Revisiting Context-Tree Weighting for Bayesian Inference

I. Papageorgiou, U. of Cambridge, ip307@cam.ac.uk
L. Mertzanis, U. of Maryland, lambros@umd.edu
A. Panotopoulou, Dartmouth College, ath1na@bu.edu
I. Kontoyiannis, U. of Cambridge, yiannis@maths.cam.ac.uk
M. Skoularidou, U. of Cambridge, ms2407@cam.ac.uk

***Abstract*—We revisit the statistical foundation of the celebrated context tree weighting (CTW) algorithm, and we develop a Bayesian modelling framework for the class of higher-order, variable-memory Markov chains, along with an associated collection of methodological tools for exact inference for discrete time series. In addition to deterministic algorithms that learn the *a posteriori* most likely models and compute their posterior probabilities, we introduce a family of variable-dimension Markov chain Monte Carlo samplers, facilitating further exploration of the posterior. The performance of the proposed methods in model selection, Markov order estimation and prediction is illustrated through simulation experiments and real-world applications.**

The full paper describing this work is available online [11].

I. Introduction

For the analysis of discrete time series with significant temporal structure, higher-order Markov chains are frequently the most natural modelling choice. But the description of a full Markov chain of order d with values in a set of size m requires the specification of $m^d(m-1)$ parameters, which makes the use of full Markov chains problematic in practice: As has been often noted [4, 15, 23], the dimension of the parameter space grows exponentially with the memory length, and the resulting model class lacks modelling wealth and flexibility. This severely hinders the important goal of balancing the bias-variance tradeoff between more complex models that fit the data closely, and simpler models that generalize well.

To address these issues and to offer better solutions to many related scientific and engineering problems, numerous approaches have been developed. These include Raftery's mixture transition distribution (MTD) models [15, 16], Rissanen's tree sources [17, 18], probabilistic suffix tree (PST) models [22], and variable-length Markov chains (VLMC) [4].

In this work we introduce a Bayesian framework for variable-memory Markov models, and we develop algorithmic tools that lead to very effective and efficient *exact* inference.

In Section II we define a class of models $\mathcal{T}(D)$ as, e.g., in [25, 28], that admit natural representations as context trees. It contains all variable-memory Markov chains with values in an alphabet A, with memory no longer than D. A family of prior distributions $\pi_D(T;\beta)$ on models $T \in \mathcal{T}(D)$ is introduced, which penalizes more complex models by an exponential amount. Given a model T, we place independent Dirichlet priors $\pi(\theta|T)$ on the associated parameters θ. We refer to the models in $\mathcal{T}(D)$ equipped with this prior structure as *Bayesian context trees* (BCT).

In Section III-A we recall the context tree weighting (CTW) algorithm [25, 28] and we show that it can be used to not only evaluate the marginal likelihoods $P(x|T) = \int P(x|\theta, T)\pi(\theta|T)d\theta$ of observations x with respect to models T, but also the *prior predictive likelihood* [8] $P^*_D(x)$,

$$P^*_D(x) = \sum_{T\in\mathcal{T}(D)} \pi_D(T;\beta)P(x|T). \tag{1}$$

Given that the most basic obstacle to performing effective Bayesian inference is the inability to obtain the normalizing factor $P^*_D(x)$ of the posterior distribution [3, 8, 21], it is clear that the exact nature of the results produced by the CTW algorithm should facilitate the development of efficient methods for numerous core statistical tasks and related applications.

In Section III-B we describe the Bayesian context tree (BCT) algorithm, and we prove that it identifies the maximum *a posteriori* probability (MAP) model. This is a generalization of the "context tree maximizing" algorithm [27, 29]. And in Section III-C we show that a new algorithm, the k-BCT algorithm, can be used to identify the k *a posteriori* most likely tree models, for any $k \geq 1$. Despite the fact that $\mathcal{T}(D)$ is vast, consisting of doubly-exponentially many models in the memory length D, the complexity of all three algorithms is only linear in D and in the length of the observations x. But the complexity of k-BCT grows with k, so its applicability is more limited; see the relevant comments in Section VII.

In order to enable broader exploration of the posterior distributions $\pi(T|x)$ and $\pi(\theta, T|x)$, in Section III-D we develop a new family of variable-dimension Markov chain Monte Carlo (MCMC) algorithms, that obtain Metropolis-within-Gibbs samples from $\pi(\theta, T|x)$, as illustrated in Section V.

