)$ has a Gaussian distribution, where clearly the expected value $\mathbf{E}[\Omega^{\exists}(n, w)] = nP^{\exists}(w)$ and $P^{\exists}(w)$ is the probability that the episode occurs at least once in a window of length w in an event sequence T over an alphabet $\mathcal{A}$. We also showed that the variance $\mathbf{Var}[\Omega^{\exists}(n, w)] \leq cn\left[P^{\exists}(w) - (P^{\exists}(w))^2\right]$ for $c > 0$. Given a Bernoulli model of the source, and for a given probability $\beta(b)$, we presented the upper threshold for detecting significant episodes $\tau_u(w) = P^{\exists}(w) + \frac{b\sqrt{\mathbf{Var}[\Omega^{\exists}(n,w)]}}{n}$, where $P^{\exists}(w)$ and $\mathbf{Var}[\Omega^{\exists}(n, w)]$ depend on the probabilistic model and episode type. For a given level $\beta(b)$ of the cumulative normal probability distribution function we select b such that $P\left(\frac{\Omega^{\exists}(n,w)}{n} > \tau_u(w)\right) \leq \beta(b)$. That is, if one *observes* more than $\tau_u(w) \cdot n$ occurrences of windows with certain episodes, it is highly unlikely that such a number is generated by the probabilistic model of the source(i.e., its probability is smaller than $\beta(b)$). The quantity $\frac{\Omega^{\exists}(n,w)}{n}$ is an estimator of $P^{\exists}(w)$ denoted $P_e^{\exists}(w)$. While developing the formula for $P^{\exists}(w)$ we found a formula for the set of all distinct windows $\mathcal{W}^{\exists}(w)$ of length w containing the serial episode S of length m at least once as a subsequence. The importance of $\mathcal{W}^{\exists}(w)$ stems from the fact that $P^{\exists}(w) = \sum_{x \in \mathcal{W}^{\exists}(w)} P(x)$ for the Markov model of an arbitrary order including 0-th order (Bernoulli), where $P(x)$ is the probability of x as a string of symbols of length w in a given model. The advantage of the Bernoulli model versus the first order Markov or higher is that for the Bernoulli model $P^{\exists}(w)$ can be computed efficiently exploiting the structure of $\mathcal{W}^{\exists}(w)$ and the fact that the model requires only $|\mathcal{A}|$ probabilities of symbols of the alphabet $\mathcal{A}$. Using generating functions and complex asymptotics we presented an asymptotic approximation of $P^{\exists}(w)$, which is of the form $P^{\exists}(w) = 1 - \Theta(\rho^w)$ for large w and $0 < \rho < 1$. We provided fast dynamic programming algorithms for computing $P^{\exists}(w)$ for a serial episode and for an arbitrary set of episodes. In [9] we mined two apparently non-memoryless sources (the English alphabet and the web access data) and showed that, even for these cases, $P^{\exists}(w)$ closely approximated the actual $P_e^{\exists}(w)$, which demonstrated that the memoryless assumption did not limit the practical usefulness of the formula. In [2] we mined a large database of *Wal-Mart* transactions for sets of episodes including composite episodes. We also showed that the threshold mechanism indeed provides a sharp detection of significant episodes by continually injecting some episodes until they exceeded the threshold.

This paper extends our previous work to the case of Markov sources of any order that are more applicable and flexible than memoryless sources. The present extension to Markov was in part motivated by concerns expressed by some of the anonymous reviewers of our previous papers that our work may not generalize to the more difficult Markov models, and the audience at the conferences expressing the view that Markov model would be more important from a practical point of view than Bernoulli.

Our threshold mechanism for Markov models is computed in the same way as for Bernoulli model. The only difference is in computing the parameters of the model i.e. we need to appropriately estimate conditional probabilities based on training data and to efficiently compute $P^{\exists}(w)$ and $\mathbf{Var}[\Omega^{\exists}(n, w)]$ using those conditional probabilities. We cannot use the dynamic programming algorithm for computing $P^{\exists}(w)$ in Bernoulli model because the idea of that algorithm was based on the fact that the probability of a symbol in Bernoulli model does not depend on its context, which is no longer true for a Markov model of order ≥ 1. Therefore in this paper we focus on finding the best methods of estimating and computing the parameters. There are three main new challenges we faced and resolved in this extension. The first is theoretical; we must establish the central limit theorem for $\Omega^{\exists}(n, w)$ for the Markov event source. The second challenge is algorithmic; we must provide algorithms for computing $P^{\exists}(w)$. Finally we must select an optimal model structure and parameters to ensure that the prediction of the model is accurate. We address the three questions in Section 2, where we prove that $\Omega^{\exists}(n, w)$ is a sum of a φ-mixing sequence of random variables meaning that the distant future is practically independent of the present and past. Therefore we can apply the central limit theorem. We also present an algorithm for computing the threshold. In Section 3 we compared the prediction performance of the fixed order Markov model with the interpolated Markov model using DNA sequences. While the first two problems can be solved exactly the issue of choosing the model structure and parameters in terms of decreasing the rate of false positive predictions remains a challenge despite recent improvements in statistical methods.

