A Unified Approach to Computing Distributions Associated with Hidden State Sequences

Donald E. K. Martin and John A. D. Aston

Abstract— **In this paper a method is given for computing distributions of statistics of hidden state sequences. The method applies to any situation for which the conditional distribution of states given observations may be modeled by a factor graph with factors that depend on current and past states but not future ones. Model structure is exploited to develop a Markov chain that facilitates efficient computation of distributions. The methodology may be used for discrete hidden state sequences perturbed by noise and/or missing values, and for state sequences that serve to classify observations. Two detailed examples are given to illustrate the computational procedure.**

Index Terms—**Auxiliary Markov chain, classification, deterministic finite automaton, distribution of pattern statistics, hidden state sequences.**

I. Introduction

An important problem in structured learning is the prediction of values of a sequence of hidden states, conditioned on observed data. A typical choice for this task is to maximize the conditional probability of states given the data and the model (producing the *Viterbi sequence*; see [1]). Whereas this method of prediction is optimal in many situations, it may not be so if one is mainly interested in inference on statistics of the hidden states. Determining sampling distributions associated with statistics of hidden states is the topic of the present paper.

In classification, runs in hidden labels correspond to regimes in the observed data. Examples of applications where statistics of regimes are of interest are [2], in the context of business cycle analysis, and [3] and [4], which classified DNA nucleotides to locate genes or *CpG* islands, respectively. Hidden states can also be the true values of noisy data with or without missing values. In that situation, any pattern of interest in the observed data is of relevance in the hidden states as well. Prototypical applications include microarray experiments, which frequently have noisy data for various experimental reasons [5], and noisy time series data [6].

Manuscript received July 2, 2008. This work was supported in part by the National Science Foundation Grant DMS-0805577.

Donald E. K. Martin is with the Department of Statistics, North Carolina State University, Raleigh, NC 27695, USA (phone: 919-515-1936; fax: 919-515-1169; e-mail: martin@ stat.ncsu.edu).

John A. D. Aston is with CRISM, Department of Statistics, Warwick University, UK (e-mail: j.a.d.aston@warwick.ac.uk).

In [7] an illustration is given of the pitfalls of basing inference for functions of hidden states on the Viterbi sequence, and a method based on Markov chains is presented to compute exact sampling distributions of statistics of patterns in states of hidden Markov models. In this paper that methodology is extended to more general settings, as it is applied to both discriminative and generative models for which the conditional distribution of states given observations can be represented as a factor graph with factors that depend on current and past states but not future ones. This is important because it supplies a way to analyze data from a variety of models, supplanting the need to develop new methodology for each individual case. In addition, the development of the auxiliary Markov chain used for computations is made more efficient through the use of deterministic finite automata, in the sense that in some cases the number of states needed in the auxiliary Markov chain is reduced.

After background information is given on the model framework, details of the computational procedure are presented, along with two examples. The final section is a conclusion.

II. The Model

Let $\mathbf{o} = (o_1, \ldots, o_T)$ be a sequence of observations, and $\mathbf{s} = (s_1, \ldots, s_T)$ the corresponding hidden states, with each s_t from a finite state space Σ. The conditional distribution of states given observations is assumed to factor according to

$$p_S(\mathbf{s}|\mathbf{o}) = \frac{1}{Z(\mathbf{o})} \Psi_0(\tilde{s}_m, \mathbf{o}) \prod_{t=m+1}^{T} \Psi_1(s_t^{(h)}, s_t, \mathbf{o}). \quad (1)$$

In (1), $s_t^{(h)}$ are *history* variables for time (or sequence location) t that include s_{t-m} and some subset of $s_{t-m+1:t-1}$ ($s_{a:b} = s_a, s_{a+1}, \ldots, s_b$, $a \le b$), so that the sequence has m-th order dependence . Also, $\tilde{s}_m \equiv s_{1:m}$ are *initialization* states. (In general, $\tilde{s}_t \equiv s_{t-m+1:t}$, $t = m, m+1, \ldots, T$). The factors Ψ_0 and Ψ_1 are non-negative *potential* functions that indicate compatibility between the states that are their arguments and some subset (which is possibly empty) of the observations $\mathbf{o}$. These functions are (or contain) parameters that are estimated from data. The *partition function* $Z(\mathbf{o})$ is a normalization constant, so that (1) defines a probability distribution.