In Section IV we compare the model selection performance of the BCT framework with that of the corresponding VLMC and MTD methods, on both real and simulated data. We find that the BCT algorithm consistently performs at least as well as VLMC and MTD, and usually gives a better model fit.

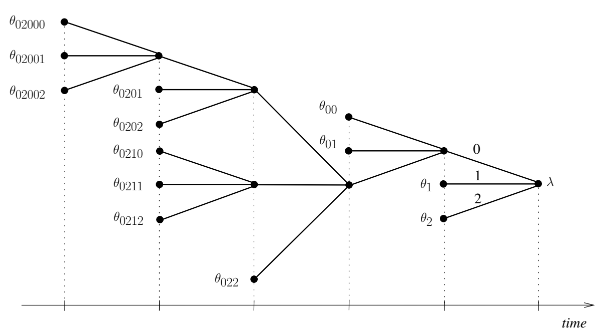
In Section VI we compare the natural predictor induced by the BCT framework with the predictors provided by the MTD, VLMC, SMC [10, 30] and CTF [23] methodologies. The BCT predictor has two significant advantages, which lead to superior performance. The first is that the *posterior predictive distribution* can be computed exactly, as $P^*_D(x_{n+1}|x_1,\ldots,x_n) = P^*_D(x_1,\ldots,x_{n+1})/P^*_D(x_1,\ldots,x_n)$, via the CTW algorithm. This way, the induced predictor is obtained by implicitly averaging over all models with respect to their exact posterior probabilities, thus avoiding the need to perform approximate model averaging via simulation or other numerical integration methods. The second advantage is that, because the CTW algorithm can be updated sequentially, so can the BCT predictor, so that it continues to "learn" from the data even past the training phase.

The Bayesian perspective adopted here is neither purely subjective, interpreting the prior and posterior as subjective descriptions of uncertainty pre- and post-data, respectively, nor purely objective, treating the resulting methods as simple black-box procedures [5]. For example, we think of the MAP model as the most accurate, data-driven representation of the regularities present in the data, but we also examine the frequentist properties of the resulting inferential procedures and evaluate them by simulation experiments.

Another point of view which naturally relates to the present development is Rissanen's celebrated Minimum Description Length (MDL) principle [9, 19, 20]. The MDL principle provides a broad operational foundation for statistical inference, as well as constructive tools and appealing metaphors for selecting prior distributions [5]. In particular, MDL considerations underpin much of the original work on the CTW algorithm [25, 27, 28] and our own choice of priors.

II. Bayesian context trees

Let $\{X_n\}$ be a dth order Markov chain, for some $d \geq 0$, with values in the alphabet $A = \{0, 1, \ldots, m-1\}$. The *model* describing $\{X_n\}$ as a variable-memory chain will always be represented by a tree as in the example below.



Let T be an m-ary tree of depth no greater than d, which is *proper*, in that, if a node in T is not a leaf, then it has exactly m children. For indices $i \leq j$, we write X_i^j for the vector of random variables $(X_i, X_{i+1}, \ldots, X_j)$ and $x_i^j \in A^{j-i+1}$ for a string $(x_i, x_{i+1}, \ldots, x_j)$ representing a realization of these random variables. The complete description of the distribution of $\{X_n\}$, in addition to the model T, requires the specification of a set of *parameters* $\theta = \{\theta_s \ ; \ s \in T\}$: Viewing T as the collection of its leaves, to every *context* $s \in T$ we associate a probability vector, $\theta_s = (\theta_s(0), \theta_s(1), \ldots, \theta_s(m-1))$. Then the likelihood induced by the model is,

$$P(x_1^n | x_{-d+1}^0) = \prod_{s \in T} \prod_{j \in A} \theta_s(j)^{a_s(j)}, \tag{2}$$

where the *count vectors* a_s are given by,

$$a_s(j) = \text{\# times symbol } j \in A \text{ follows context } s \text{ in } x_1^n. \tag{3}$$

Model prior. For $D \geq 0$ and $A = \{0, 1, \ldots, m-1\}$, let $\mathcal{T}(D)$ denote the collection of all (proper) tree models T on A with depth no greater than D. Given an arbitrary $\beta \in (0,1)$, we define the prior distribution,

$$\pi(T) = \pi_D(T; \beta) = \alpha^{|T|-1} \beta^{|T| - L_D(T)}, \tag{4}$$

where $\alpha = (1-\beta)^{1/(m-1)}$, $|T|$ is the number of leaves of T, and $L_D(T)$ is the number of leaves T has at depth D.