Given an alphabet of size $|\mathcal{A}|$ we could model the event stream T by a fixed order Markov model of order k with $\mathcal{A}^{k+1}$ conditional probabilities to be estimated from the training data. Ideally, longer context length k is always preferable than shorter contexts. Thus, higher-order model should always do at least as well as, and frequently better than lower-order models. In practice, when using the model of order k, if the training sequence is of length N, then there are only $N-k$ strings of size $k+1$ available to estimate $|\mathcal{A}|^{k+1}$ conditional probabilities and frequencies of some of the $|\mathcal{A}|^k$ context strings become too small or even zero. Therefore, an attempt to derive a model of too high order will lead to over-fitting. More formally, let c_k be a context of length k in a model of order k and $a_i \in \mathcal{A}$. We estimate the conditional probabilities $P(a_i|c_k)$ using the maximum likelihood (ML) estimate given by the following formula $P(a_i|c_k) = \frac{n(c_k a_i)}{\sum_{i=1}^{|\mathcal{A}|} n(c_k a_i)}$, where $n(c_k a_i)$ is the frequency of the string $c_k a_i$ observed in the training set. Notice that even if the training set is too short to accurately estimate all probabilities, for some probabilities $P(a_i|c_k)$ the number of occurrences of the string $c_k a_i$ will be sufficient and should be accepted by the model. The problem of parameter estimation of Markov models from sparse training data is known as *smoothing* and has been widely discussed in the literature on statistical modeling [7]. The smoothing is a technique for adjusting the maximum likelihood estimates of probabilities to produce more accurate probabilities. The name smoothing comes from the fact that these methods tend to make the probabilities more uniform, by adjusting low probabilities as zero upward and higher probabilities downward. Not only do smoothing methods generally prevent zero probabilities, but they also improve the accuracy of the model as the whole. Whenever a probability is estimated from a fewer counts, smoothing has the potential to significantly improve estimation. In particular, techniques as *interpolation* and *back-off* have been implemented to deal with sparse data. The idea is to use a Markov chain with a new structure, where a conditional probability of order k is a combination of equal and lower order probabilities weighted by interpolation parameters, giving hight weight to probability estimates corresponding to high frequency contexts and lower weight to estimates corresponding to low frequency contexts. Besides the interpolated Markov model (IMM) and the back-off model there are other smoothing techniques available including *contexts tree weighting* [16] or *interpolated context models* (ICM) [8] which is a further extension of IMM. While in IMM we estimate the probability $P(a_i|c_k)$ of a symbol a_i based on variable length contexts immediately preceding a_i the ICM is more flexible and general by allowing to choose any contexts (not just those adjacent to a_i). However we focus in this paper on the interpolated Markov model and compare its performance with a fixed order Markov model for detecting significant episodes. Thus, we are interested in Markov models that define conditional probabilities $P(a_i|c_k)$ as a linear combination of conditional probabilities associated with contexts c_i of smaller lengths contained in c_k for $i = k-1, k-2, \ldots 0$. The following recursion defines a value of the interpolated conditional probability in IMM:

$$\begin{aligned} P_{IMM}(a_i|c_k) &= \lambda(c_k) \cdot P(a_i|c_k) + \\ & \quad [1-\lambda(c_k)] \cdot P_{IMM}(a_i|c_{k-1}), \end{aligned}$$