The representation (1) is fairly general in that conditional probabilities from various graphical models take that form. The model is a special case of a *conditional random field* (CRF) [8]. When $s_t^{(h)} = s_{t-1}$, the model is a *linear chain CRF*, and a linear chain CRF with vector states is a *dynamic conditional random field* [9]. The model (1) can also be applied to generative models for the joint distribution of states and observations. For example, a *hidden Markov model (HMM)*

$$p(\mathbf{s},\mathbf{o}) = \underbrace{g_1(s_1)\gamma(o_1|s_1)}_{\Psi_0(s_1,o_1)} \prod_{t=2}^{T} \underbrace{g_2(s_t|s_{t-1})\gamma(o_t|s_t)}_{\Psi_1(s_{t-1},s_t,o_t)}$$

has a conditional distribution $p(\mathbf{s}|\mathbf{o})$ given by (1), with $m=1$, potential functions Ψ_0 and Ψ_1 as indicated, and $Z(\mathbf{o}) = \sum_{\mathbf{s}} p(\mathbf{s},\mathbf{o})$. *Dynamic Bayesian networks* [10] may be represented as an HMM with vector states, and thus also satisfy (1). Other models may also be represented as (1) as well.

The computation of $Z(\mathbf{o})$ is facilitated by the *backward variables*, defined by $\beta_T(\tilde{s}_T,\mathbf{o}) \equiv 1$ for all $\tilde{s}_T$, and $\beta_t(\tilde{s}_t,\mathbf{o}) \equiv \sum_{s_{t+1:T}} \prod_{\tau=t+1}^{T} \Psi_1(s_\tau^{(h)}, s_\tau, \mathbf{o})$ for $t = T-1, T-2$, …, m, which lend themselves to recursive computation: $\beta_t(\tilde{s}_t,\mathbf{o}) \equiv \sum_{s_{t+1}} \beta_{t+1}(\tilde{s}_{t+1},\mathbf{o})\Psi_1(s_{t+1}^{(h)}, s_{t+1}, \mathbf{o})$. Using the backward variables, $Z(\mathbf{o})$ is obtained as $Z(\mathbf{o}) = \sum_{\tilde{s}_m} \Psi_0(\tilde{s}_m)\beta_m(\tilde{s}_m)$.

The model (1) has a Markovian property that is useful. For $t \geq m+1$,

$$p_S(s_t|s_{1:t-1},\mathbf{o}) = \frac{p_S(s_{1:t}|\mathbf{o})}{p_S(s_{1:t-1}|\mathbf{o})} = \frac{\sum_{s_{t+1:T}} p_S(\mathbf{s}|\mathbf{o})}{\sum_{s_{t:T}} p_S(\mathbf{s}|\mathbf{o})}$$

$$= \frac{\Psi_1(s_t^{(h)}, s_t, \mathbf{o})\beta_t(\tilde{s}_t,\mathbf{o})}{\beta_{t-1}(\tilde{s}_{t-1},\mathbf{o})} = p_S(s_t|\tilde{s}_{t-1},\mathbf{o}). \qquad (2)$$

Thus conditioned on the observations, transition probabilities in the state sequence given all previous states depend on only the last m states. This Markovian structure is exploited in the next section to develop an auxiliary Markov chain (AMC) $\{Y_\tau\}_{\tau=m}^{T}$ that eases the computation of conditional distributions of statistics of $\mathbf{s}$.

III. Computation of Distributions in Hidden State Sequences

Terminology related to patterns is now given as a prelude to a description of the computational algorithm.

A. Preliminary notation and terminology

The *alphabet* Σ is a nonempty finite set whose elements are called *letters*. A *word* w over Σ is a sequence of letters, and its length $|w|$ is the number of letters forming the word. The word of length zero is denoted by ε. Σ^k, $k \geq 0$ is the set of all words formed by taking k letters from Σ, with $\Sigma^0 = \{\varepsilon\}$. The set of all words over Σ is denoted by Σ^*; $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$ Let $\Sigma^{\leq j} \equiv \{w \in \Sigma^* : |w| \leq j\}$. For words u and v over Σ, the *concatenation* uv is the word formed by adding v to the end of u. A word v is called a *suffix* of u if $u = wv$ for some $w \in \Sigma^*$, and v is a *prefix* of u if $u = vw$, $w \in \Sigma^*$. For $w \in \Sigma^*$, define $w_{(j)}$ to be the longest $v \in \Sigma^{\leq j}$ that is a suffix of w. Finally, a *pattern* L is any subset of Σ^*.

B. Markov chain embedding

Distributions associated with patterns and statistics in a state sequence may be computed through an *auxiliary Markov chain* (AMC) that sequentially processes each random variable of the sequence and returns probabilities for each of the states of the chain. This technique was formalized in [11], and has been used extensively, including in [12] and [13] to compute generalized *sooner* and *later* waiting time distributions of collections of patterns that have to occur pattern specific numbers of times, and, as stated earlier, in [7], where distributions of patterns in state sequences of HMMs are computed.