Lemma 2.1: For any $D \geq 0$ and any $\beta \in (0,1)$: $\sum_{T \in \mathcal{T}(D)} \pi_D(T; \beta) = 1$.

Prior on θ. Given a model $T \in \mathcal{T}(D)$, we place an independent Dirichlet prior with parameters $(1/2, 1/2, \ldots, 1/2)$ on each θ_s. In order to avoid cumbersome notation, in what follows we often write x for the entire time series x_1^n and suppress the dependence on its initial context x_{-d+1}^0.

Choice of β. Simple computation shows that $\pi_D(T; \beta)$ penalizes larger trees by an exponential amount as long as $\beta \geq 1/2$, and larger values of β make the penalization more severe. Also, for larger alphabet sizes, $\alpha = (1-\beta)^{1/(m-1)}$ becomes very close to 1 and the second factor dominates, an effect which is unintuitive and less desirable. Therefore, in practice we will always take $\beta \approx 1 - 2^{-m+1}$, so that $\alpha \approx 1/2$.

A useful property of the BCT framework is that the parameters θ can easily be integrated out, so that the *marginal likelihoods* $P(x|T)$ can be expressed in closed form.

Lemma 2.2: The marginal likelihood $P(x|T)$ of the observations x given a model T is,

$$P(x|T) = \int P(x|\theta, T)\pi(\theta|T) d\theta = \prod_{s \in T} P_e(a_s),$$

where the count vectors a_s are defined in (3) and the *estimated probabilities* $P_e(a_s)$ are defined by,

$$P_e(a_s) = \frac{\prod_{j=0}^{m-1} [(1/2)(3/2)\cdots(a_s(j) - 1/2)]}{(m/2)(m/2+1)\cdots(m/2 + M_s - 1)}, \tag{5}$$

where $M_s = a_s(0) + a_s(1) + \cdots + a_s(m-1)$, with the convention that any empty product is taken to be equal to 1.

III. Methodology

A. CTW: *The context tree weighting algorithm*

Recall the general version of the CTW algorithm [25, 28], where the *weighted probabilities* $P_{w,s}$ are computed, starting at the leaves and proceeding recursively towards the root, as,

$$P_{w,s} = \begin{cases} P_e(a_s), & \text{if } s \text{ is a leaf,} \\ \beta P_e(a_s) + (1-\beta)\prod_{j=0}^{m-1} P_{w,sj}, & \text{otherwise,} \end{cases}$$

where sj is the concatenation of context s and symbol j.

Theorem 3.1: The mixture probability $P_{w,\lambda}$ at the root λ computed by CTW is exactly the prior predictive likelihood of the observations, $P_{w,\lambda} = P_D^*(x)$ as in (1).

B. BCT: *The Bayesian context tree algorithm*

Like the CTW, the context tree maximizing (CTM) algorithm of [27, 29] can be generalized to non-binary alphabets and general β, with respect to the *maximal probabilities*,

$$P_{m,s} = \begin{cases} P_e(a_s), & \text{if } s \text{ is a leaf at depth } D, \\ \beta, & \text{if } s \text{ is a leaf at depth } < D, \\ \max\left\{\beta P_e(a_s),\ (1-\beta)\prod_{j=0}^{m-1} P_{m,sj}\right\}, & \text{otherwise.} \end{cases}$$

Theorem 3.2: For all $\beta \geq 1/2$, the tree T_1^* produced by the BCT algorithm is the MAP tree model.

C. k-BCT: *The top-k Bayesian context trees algorithm*

The k-BCT is one of the main novel contributions of this work. Although it is conceptually a natural generalization of BCT, its precise description is quite lengthy; it is given in the full version [11] and in [12].

Theorem 3.3: For any $\beta \geq 1/2$, the trees $T_1^*, T_2^*, \ldots, T_k^*$ produced by the k-BCT algorithm are the k *a posteriori* most likely tree models.

For a fixed maximal depth $D \geq 0$ and a fixed β, we observe that, for any model $T \in \mathcal{T}(D)$, the posterior probability $\pi(T|x)$ can easily be computed, using Lemma 2.2, $\pi(T|x) = P(x|T)\pi(T)/P_D^*(x)$. Also, conditional on a model T and observations x, the distribution $\pi(\theta|x,T)$ of the parameters can easily be seen to be the product over all $s \in T$ of Dirichlet distributions with parameters $(a_s(0)+1/2, a_s(1)+1/2, \ldots, a_s(m-1)+1/2)$.