where $0 \leq \lambda(c_k) \leq 1$ and $P(a_i|c_k)$ is the probability estimate using the maximum likelihood (ML) estimate from the training data. For contexts c_k not observed in the training data, i.e., if $n(c_k, a_i) = 0$, we set $P(a_i|c_k) = P(a_i|c_{k-n})$ for $n = min_{1\leq n\leq k}\{n|n(c_k, a_i) > 0\}$. The value of the parameter $\lambda(c_k)$ can be interpreted in many ways. We can view the parameters (i) as combining the predictions from Markov models of varying length, or (ii) as hidden transitions from a higher order Markov model to a lower Markov model or finally (iii) as beliefs about conditional independence. According to the interpretation (i), longer histories support stronger predictions, while shorter histories have more accurate statistics. Interpolating the predictions from histories of different lengths results in more accurate predictions that can be obtain from any fixed history length. According to interpretation (ii), the interpolation parameters model our beliefs about how much of the past is necessary to predict a state transition in an underlying Markov source of unknown order. Alternatively, (iii) interprets the parameters as our beliefs about conditional independence with probability $[1-\lambda(c_k)]$ that the future does not depend on the first symbol of c_k. In general if the frequency of a context c_k is sufficiently high, the value of $\lambda(c_k)$ is close to 1. In the opposite case $\lambda(c_k)$ is close to zero and the interpolation probability $P_{IMM}(a_i|c_k)$ gains more from $P_{IMM}(a_i|c_{k-1})$. However the problem of finding interpolation parameters is still more of an art than an exact science. In our experiments we used a modification of the method based on χ^2-test introduced in [13].

There is yet another reason why we would prefer non-uniform Markov models for discovering significant episodes versus fixed order models. Sometimes we have an expert knowledge about the source of events and we know for example that some symbols occur only in *q-grams* (strings of n ordered elements). Using a fixed order Markov model seems to be too excessive while the Bernoulli model too inaccurate to capture the dependency of symbols in the *q-grams*. Therefore the variable model does exactly meet our requirement, and we can set $\lambda(c_k) = 0$ to all contexts but the few for the *q-grams*.

We conduct our experiments on genomic data represented as strings of nucleotide symbols over the alphabet $\mathcal{A} = \{A, C, G, T\}$. Markov models of DNA sequences have frequently been used in gene finding algorithms [13], where the interest was in finding strings of symbols instead of subsequences in the form of episodes. The novelty of our approach is that we treat the genomic sequence as a stream of symbols generated by a Markov model of an unknown order and we do not consider any biological structures as coding/non-coding regions in the DNA. Furthermore we do not score the testing sequence using a trained Markov model as in the work on gene discovery in order to determine whether the sequence has been generated by the model. Adapting the method of scoring the sequence for episode discovery would mean training a separate Markov model for every combination of window length and episodes type. Note that in the episode framework we consider $\Omega^{\exists}(n, w)$ the number of windows containing an episode as a subsequence, which is a function of the Markov chain rather than a well defined structure (coding/non-coding regions) of the sequence as in the gene discovery methods. Therefore in our method for the reliable detection of significant episodes we use the Markov model only to compute the expected value and variance of $\Omega^{\exists}(n, w)$ needed for the threshold computation. Because of the sequential nature of Markov sources we consider only serial episodes while using Markov models. We do not test the threshold $\tau_u(w)$ directly in experiments by computing its value and simulating occurrences of significant episode as in [9, 2] because we already showed in [9, 2] that the accuracy of the threshold is determined by prediction accuracy of the formula for $P^{\exists}(w)$. Therefore we test the threshold indirectly by focusing on the predictive performance of the formula for $P^{\exists}(w)$ for Markov models.

Perhaps the most intriguing question is whether we can improve our detection method on DNA data in terms of accuracy by employing the Markov model rather than the Bernoulli model. As we will see in experiments the answer to this question is affirmative.

The paper is organized as follows. In section 2 we present our main results containing theoretical foundation. Section 3 contains experimental results demonstrating the applicability of the derived formulas.

Interested readers can visit our on-line threshold calculator at `http://www-cgi.cs.purdue.edu/cgi-bin/gwadera/demo.cgi` for a demonstration of the reliable threshold computation.

2 Analytical results

2.1 Definition of the problem of identification of significant episodes

For clarity of the presentation we analyze only the case of a single serial episode $S = S[1]S[2]\dots S[m]$ of length m but the results can be generalized to an arbitrary set of serial episodes. Of course we do not analyze parallel episodes since they are unordered sequences.

The problem of identification of significant episodes in the Markov source T can be stated as follows. Given:

- a Markov model of order k with conditional probabilities $p(a_i|c_k)$, where $a_i \in \mathcal{A}$ and $c_k \in \mathcal{A}^k$.
- $\Omega^{\exists}(n, w)$, the observed number of windows of length w containing at least one occurrence of a serial episode $S = S[1]S[2]\dots S[m]$ after n shifts of the window
- a level $\beta(b)$ (e.g., $\beta(b) = 10^{-5}$).