The work of the latter reference is extended here to more general factor graphs of the form (1). The AMC that is formed for computation is of the form (Φ,θ), where θ is a vector holding the statistic(s) of interest, and Φ is a set of supplementary variables needed to form the Markov chain. Φ consists of the variable q that gives progress into the pattern of interest, an m-tuple $\tilde{s}_t$ that holds the last m values of $s_{1:t}$, and possibly other variables, though no other variables are used in Φ in the remainder of this paper. Deterministic finite automata assist with setting up possible values of q in an efficient manner.

A *deterministic finite automaton (DFA)* D is a five-tuple, $D = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of automaton states, Σ an alphabet, $\delta : Q \times \Sigma \to Q$ a transition function, q_0 an initial state, and F a set of final states. The transitions of a DFA may be represented as a directed graph with labeled edges, where an arrow labeled with the character $a \in \Sigma$ connecting a state q to a state r is drawn provided that $\delta(q,a) = r$.

A state $r \in Q$ is *accessible* if there is a path from q_0 to r, and *coaccessible* if there exists a path from r to one of the final states. The automaton *recognizes* all words that can be formed by concatenating from left to right the labels of edges visited by any path over its transition graph that starts at q_0 and ends at a state of F.

In an Aho-Corasick automaton [14], $q_0 = \varepsilon$, states of Q represent prefixes of words of the pattern, and final states represent matches with words of the pattern. In some cases, an Aho-Corasick automaton will not be minimal, in the sense that the pattern in question can be recognized with fewer automaton states. For improved efficiency we use the

procedure of [15], which involves determining equivalence classes for states Q of D based on its transitions, and defining a minimal automaton $D_{\min}$ with states that correspond to equivalence classes. Values of q correspond to states of $D_{\min}$.

An m-th order automaton $D^{(m)}$ ([16]) may be formed by coupling words of $\Sigma^{\le m}$ with states of $D_{\min}$, and then deleting any states that are not accessible and coaccessible. Elements $(q, \tilde{s}_t)$ of Φ are then states of $D^{(m)}$.

The variables of Φ and of θ at any time t are uniquely determined from $s_{1:t}$, and thus, so are initial and transition probabilities of the AMC $\{Y_\tau\}_{\tau=m}^{T}$. If (q_m, θ_m) are the values of q and θ determined from $\tilde{s}_m$, then

$$P[Y_m = (q_m, \tilde{s}_m, \theta_m)] = p_S(\tilde{s}_m | \mathbf{o})$$
$$= \sum_{s_{m+1:T}} p_S(\mathbf{s}|\mathbf{o}) = \left[Z(\mathbf{o})^{-1} \right] \Psi_0(\tilde{s}_m, \mathbf{o}) \beta_m(\tilde{s}_m, \mathbf{o}), \quad (3)$$

and if $(q', \tilde{s}_{t-1}, \theta') \to (q, \tilde{s}_t, \theta)$ with the occurrence of s_t at time t,

$$P[Y_t = (q, \tilde{s}_t, \theta) | Y_{t-1} = (q', \tilde{s}_{t-1}, \theta')] = p(s_t | \tilde{s}_{t-1}). \quad (4)$$

Equations (3) and (4) lead to a recursive method for computing $\psi_t(q, s, \theta) \equiv P[Y_t = (q, s, \theta)]$, $t = m, \ldots, T$. If ξ_m holds the initial distribution of Y_m, and the matrix Ω_τ has probabilities for transitions of $Y_{\tau-1}$ to Y_τ, $\tau = m+1, \ldots, T$, then

$$\psi_t = \xi_m \sum_{\tau=m+1}^{t} \Omega_\tau, \quad (5)$$

which lends itself to recursive computation:

$$\psi_m = \xi_m,$$
$$\psi_\tau = \psi_{\tau-1} \Omega_\tau, \ \tau = m+1, \ldots, t.$$

The last equation can be written as

$$\psi_{\tau,j} = \psi_{\tau-1} \Omega_{\tau,j}, \quad (6)$$

where $\psi_{\tau,j}$ is the probability of the j-th state of the AMC when states are ordered (j corresponds to some $(q, \tilde{s}_t, \theta)$), and $\Omega_{t,j}$ is the j-th column of Ω_t. Equation (6) is the basis of a dynamic program for computing probabilities for the AMC lying in its various states. From $\psi_t(q, \tilde{s}_t, \theta)$, one obtains probabilities for θ by summing over $(q, \tilde{s}_t)$.

IV. Examples

Two examples are now given to make the computational algorithm more transparent.