D. *MCMC samplers*

RW sampler. The random walk (RW) sampler for $\pi(T|x)$ is a Metropolis-Hastings algorithm [21]. At each iteration, given the current model T, either a new branch of m leaves is added to a uniformly chosen node where such an addition is possible, with prob. 1/2, or a uniformly chosen existing branch of m leaves is removed, again with prob. 1/2. If the current tree is simply the root λ, then a branch is always added, and if it is the complete tree, then a branch is always removed. The corresponding proposal distribution is easy to compute. Then a standard Metropolis-Hastings accept/reject step ensures that the stationary distribution of this chain is indeed $\pi(T|x)$.

Jump sampler. This is a modification of the RW sampler, which, in addition to nearest neighbour moves, also allows for jumps to any one of the k most likely models. The computations of the proposal distribution and the accept/reject probabilities are only slightly more complex. This way we overcome the common difficulty of RW samplers to move between separated modes of multimodal posterior distributions.

Metropolis-within-Gibbs sampling. Being able to obtain MCMC samples $\{T^{(t)}\}$ for $\pi(T|x)$, and knowing the full conditional density $\pi(\theta|x,T)$ of the parameters explicitly as mentioned earlier, it is simple to obtain a corresponding sequence of Metropolis-within-Gibbs samples $\{(\theta^{(t)}, T^{(t)})\}$ for the posterior $\pi(\theta, T|x)$ jointly on models and parameters. This can be done by drawing a conditionally independent sample $\theta^{(t)} \sim \pi(\theta|x, T^{(t)})$ at each MCMC iteration step t.

IV. Model selection

We compare the model selection performance of the algorithms of Section III with the VLMC and MTD approaches. For VLMC we report results for the default value of its cut-off parameter K ("default-VLMC"), as well as for the values of K that optimize the BIC and the AIC score ("best-BIC-VLMC" and "best-AIC-VLMC"). For MTD we examine the models produced by both its 'single-matrix' version [15] MTD, and the 'multi-matrix' version [2] MTDg. For each data set we run the MTD algorithm for a range of possible depths D and choose the value that minimizes the corresponding BIC or AIC score. We refer to the resulting models as the best-BIC-MTD and best-AIC-MTD models. Similarly, we obtain the best-BIC-MTDg and best-AIC-MTDg models.

A. Simulated data. Consider $n = 1000$ samples generated from a 5th order variable-memory chain $\{X_n\}$ on the alphabet $A = \{0, 1, 2\}$ of $m = 3$ letters, with model given by the tree T shown in the example of Section II. With $D = 10$, and $\beta = 1 - 2^{-m+1} = 3/4$, the five *a posteriori* most likely models produced by the k-BCT algorithm are described in Figure 1. The MAP model is a depth-4 subtree of the true underlying model, with prior probability $\pi(T_1^*) \approx 2.9 \times 10^{-4}$ and posterior $\pi(T_1^*|x) \approx 0.2702$. The true model appears as T_4^*, with posterior $\pi(T_4^*|x) \approx 0.0213$. The sum of the posterior probabilities of the top 5 models is ≈ 0.4737.

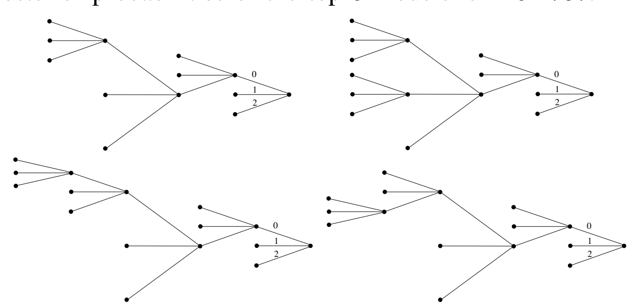


Fig. 1: The first, second, third, and fifth *a posteriori* most likely models $T_1^*, T_2^*, T_3^*, T_5^*$. The posterior odds $\pi(T_1^*|x)/\pi(T_i^*|x)$ with respect to the MAP model T_1^* are approximately 2.369, 5.358, 12.69, and 15.24, for $i = 2, 3, 4, 5$, respectively.