Is the observed frequent episode S statistically significant? In Section 2.2 we prove that $\Omega^{\exists}(n, w)$ is normally distributed (Theorem 1). This will allow us to compute the threshold as follows

$$(1)\qquad \begin{cases} \tau_u(w) &= P^{\exists}(w) + \frac{b\sqrt{\mathbf{Var}[\Omega^{\exists}(n,w)]}}{n} \\ \beta(b) &= \frac{1}{\sqrt{2\pi}}\int_b^{\infty} e^{\frac{-t^2}{2}}\,dt \end{cases}$$

where $\mathbf{Var}[\Omega^{\exists}(n, w)] \le [n + (2n - w)(w - 1)][P^{\exists}(w) - (P^{\exists}(w))^2]$ and $P^{\exists}(w)$ depends on the episode and order of Markov model. Once the threshold $\tau_u(w)$ is computed if $\frac{\Omega^{\exists}(n,w)}{n} > \tau_u(w)$ then the probability that the episode S is not significant is less than $\beta(b)$.

In section 2.3 we provide an algorithm for computing $P^{\exists}(w)$.

2.2 Central limit for $\Omega^{\exists}(n, w)$

We consider a stationary and ergodic infinite Markov source $T = t_1 t_2 \dots$ generating symbols from an alphabet $\mathcal{A} = \{a_1, a_2, \dots, a_{|\mathcal{A}|}\}$ where the distribution of every symbol at any position in the source depends on the first k symbols preceding called the context c_k.

Definition 1 *A stochastic source over an alphabet $\mathcal{A}$, is said to have the* **Markov property** *if there exists a natural positive number k such that:*

$$(2)\qquad P(t_1, \dots t_n) = P(t_1, \dots t_k) \cdot \prod_{i=k+1}^{n} P(t_i | t_{i-1}, \dots, t_{i-k})$$

for all $n > k$ and $(t_1, \ldots t_n) \in \mathcal{A}$. The minimal number k such that the equation (2) holds, is called the **order** *of the source.*

A source with the Markov property and order k is termed a k-order Markov source. The stationary property means that the conditional probability $P(t_i|t_{i-1}, \ldots, t_{i-k})$ does not depend on i.

Now we derive the normal limiting distribution of $\Omega^{\exists}(n, w)$. Observe that

$$\Omega^{\exists}(n, w) = \sum_{i=1}^{n} I_i^{\exists}$$

where

$$I_i^{\exists} = \begin{cases} 1 & \text{the episode } S \text{ occurs at least once as a subsequence in the window ending at position } i \text{ in } T; \\ 0 & \text{otherwise,} \end{cases}$$

where i is the relative position with respect to the first position ($i = 1$). Thus, we easily have $\mathbf{E}[I_i^{\exists}] = P^{\exists}(w)$, $\mathbf{Var}[I_i^{\exists}] = P^{\exists}(w) - (P^{\exists}(w))^2$ and $\mathbf{E}[\Omega^{\exists}(n, w)] = nP^{\exists}(w)$.

Definition 2 *Let $\varphi_1, \varphi_2, \ldots$ be a sequence of numbers such that $\varphi_n \to 0$. A stationary sequence of random variables $X_1, X_2, \ldots X_n$ is φ-mixing if $|P(E_2|E_1) - P(E_2)| \le \varphi_n$. for every j and $E_1 \in (X_1, \ldots, X_j)$ and every k and $E_2 \in (X_{j+n}, \ldots, X_{j+n+k})$.*

In this definition, E_1 is an event that depends only on $X_1, \ldots, X_j$, and E_2 is an event that depends only on $X_{j+n}, \ldots, X_{j+n+k}$. The condition requires that E_1 and E_2 are almost independent in the sense that $|P(E_2|E_1) - P(E_2)|$ is small for large n.

According to [4], let $Y_1, Y_2, \ldots Y_n$ be a Markov chain with finite state space and positive transition probabilities p_{ij}. Let $X_i = f(Y_i)$, where f is some real function of the state space. If the initial probabilities are stationary then $Y_1, Y_2, \ldots Y_n$ is stationary. Moreover $|p_{ij}^n - p_j| \le \rho^n$ where $\rho < 1$. Let r be the number of states then $X_1, X_2, \ldots X_n$ is φ_n-mixing with $\varphi_n = r\rho^n$. For a proof see [4] Example 27.6.

Based on Theorem 27.5 in [4] and the fact that $I_1^{\exists}, I_2^{\exists}, \ldots I_n^{\exists}$ is φ-mixing the following theorem holds

Theorem 1 *The random variable $\Omega^{\exists}(n, w)$ obeys the Central Limit Theorem in the sense that its distribution is asymptotically normal, for $a, b = O(1)$ we have*

$$\lim_{n\to\infty} P\left\{ a < \frac{\Omega^{\exists}(n, w) - \mathbf{E}[\Omega^{\exists}(n, w)]}{\sqrt{\mathbf{Var}[\Omega^{\exists}(n, w)]}} < b \right\} = \frac{1}{\sqrt{2\pi}} \int_a^b e^{\frac{-t^2}{2}} dt$$

for w fixed.