A. Distribution of Chi motif

Assume that an observed DNA sequence $\mathbf{o}$ is recorded with error, and one is interested in determining the distribution of the *Chi motif* $L_1 = G*TGGTGG$ ($* \in \Sigma = \{A, C, G, T\}$) in the true underlying values $\mathbf{s}$, conditional on $\mathbf{o}$. Recognition of Chi sites by RecBCD is involved in repair of broken DNA, and thus this motif occurs frequently in certain genomes, such as *E. Coli* [17]. The conditional distribution of states given observations is modeled through an HMM:

$$p_S(\mathbf{s}|\mathbf{o}) = \left[Z(\mathbf{o}) \right]^{-1} g_1(s_1) \gamma(o_1 | s_1) \prod_{t=2}^{T} g_2(s_t | s_{t-1}) \gamma(o_t | s_t),$$

where $Z(\mathbf{o}) = \sum_{\mathbf{s}} g_1(s_1) \gamma(o_1 | s_1) \prod_{t=2}^{T} g_2(s_t | s_{t-1}) \gamma(o_t | s_t)$. The recursion to obtain the backward variables β_t is well known [1], and $Z(\mathbf{o})$ is easily obtained from the backward variables. From (2), the conditional probability of s_1 given $\mathbf{o}$ is

$$p_S(s_1 | \mathbf{o}) = \frac{g_1(s_1) \gamma(o_1 | s_1) \beta_1(s_1)}{\sum_{s_1} g_1(s_1) \gamma(o_1 | s_1) \beta_1(s_1)}, \quad (7)$$

and from (3), conditional transition probabilities are

$$p_S(s_t | s_{t-1}, \mathbf{o}) = \frac{g_2(s_t | s_{t-1}) \gamma(o_t | s_t) \beta_t(s_t)}{\beta_{t-1}(s_{t-1})}. \quad (8)$$

The model parameters g_1, g_2, and γ may be estimated by the Baum-Welch method (see [1]). The minimal DFA $D = (Q, \Sigma, \delta, \{0\}, \{9\})$ that recognizes L_1 is depicted in Fig. 1. As pattern prefixes are grouped in states of D (for example, word prefixes *GAT*, *GCT*, *GGT* and *GTT* are all represented by state 4 of D), the minimal DFA has only 10 states, instead of the 30 states (word prefixes) of an Aho-Corasick automaton, so that the AMC has 1/3 the states, compared with its formation based on word prefixes. For Fig. 1, it is assumed that counting is *overlapping*, where all occurrences of L_1 are counted. Under *non-overlapping counting*, transitions from state 9 are the same as those from state 0.

Since for an HMM the state sequence $\mathbf{s}$ has first-order dependence, m is set to one and the minimal first-order DFA (not shown) also carries the last state value.

The AMC $\{Y_\tau\}_{\tau=1}^{T}$ has states of the form (q, s_t, θ), where θ is the number of occurrences of L_1. To illustrate the correspondence between the AMC and $\mathbf{s}$, for the sequence $s_{1:12} = ACGATGGTGGTG$, $(Y_1, \ldots, Y_{12}) = ((0, A, 0), (0, C, 0),$

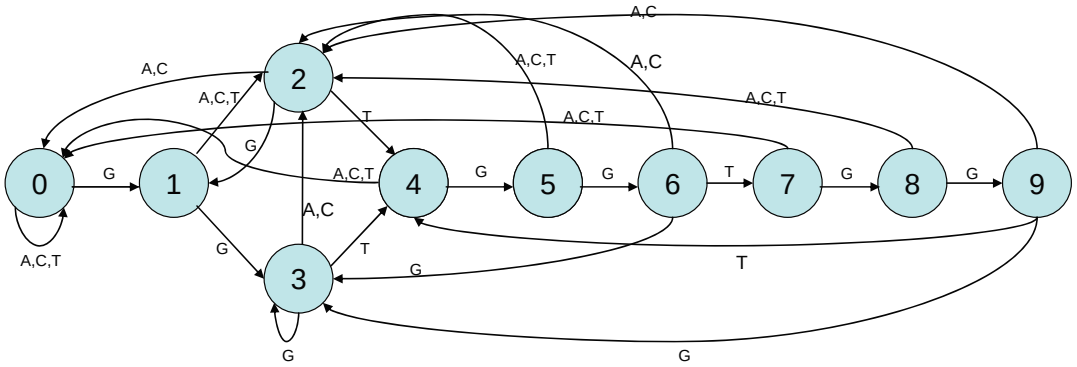

Fig. 1. Minimal DFA D that recognizes pattern $L_1 = G*TGGTGG$, where $* \in \{A,T,G,C\}$ (overlapping counting)