The default-VLMC model is the first tree shown in Figure 2; only about half of its nodes appear in the true underlying model. It has a worse BIC score and a better AIC score than the MAP model. The best-BIC-VLMC produces the small tree of depth 3 shown second in Figure 2, which is a subtree of the true model; it has a good BIC score and a poor AIC score. In sharp contrast, the best-AIC-VLMC produces a clearly overfitted model of depth 6, shown third in Figure 2. Although it has a poor BIC score, its AIC score is good, as expected.

Finally, the best-BIC-MTD and the best-BIC-MTDg both give $D = 0$, whereas the the best-AIC-MTD gives $D = 3$ and the best-AIC-MTDg gives $D = 2$. Their scores are generally quite a bit worse than those of the MAP model or the models produced by VLMC.

Overall, the BCT and the best-BIC-VLMC algorithms achieve the best performance. Their BIC and AIC scores are within $< 1\%$ of each other. More importantly, the BCT MAP model T_1^* has an additional full branch at depth 4 that reveals more of the true underlying structure, and the k-BCT identifies the true model as T_4^*.

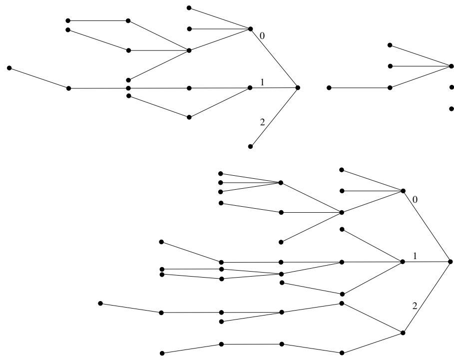


Fig. 2: The models produced by the default-VLMC, the best-BIC-VLMC and the best-AIC-VLMC methods.

Similar conclusions are drawn from numerous other examples [11]. The BCT framework consistently gives the most accurate model fit, with the best-BIC version of VLMC often giving similar results. The BCT algorithm is more efficient than either VLMC or MTD, typically by at least two orders of magnitude. The BCT framework has the additional advantage of identifying the top k *a posteriori* most likely models, together with their exact prior and posterior probabilities.

B. SARS-CoV-2 genome. The severe acute respiratory syndrome coronavirus 2, SARS-CoV-2, is the novel coronavirus responsible for the Covid-19 global pandemic. Here we examine the SARS-CoV-2 genome, which consists of $n = 29,903$ base pairs. We translate the four-letter DNA alphabet to $\{0, 1, 2, 3\}$ via the map (A,C,G,T) $\mapsto (0, 1, 2, 3)$.

The top 3 models obtained by the k-BCT algorithm with $D = 10$, $\beta = 1 - 2^{-m+1} = 7/8$ and $k = 3$ are shown as the first three trees in Figure 3. The MAP model has posterior $\pi(T_1^*|x) \approx 0.963$ and prior $\pi(T_1^*) \approx 4.3 \times 10^{-5}$. The sum of the posterior probabilities of these three models is ≈ 0.9994.

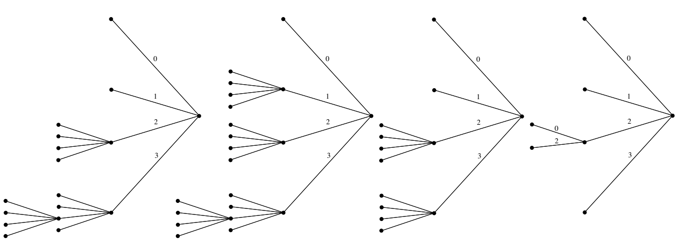


Fig. 3: First three trees: The *a posteriori* most likely models T_1^*, T_2^*, T_3^* obtained by the k-BCT algorithm on the SARS-CoV-2 genome. The posterior odds $\pi(T_1^*|x)/\pi(T_2^*|x)$ and $\pi(T_1^*|x)/\pi(T_3^*|x)$ are approximately 35.75 and 101.4, respectively. Last tree: The best-BIC-VLMC model; its AIC and BIC scores are both within 0.1% of those of T_1^*.

The best-BIC-VLMC model is the depth-2 subtree of T_1^* shown last in Figure 3, while both best-BIC-MTD and best-BIC-MTDg give $D = 1$. The AIC and BIC scores of all models are within 0.3% of each other. An interesting observation here is that k-BCT gives models of depth 3 with very high confidence. This may be because BCT finds evidence of the fact that DNA naturally gets encoded into triplets of bases to form codons that specify particular amino acids.