Theorem 1 leads to the Formula (1) for the threshold.

2.3 Algorithm for $P^{\exists}(w)$

In this section we present an algorithm for computing $P^{\exists}(w)$ for a Markov model of any order.

The probability of existence of an episode α can be expressed as follows

$$P^{\exists}(w) = \sum_{x \in \mathcal{W}^{\exists}(w)} P(x). \tag{3}$$

where $\mathcal{W}^{\exists}(w)$ is the set of all distinct windows of length w containing the episode as a subsequence. Let x_i be the i-th symbol of a window $x \in \mathcal{W}^{\exists}(w)$ then the probability of the window can be computed as follows

$$P(x) = P(x_0)P(x_1|x_0) \ldots P(x_{k-1}|x_{k-2} \ldots x_0) \cdot \prod_{i=k}^{w-1} P(x_i|x_{i-1} \ldots x_{i-k})$$

where k is the order of the model. Thus, in order to compute the probability for order k we need models for all lower orders $k-1, k-2, \ldots, 0$. In our papers [9, 2] we gave formulas for $\mathcal{W}^{\exists}(w)$ for a single serial episode $S = S[1]S[2] \ldots S[m]$ and for an arbitrary set of serial episodes $\mathbf{S} = \{S_1, S_2, \ldots S_m\}$ respectively. In particular, $\mathcal{W}^{\exists}(w)$ for a single serial episode can be enumerated as follows

$$\mathcal{W}^{\exists}(w) = \bigcup_{\sum_{k=1}^{m+1} n_k = w-m} \overline{S[1]}^{n_1} \times S[1] \times \ldots \times \overline{S[m]}^{n_m} \times S[m] \times \mathcal{A}^{n_{m+1}},$$

where $\overline{a}$ denotes $\mathcal{A} - a$ for $a \in \mathcal{A}$. Using formula (4) directly is computationally very costly because $|\mathcal{W}^{\exists}(w)|$ is $O(\mathcal{A}^{w-m})$ [9]. Furthermore computing $P(x)$ for every window x independently would be inefficient because many windows share the same prefix. Therefore we propose a computational method where we enumerate the windows according to a depth-first traversal of a trie build from the members of $\mathcal{W}^{\exists}(w)$ without the trailing $\mathcal{A}^{n_{m+1}}$ that contributes a factor of 1 to the computation of the probability. The idea of this method is that probability of each prefix is computed once even if many windows share the same prefix. Of course we do not build the whole trie but we create one path at a time, i.e. the space is constant of order $O(w)$. An example of such a trie for $S = abc$ and $w = 4$ is shown in Figure 1.

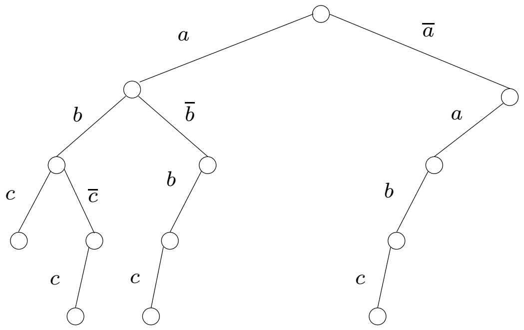

Figure 1: A trie for $\mathcal{W}^{\exists}(3)$ the set of windows of length $w = 4$ containing $S = abc$ as a subsequence

3 Experimental results

The ultimate measure of a statistical model is its predictive performance in the domain of interest. Therefore in experiments we compare the actual (observed) $P_e^{\exists}(w) = \frac{\Omega^{\exists}(n,w)}{n}$ value with $P^{\exists}(w)$ obtained using the trained model. As we stated in the introduction, we consider only a serial episode S of length m in experiments. We used an algorithm for finding occurrences of a serial episode to find $\Omega^{\exists}(n, w)$. To evaluate the performance of the model we used the following distance measure between two functions

$$(4) \qquad d \quad = \quad \left[\frac{1}{r} \sum_{i=1}^{r} \frac{|P_e^{\exists}(w_i) - P^{\exists}(w_i)|}{P_e^{\exists}(w_i)} \right] 100\%$$