$(1,G,0),(2,A,0),(4,T,0),(5,G,0),(6,G,0),(7,T,0),(8,G,0),$
$(9,G,1),(4,T,1),(5,G,1))$ under overlapping counting, with $(Y_{11},Y_{12}) = ((0,T,1),(1,G,1))$ under non-overlapping counting. Recall that $p_S(s_1|\mathbf{o})$ and $p_S(s_t|s_{t-1},\mathbf{o})$ are respectively computed from (7) and (8), and note that for an HMM, probabilities for s_t depend on o_t, but no other observations. The initial distribution of Y_1 is $\psi_1(0,a,0) = p_S(a|\mathbf{o})$ for $a \in \{A,C,T\}$, and $\psi_1(1,G,0) = p_S(G|\mathbf{o})$. If $I(\upsilon)=1$ if υ is true, and zero otherwise, transition probabilities for Y_t, $t \geq 2$ may be represented as:

$$\psi_t(0,a,\theta) = \sum_{b=A,C,T}\left[\psi_{t-1}(0,b,\theta)+\psi_{t-1}(2,b,\theta)I(a\neq T)\right]p_S(a|b,\mathbf{o}) + \sum_{q=4,7}\psi_{t-1}(q,T,\theta)\,p_S(a|T,\mathbf{o}),\ a\in\{A,C,T\};$$

$$\psi_t(1,G,\theta) = \sum_{q=0,2}\sum_{a=A,C,T}\psi_{t-1}(q,a,\theta)p_S(G|a,\mathbf{o});$$

$$\psi_t(2,a,\theta) = p_S(a|G,\mathbf{o})\Big[\sum_{q=1,5,8}\psi_{t-1}(q,G,\theta) + \sum_{q=3,6,9}\psi_{t-1}(q,G,\theta)I(a\neq T)\Big],\ a\in\{A,C,T\};$$

$$\psi_t(3,G,\theta) = \sum_{q=1,3,6,9}\psi_{t-1}(q,G,\theta)p_S(G|G,\mathbf{o});$$

$$\psi_t(4,T,\theta) = \sum_{a=A,C,T}\psi_{t-1}(2,a,\theta)p_S(T|a,\mathbf{o}) + \sum_{q=3,9}\psi_{t-1}(q,G,\theta)\,p_S(T|G,\mathbf{o});$$

$$\psi_t(q,G,\theta) = \psi_{t-1}(q-1,T,\theta)\,p_S(G|T,\mathbf{o}),\ q=5,8;$$

$$\psi_t(6,G,\theta) = \psi_{t-1}(5,G,\theta)\,p_S(G|G,\mathbf{o});$$

$$\psi_t(7,T,\theta) = \psi_{t-1}(6,G,\theta)\,p_S(T|G,\mathbf{o});$$

$$\psi_t(9,G,\theta) = \psi_{t-1}(8,G,\theta-1)\,p_S(G|G,\mathbf{o}),\ \theta\geq 1.$$

The distribution of the number of occurrences of L_1 at time t is computed by summing $\psi_t(q,s_t,\theta)$ over q and s_t.

B. Success runs with gaps

This example is motivated by the problem of locating *CpG* islands, a segment of DNA in which the frequency of the *CG* dinucleotide is higher than in other regions. Whereas because of methylation there is a high chance that the *C* of *CG* will mutate to a *T*, upstream from a gene the methylation process is suppressed in a short region (a *CpG* island) of length 300-5,000 nucleotides so that *CG* pairs are more frequent [18]. Thus *CpG* islands are useful for identifying genes.

To identify *CpG* islands, [4] modeled the data generation process as an HMM, and used the Viterbi algorithm to segment the state sequence. Due to minimal length and gap restrictions that are frequently placed on islands ([19],[20]), [4] subjected the Viterbi sequence to post-processing procedures to ensure that islands are at least 500 nucleotides long, with gaps between islands of at least 500 nucleotides. In [7] it was shown that inference based on using the most likely sequence as if it is deterministically correct may not be optimal. The approach of this paper gives a way to compute the complete sampling distribution of statistics of *CpG* islands over all possible state sequences.

The joint distribution of the number of "islands" (runs with minimal gaps between them) and the number of observations falling in them (a statistic that was reported in [20]) is computed based on the CRF model (1), with $m=2$, history variables $s_t^{(h)} = (s_{t-2},s_{t-1})$, and states representing labels of whether or not the sequence is in an island.