V. Posterior exploration

We consider the daily changes in Standard & Poor's (S&P) index, from January 2, 1928 until October 7, 2016, quantized to $m = 7$ values. Based on the resulting $n = 22900$ points x_i, the top $k = 5$ *a posteriori* most likely models obtained by the k-BCT algorithm with maximum tree depth $D = 260$ (corresponding to approximately one calendar year's trading days), are described in Figure 4.

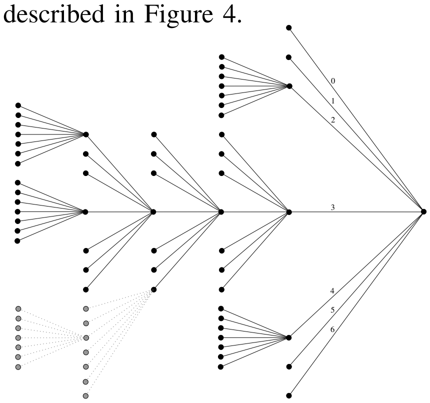


Fig. 4: The tree shown *without* the two dotted branches is the MAP model T_1^*. Its posterior $\pi(T_1^*|x) \approx 0.0174$ and its prior $\pi(T_1^*) \approx 5.7 \times 10^{-11}$. The whole tree shown is the fifth *a posteriori* most likely model T_5^*, and T_2^*, T_3^* and T_4^* were found to be small variations around T_1^* and T_5^*, all with depth 5. The corresponding posterior odds $\pi(T_1^*|x)/\pi(T_i^*|x)$, for $i = 2, 3, 4, 5$, are 1.094, 1.367, 1.496 and 2.467, respectively.

The shape of the MAP model contains significant information. Since its maximal depth is 5, in order to determine the distribution of the next sample we never have to look back more than five days, i.e., a week of trading days. The smaller the changes in the most recent S&P values, the further back we need to look in order to predict tomorrow's value.

The sum of the posterior probabilities of the top 5 models is less than 6.5%. But here the complexity of k-BCT grows significantly for values much larger than $k = 5$. In order to explore $\pi(T|x)$ further, we ran the RW sampler with $T^{(0)} = T_1^*$ for $N = 10^6$ iterations. The MCMC frequencies of the 25 most visited models shown in Figure 5 indicate that the sampler has converged after $N = 10^6$ iterations.

The MCMC output can also be used for Markov order estimation, by providing an approximation to the posterior distribution on model depth. The empirical distributions of the model depths obtained in five repetitions of the same experiment are shown in Figure 5.

VI. Prediction

Given a *training sequence* x_1^t of length t, we wish to sequentially predict the next n values of the *test sequence* x_{t+1}^{t+n}. At each step, we assume that the predictor provides a probability distribution for the next value, and we evaluate its performance by the normalized, cumulative log-loss.

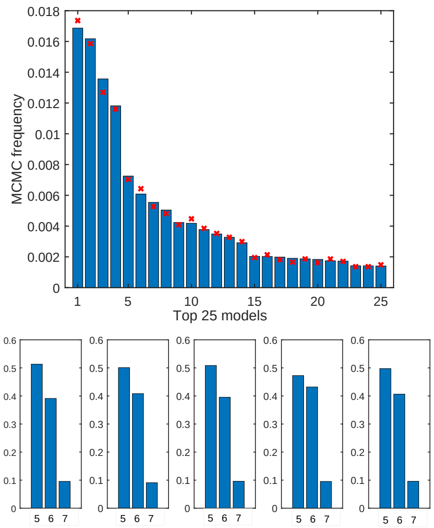


Fig. 5: Top: MCMC histogram of the empirical frequencies of the 25 most frequently visited models, after 10^6 iterations. The corresponding true posterior probabilities are marked with a red ‘x’. Bottom: Markov order estimation: The histograms show the empirical frequencies of the depths of the 1,000 most visited models after 10^6 iterations, in five MCMC runs.