where $w_1 < w_2 < \ldots w_r$ are the tested window sizes. We trained and tested our predictions on the following genomic sequences: *Haemophilius influenzea* of length $1,830,025$, *Helicobacter pylori* of length $1,667,826$ and *Human chromosome* 22 two segments of length $234,227$ and $3,661,561$ respectively. We estimated the conditional probabilities using the maximum likelihood estimator for both the fixed order and interpolated model IMM. We used *Helicobacter pylori* and the first segment of *Human chromosome 22* as training sets. We built fixed order Markov model of order: $0, 1, 2, 3, 4, 5$ and IMM of order $0, 1, 2, 3, 4, 5$ for each training set. All our algorithms have been implemented in C++ and run under Linux operating system. The first experiment on *Haemophilius influenzea* uses the fixed order Markov model. The IMM algorithm is presented in Section 3.2 and experiments using the IMM are presented in sections 3.3 and 3.4.

3.1 Haemophilius influenzea

In this experiment we wanted to experimentally confirm the correctness of our theoretical results including the proof of the central limit theorem, the derived formula for $P^{\exists}(w)$ and the algorithm for computing $P^{\exists}(w)$. We expected to achieve a better prediction accuracy using the Markov order 5 comparing to the order 0. To exclude a possibility of model misbehavior (overfitting, etc.) we used the fixed order model and the same sequence of *Haemophilius influenzea* as a training and testing source. We set a serial episode $S = CCGT$ and for each model order for $k = 0, 1, 2, 3, 4, 5$ we computed $P^{\exists}(w)$ and compared to an observed $P_e^{\exists}(w)$ for $w = [5, 20]$ by computing the prediction error using Equation 4. The computed prediction errors for orders $k = 0, 1, 2, 3, 4, 5$ are represented by a bar graph in Figure 2. Clearly the prediction error decreases monotonically starting from order 1 up. Order 5 gives the best prediction significantly outperforming the order 0. This validates our theoretical and algorithmic results.

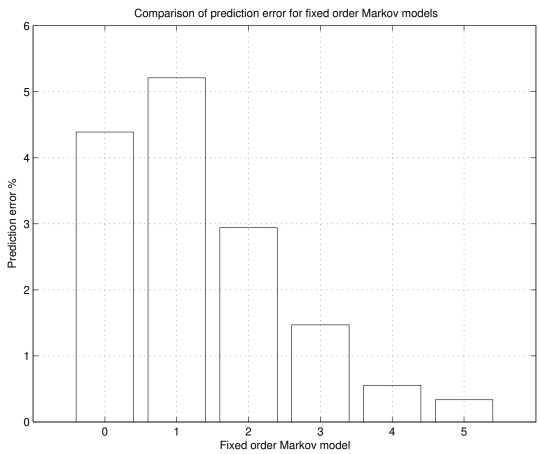

Figure 2: Prediction error between $P^{\exists}(w)$ (computed) and $P_e^{\exists}(w)$ (observed) for a serial episode using Markov models of fixed order: $0, 1, 2, 3, 4, 5$

3.2 Interpolated model

We used the χ^2-confidence based interpolation method introduced in the GLIMMER gene finding algorithm in [13] for computing the $\lambda(c_i)$ in the equation for the conditional probability in the interpolated Markov model

$$\begin{aligned} P_{IMM}(a_i|c_k) \quad &= \quad \lambda(c_k) \cdot P(a_i|c_k) + \\ & \qquad [1 - \lambda(c_k)] \cdot P_{IMM}(a_i|c_{k-1}). \end{aligned}$$

Algorithm 1: Computation of interpolation parameters λ

input : $n(c_i), n(c_i, a_j), N$

output: $\lambda(c_i)$

begin

  for $i = 1$ ***to*** k **do**

    $th = (N - i + 1)P(c_i)$;

    if $n(c_i) \leq th$ **then**

      $\lambda(c_i) = 1$

    else

      $chisquare = 0$;

      for $j = 1$ ***to*** $|\mathcal{A}|$ **do**

        $chisquare+ = \frac{(n(c_i,a_j) - n(c_i) * P_{IMM}(a_j|c_i))}{n(c_i) * P_{IMM}(a_j|c_i)}$;