The shared potential functions are typically modeled as

$$\Psi_1(s_{t-1},s_t,\mathbf{o}) = \exp\Big(\sum_j w_j f_j(s_{t-2},s_{t-1},s_t,\mathbf{o})\Big),$$

where $\{f_j\}$ are a set of real-valued *feature functions*, and $\{w_j\}$ are weights that are estimated using training data

(training of CRF parameters is discussed in [21]). Conditional random fields allow the flexibility to model multiple interacting or long-range features of the observations, models that are intractable when using generative models like HMMs. In the case of *CpG* islands, $C+G$ content in $x_{t-l+1},\ldots,x_t$ for some l could be stored in feature functions to assist in identifying islands.

Let $\theta=(\phi,\Delta)$, where ϕ is the number of islands and Δ is the number of observations in them, and let $\Sigma=\{a,b\}$, where b means that the observation is labeled as being in an island, and a that it is not in an island. The statistics (ϕ,Δ) are defined as follows.

- An island begins with the first b of a run of b's of length at least k and doesn't end until the last b preceding the first subsequent run of $a's$ of length at least g. If the last island has not ended before time T, it is considered as having ended at the last b of the sequence. Define, for $\phi\geq 1$, $\Delta\equiv\sum_{j=1}^{\phi}(e_j-\xi_j+1)$, where e_j (ξ_j) denotes the index of the end (beginning) of the j-th island, and $\Delta\equiv 0$ if $\phi=0$.

These definitions mimic counts of islands and observations in them when the labels are subjected to post-processing as in [4]. The distribution of Δ differs from that of the "sum of heads" statistic [22] because for Δ, non-*CpG* island states can be counted as being in an island after post-processing. To illustrate the definitions, if $k=5$ and $g=2$, for the sequence

$\mathbf{s}=aaababab\overbrace{bbbbbbbabbbb}^{island\ 1}aabbbaaaa\overbrace{bbbbbb}^{island\ 2}a$ of length $T=40$ we have $\phi=2$ and $\Delta=18$.

For this example $Q=\{0,1,\ldots,k,\ldots,k+g-1\}$, $q_0=\{0\}$, and $F=\{k,\ldots,k+g-1\}$, so that the automaton D recognizes words that end in an island. The states $0,1,\ldots,k-1$ indicate the length of the current run of the letter b. For final states of the form $k+\alpha$, $\alpha=0,\ldots,g-1$, α indicates the current length of the run of $a's$.

Taking the cross product of states of D with $\Sigma^{\leq 2}=\{\varepsilon,a,b,aa,ab,ba,bb\}$ and then trimming inaccessible states, one obtains the second-order DFA $D^{(2)}$ with initial state $(0,\varepsilon)$ that is depicted in Fig. 2.

The AMC $\{Y_\tau\}_{\tau=2}^{T}$ has states of the form $(q,\tilde{s}_t,\phi,\Delta)$, where $(q,\tilde{s}_t)$ are states of $D^{(2)}$, and (ϕ,Δ) are incremented with visits to $D^{(2)}$'s final states. The value of ϕ is incremented by 1 and Δ is incremented by k when state (k,bb) is entered from $(k-1,bb)$, indicating a k-run of $b's$. With each subsequent visit to one of the final states, Δ is incremented by one. To compensate for counting visits to non-island states when the island is being left, $g-1$ is subtracted from Δ on the transition from $(k+g-1,aa)$ to $(0,aa)$ (or from $(k+1,ba)$ to $(0,aa)$ when $g=2$).

For $\phi=1,\ldots,\varsigma+\lfloor[T-\varsigma(k+g)]/k\rfloor$ and $\Delta=1,\ldots,T$ ($\lfloor x\rfloor$ is the greatest integer less than or equal to x and $\varsigma=\lfloor T/(k+g)\rfloor$), let $\psi_t(q,\tilde{s}_t,\phi,\Delta)=P[Y_t=(q,\tilde{s}_t,\phi,\Delta)]$. The backward variables are obtained as described earlier. The initial distribution and transition probabilities for the state sequence $\mathbf{s}$ conditional on $\mathbf{o}$ are, respectively,

$$p_S(\tilde{s}_2|\mathbf{o})=\frac{\Psi_0(\tilde{s}_2)\beta_2(\tilde{s}_2,\mathbf{o})}{\sum_{\tilde{s}_2}\Psi_0(\tilde{s}_2)\beta_2(\tilde{s}_2,\mathbf{o})}, \tag{9}$$

$$p_S(s_t|\tilde{s}_{t-1},\mathbf{o})=\frac{\Psi_1(\tilde{s}_{t-1},s_t,\mathbf{o})\beta_t(\tilde{s}_t,\mathbf{o})}{\beta_{t-1}(\tilde{s}_{t-1},\mathbf{o})}. \tag{10}$$