      $p = gammp(chisquare, \mathcal{A} - 1)$;

      if $p < 0.5$ **then**

        $\lambda(c_i) = 0$

      else

        $\lambda(c_i) = \frac{p * n(c_i)}{th}$

        for $j = 1$ ***to*** $|\mathcal{A}|$ **do**

          $P_{IMM}(a_j|c_i) = \lambda(c_i)\frac{n(c_i,a_j)}{n(c_i)} +$
          $[1 - \lambda(c_k)] \cdot P_{IMM}(a_j|c_{i-1})$

end

The algorithm takes as its input the following parameters: $n(c_i)$ the frequency of a context of length c_i, $n(c_i, a_j)$ the frequency of the the string $c_i a_j$ where $a_j \in \mathcal{A}$ and N, the length of the training set. The function $gammp(chi, df)$ computes the probability that the χ^2 random variable is smaller that $chisquare$ i.e. it computes the cumulative distribution function of the χ^2 for $\mathcal{A} - 1$ degrees of freedom. The GLIMMER system used a fixed value for $th = 400$. We interpreted the threshold as an expected number of occurrences of a context c_i of length i as a string in the training set for memoryless source $\mathbf{E}[n(c_i)] = (N - i + 1)P(c_i)$. Thus, we set $th = (N - i + 1)P(c_i)$, where $P(c_i)$ is the probability of the context in the order 0 Markov model. Alternatively we could use $th = (N - i + 1)P(c_i) - \sqrt{\mathbf{Var}[n(c_i)]}$. The following two sections use the IMM, computed by Algorithm 1.

3.3 Helicobacter pylori

In this experiment we compared the fixed order 5 with IMM of order 5. We used the same fixed order models trained on *Haemophilius influenzea* as in the previous experiment and we also built an IMM using *Haemophilius influenzea*. We tested the performance of both models on *Helicobacter pylori* as a testing source. The purpose of this experiment was to check whether the IMM outperforms the fixed order model. Because we test the model on a different organism (*Helicobacter pylori*) than the training (*Haemophilius influenzea*) we expected the IMM to perform better because of its smoothing properties whereas for the fixed order we expected it to suffer from over-fitting. Also we did not expect a significant improvement in accuracy of IMM because the training set of size (1,830,025) was sufficiently large to find all context strings. We set a serial episode $S = CCGT$ and for each model for $k = 0, 1, 2, 3, 4, 5$ and IMM we compared $P^{\exists}(w)$ with the observed $P_e^{\exists}(w)$ for $w = [5, 20]$ by computing the prediction error given in Equation 4. The results, shown as a bar graph in Figure 3 confirm our expectations and the IMM performed slightly better that order 5. Also order 1 turned out to be the winner probably because there is a difference in the structure of DNA of *Helicobacter pylori* and *Haemophilius influenzea* and the order 1 capture the necessary structure without over-fitting.

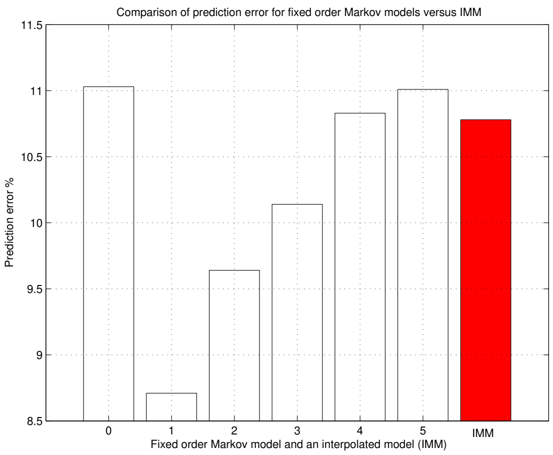

Figure 3: Prediction error between $P^{\exists}(w)$ (computed) and $P_e^{\exists}(w)$ (observed) for a serial episode using Markov models of fixed order: $0, 1, 2, 3, 4, 5$ and IMM order 5

3.4 Human chromosome 22

In this experiment we wanted to check whether the order 5 IMM outperforms the fixed order 5 when both are trained from sparse training data. To accomplish it we chose the first segment of *Human chromosome 22* of length $234,227$ as a training set and we tested both models on the second segment of the same chromosome of length $3,661,561$. We set a serial episode $S = CCGT$ and computed $P^{\exists}(w)$ for both models and compared to the observed $P_e^{\exists}(w)$ for $w = [5, 20]$. We plotted the results in Figure 4 to show the shape of the curves and we computed the prediction error. Again the IMM performed slightly better with the prediction error of order 11% versus fixed order 13%. From the figure we can see that IMM better approximates the observed $P_e^{\exists}(w)$.