We assume that $k,g>2$. To obtain ψ_t recursively, begin with

$$\psi_2(0,aa,0,0)=p_S(aa|\mathbf{o}),\ \psi_2(0,ba,0,0)=p_S(ba|\mathbf{o}),$$
$$\psi_2(1,ab,0,0)=p_S(ab|\mathbf{o}),\ \psi_2(2,bb,0,0)=p_S(bb|\mathbf{o});$$

initial probabilities for other states are zero. For $t=3,\ldots,T$,

$$\begin{aligned}\psi_t(0,aa,\phi,\Delta)&=\psi_{t-1}(0,ba,\phi,\Delta)p(a|ba,\mathbf{o})\\&+\psi_{t-1}(k+g-1,aa,\phi,\Delta+g-1)I(\Delta\geq k)p(a|aa,\mathbf{o})\\&+\psi_{t-1}(0,aa,\phi,\Delta)p(a|aa,\mathbf{o});\end{aligned}$$

$$\begin{aligned}\psi_t(0,ba,\phi,\Delta)&=\psi_{t-1}(1,ab,\phi,\Delta)p(a|ab,\mathbf{o})\\&+\sum_{j=2}^{k-1}\psi_{t-1}(j,bb,\phi,\Delta)p(a|bb,\mathbf{o});\end{aligned}$$

$$\begin{aligned}\psi_t(1,ab,\phi,\Delta)&=\psi_{t-1}(0,aa,\phi,\Delta)p(b|aa,\mathbf{o})\\&+\psi_{t-1}(0,ba,\phi,\Delta)p(b|ba,\mathbf{o});\end{aligned}$$

$$\psi_t(2,bb,\phi,\Delta)=\psi_{t-1}(1,ab,\phi,\Delta)p(b|ab,\mathbf{o});$$

$$\psi_t(j,bb,\phi,\Delta)=\psi_{t-1}(j-1,bb,\phi,\Delta)p(b|bb,\mathbf{o}),\ j=3,\ldots,k-1;$$

$$\begin{aligned}\psi_t(k,bb,\phi,\Delta)&=\left[\psi_{t-1}(k-1,bb,\phi-1,\Delta-k)I(\phi\geq 1,\Delta\geq k)\right]\\&\times p(b|bb,\mathbf{o})+\left[\psi_{t-1}(k,bb,\phi,\Delta-1)p(b|bb,\mathbf{o})\right.\\&\left.+\psi_{t-1}(k,ab,\phi,\Delta-1)p(b|ab,\mathbf{o})\right]I(\Delta\geq k+1).\end{aligned}$$

For $\Delta\geq k+1$,

$$\begin{aligned}\psi_t(k,ab,\phi,\Delta)&=\psi_{t-1}(k+1,ba,\phi,\Delta-1)p(b|ba,\mathbf{o})\\&+\sum_{\alpha=2}^{g-1}\psi_{t-1}(k+\alpha,aa,\phi,\Delta-1)p(b|aa,\mathbf{o});\end{aligned}$$

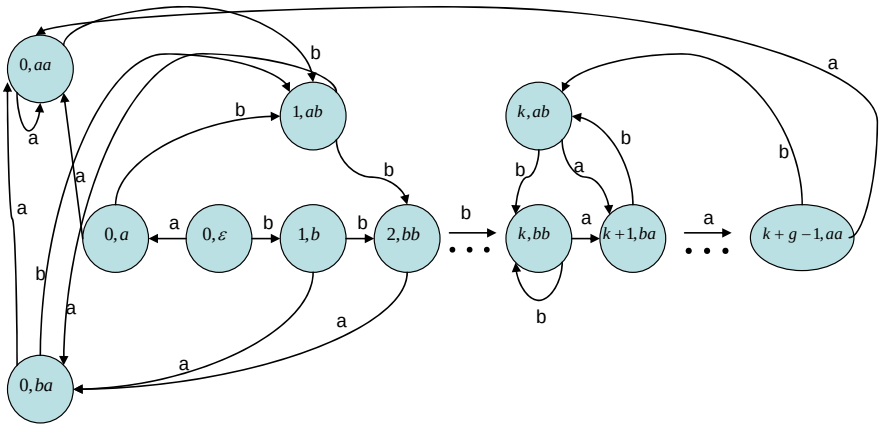

Fig. 2. Second-order DFA $D^{(2)}$ that recognizes words ending in an island

$$\psi_t(k+1, ba, \phi, \Delta) = \psi_{t-1}(k, bb, \phi, \Delta-1)\, p(a \,|\, bb, \mathbf{o}) + \psi_{t-1}(k, ab, \phi, \Delta-1)\, p(a \,|\, ab, \mathbf{o});$$

$$\psi_t(k+2, aa, \phi, \Delta) = \psi_{t-1}(k+1, ba, \phi, \Delta-1)\, p(a \,|\, ba, \mathbf{o});$$

$$\psi_t(k+\alpha, aa, \phi, \Delta) = \psi_{t-1}(k+\alpha-1, aa, \phi, \Delta-1)\, p(a \,|\, aa, \mathbf{o});$$

$(2 < \alpha \le g-1)$.

The joint distribution of $\theta = (\phi, \Delta)$ is obtained by summing $\psi_t(q, \tilde{s}_t, \phi, \Delta)$ over values of $(q, \tilde{s}_t)$.

V. Conclusion

In [7], a method was introduced for computing distributions of statistics of state sequences of hidden Markov models. This paper extends the methodology to more general classes of hidden state sequences. It is shown that distributions can be computed in graphical models, both discriminative and generative, which have factors that depend on current and past states, but not future ones. Thus the results have very general applications. In future work, the authors will apply the methodology to analyzing real data.

References

[1] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition, *Proceedings of the IEEE*, 77(2), 1989, pp. 257-289.

[2] J. D. Hamilton, "A new approach to the economic analysis of non-stationary time series and the business cycle," *Econometrica*, 57, 1989, pp. 357-384.

[3] A. Krogh, "Two methods for improving performance of a HMM and their application for gene finding," In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander, and A. Valencia, eds., *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, 1997, pp. 179-186, AAAI Press.

[4] R. Durbin, S. R. Eddy, A. Krogh, G. Mitchinson, *Biological Sequence Analysis: Probability Models of Proteins and Nucleic Acids.* Cambridge University Press, 1985, ch. 3.

[5] J. Blanchet, and M. Vignes, "Combined expression data with missing values and gene interaction network analysis: a Markovian integrated approach," *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering*, 2007, pp. 366-373.

[6] M. A. Tingley and L. McLean, "Detection of patterns in noisy time series," *The Canadian Journal of Statistics*, 29(2), 2001, pp. 217-237.

[7] J. A. D. Aston and D. E. K. Martin, "Distributions associated with general runs and patterns in hidden Markov models," *Annals of Applied Statistics*, 1(2), 2007, pp. 585-611.

[8] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings of the ICML*, 2001.

[9] C. Sutton, A. McCallum, and K. Rohanimanesh, "Dynamic conditional random fields," *Journal of Machine Learning Research*, 8, 2007, pp. 693-723.

[10] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence*, 5, 1995, pp. 142-150.

[11] J. C. Fu and M. V. Koutras, "Distribution theory of runs: A Markov chain approach," *Journal of the American Statistical Association*, 89, 1994, pp. 1050-1058.

[12] J. A. D. Aston and D. E. K. Martin, "Waiting time distributions of competing patterns in higher-order Markovian sequences," *Journal of Applied Probability*, 42, 2005, pp. 977-988.

[13] D. E. K. Martin and J. A. D. Aston, "Waiting time distributions of generalized later patterns," *Computational Statistics and Data Analysis*, 52(11), 2008, pp. 4879-4890.

[14] A. V. Aho and M. J. Corasick, "Efficient string matching: An aid to bibliography search," *Communications of the ACM*, 1975, 18(6), pp. 333-340.

[15] J. Hopcroft, "An $n \log n$ algorithm for minimizing states in a finite automaton," in *Proceedings of the International Symposium on the Theory of Machines and Computations*, 1971, pp. 189-196, Haifa, Israel.

[16] M. E. Lladser, "Minimal Markov chain embeddings of pattern problems," Proceedings of the 2007 Information Theory and Applications Workshop. University of California, San Diego.

[17] S. Robin, F. Rodolphe, and S. Schbath, *DNA, Words and Models: Statistics of Exceptional Words.* Cambridge University Press, 2005.

[18] A. Bird, "CpG-rich islands as gene markers in the vertebrate nucleus," *Trends in Genetics*, 3, 1987, pp. 342-347.

[19] D. Takai and P. A. Jones, "Comprehensive analysis of CpG islands in human chromosomes 21 and 22," *Proceedings of the National Academy of Science USA*, 2002, 99, pp. 3740-3745.

[20] Y. Wang and F. C. C. Leung, "An evaluation of new criteria for CpG islands in the human genome as gene markers," *Bioinformatics*, 20(7), 2004, pp. 1170-1177.

[21] S. V. N. Vishwanathun, N. N. Schraudolph, M. W. Schmidt, K. P. Murphy, "Accelerated training of conditional random fields with stochastic gradient methods," *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 969-976.

[22] D. E. K. Martin, "The exact joint distribution of the sum of heads and apparent size statistics in a 'tandem repeats finder' algorithm," *Bulletin of Mathematical Biology*, 68, 2006, pp. 2353-2364.