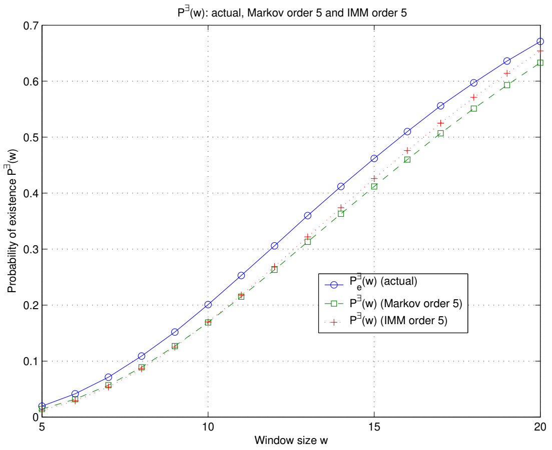

Figure 4: Observed $P_e^{\exists}(w)$ and computed $P^{\exists}(w)$ for a serial episode using fixed order 5 Markov and IMM order 5

4 Conclusions and extensions

We presented a new method for a reliable identification of significant episodes in Markov source of an arbitrary order. As a measure of significance we used $\Omega^{\exists}(n, w)$ the number of windows in which the episode occurred in the event stream. We proved that $\Omega^{\exists}(n, w)$ is a sum of so called φ-mixing random variables and obeys the central limit theorem, which leads to a computational formula for the threshold $\tau_u(w)$ for discovering significant episodes. We proposed to use non-uniform Markov models with the threshold mechanism because of their flexibility for modeling a wide variety of event sources. In particular we compared the interpolated Markov model with the fixed order Markov model in experiments conducted on DNA sequences. We showed that the IMM slightly outperforms the fixed order model in terms of prediction accuracy. We also showed that for DNA source the use of Markov models outperforms memoryless models in terms of accuracy in predicting occurrences of episodes even though Markov source can be susceptible to over-fitting. The drawback of using the Markov models is the high computational cost of computing the threshold. This could be overcome by using a combination of the a memoryless model and a Markov model. In such a technique we could use the Markov model for small values of w where the accuracy of the prediction would be crucial and we could use the Bernoulli model for large w, where $P^{\exists}(w)$ for both the Markov and the Bernoulli model converge to 1. One of the future extensions of this work could be to provide an efficient algorithm for computing $P^{\exists}(w)$ or to further improve the performance of non-uniform models.

References

[1] A. Apostolico and M. Atallah (2002), Compact Recognizers of Episode Sequences, *Information and Computation*, 174, 180-192.

[2] M. Atallah, R. Gwadera and W. Szpankowski, Detection of significant sets of episodes in event sequences, In *Fourth IEEE International Conference on Data Mining*, pages 67-74, Brighton UK.

[3] R. K. Azad and M. Borodovsky, Effects of choice of DNA sequence model structure on gene identification accuracy, *Bioinformatics* 2004.

[4] P. Billingsley (1986), *Probability and measure*, John Wiley, New York.

[5] L. Boasson, P. Sequels, I. Guessarian, and Y. Matiyasevich (1999), Window-Accumulated Subsequence Matching Problem is Linear, *Proc. PODS*, 327-336.

[6] S. Brin, R. Motwani, C. Silverstein, *Beyond Market Baskets: Generalizing Association Rules to Correlations*, SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona

[7] Stanley F. Chen, Joshua Goodman, *An Empirical Study of Smoothing Techniques for Language* , Proceedings of the Thirty-Fourth Annual Meeting of the Association for Computational Linguistics, 1996

[8] Delcher AL, Harmon D, Kasif S, White O, Salzberg SL, Improved microbial gene identification with GLIMMER *Nucleic Acids Research*, Vol. 27, No 23, 1999.

[9] R. Gwadera, M. Atallah, and W. Szpankowski, Reliable detection of episodes in event sequences, In *Third IEEE International Conference on Data Mining*, pages 67-74, Melbourne Florida.

[10] J. Han, J. Pei, Y. Yin, R. Mao, Mining Frequent Patterns without Candidate Generation: A

Frequent-Pattern Tree Approach, *Data Mining and Knowledge Discovery*, 8, 53-87, 2004

[11] H. Mannila, H. Toivonen, and A. Verkamo (1997), Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery*, 1(3), 241-258.

[12] M. Régnier and W. Szpankowski (1998), On pattern frequency occurrences in a Markovian sequence, *Algorithmica*, 22, 631-649.

[13] Salzberg SL, Delcher AL, Kasif S, White O, Microbial gene identification using interpolated Markov models, *Nucleic Acids Research*, Vol. 26, No 2, 1998.

[14] W. Szpankowski (2001), *Average Case Analysis of Algorithms on Sequence*, John Wiley, New York.

[15] M. J. Weinberger, J. Rissanen, and M. Feder, A universal finite memory source, *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 643–652, May 1995. 48

[16] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, The context-tree weighting method: Basic properties, *IEEE Trans. Inform. Theory*, vol. IT-41, pp. 653–664, May 1995 John Wiley, New York.