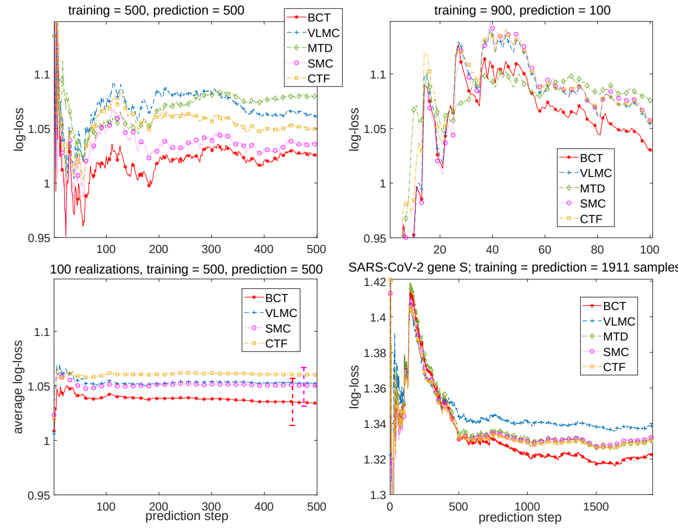


Fig. 6: Top plots: Log-loss achieved on the simulated data by all methods, as a function of the size of the test data. Bottom left: Log-loss achieved on the simulated data by BCT, VLMC, SMC and CTF on $n = 500$ test samples with $t = 500$ training samples, averaged over 100 independent repetitions of the same experiment. One-standard-deviation error bars are also plotted for BCT and SMC near the end of the test data. Bottom right: Log-loss achieved on the SARS-CoV-2 gene (S).

We compare the performance of the natural BCT predictor in terms of the posterior predictive distribution, with that of VLMC, MTD, the Sparse Markov Chain (SMC) methodology [10, 30], and the Conditional Tensor Factorization (CTF) models of [23]. Unlike BCT, all other methods first select a model and associated parameters, and then perform prediction according to those. Throughout our experiments, we take the maximal depth to be $D = 10$ for BCT, MTD, SMC and CTF. Following standard practice [1] in most cases we split the data 50-50 into a training set and a test set.

Simulated data. The experiments here are based on 1,000 samples from the 5th order chain in Section IV. The log-loss achieved by all five predictors is shown in Figure 6.

SARS-CoV-2 gene. Here we examine the spike (S) gene, in positions 21,563–25,384 of the SARS-CoV-2 genome. The data consists of a 3,822 bp-long gene sequence. The last plot in Figure 6 shows the prediction results obtained by all methods.

Remarks. Above and in numerous other experiments, the BCT predictor was found to be consistently better than the other four methods, achieving a log-loss between 1.2% and 4.8% better than that of the second best method in each case. The method that performed the closest to BCT in most cases was CTF, which usually identified the same Markov order as the other methods. The VLMC and MTD predictors were found to be consistently and significantly less effective.

VII. Discussion

This work develops a new, broad and systematic Bayesian framework for the analysis of discrete time series, based on the class of variable-memory Markov models, along with a collection of algorithmic tools for exact inference, posterior exploration, and prediction. In addition to CTW and BCT, one of our main novel contributions is the k-BCT algorithm, which identifies the k *a posteriori* most likely models.

Theorems 3.1–3.3 establish the validity of these algorithms; their proofs are by induction on D and they exploit the specific form of the priors and of the induced marginal likelihood. The algorithms are implemented in the publicly available `R` package `BCT` [13]; their complexities are of $O(nmD)$, $O(nmD)$ and $O(nmDk^m)$, respectively (importantly, linear in n for all three), but the compexity of k-BCT can be reduced to $O(nmD \times (mk) \log(mk))$ by employing a best-first search to find only the top-k combinations by using a priority queue to maintain all possible candidates and efficiently pick the next best. Two novel, variable-dimension MCMC algorithms were also introduced, that allow for broad posterior exploration.

Although Bayesian approaches to the CTW have been explored before [27, 28, 29], the present framework is based on a new prior structure, accompanied by an extensive collection of algorithms that not only provide a comprehensive picture of the posterior, but also yield themselves to the development of very effective techniques for numerous statistical tasks. This was illustrated, e.g., by the superior performance of the BCT predictor compared with other state-of-the-art methods.

Finally we briefly mention some of the many directions of possible extensions and applications, some of which we are currently investigating: worst-case bounds on the prior predictive likelihood can be established, as outlined in [11]; better estimation and weighting in the CTW and BCT with larger alphabet sizes may be obtained using techniques from [14]; improvements to the local probability estimators are possible using ideas in [24]; and effective algorithms are straightforward to develop for a variety of other applications, including Markov order estimation [6], entropy estimation [7], and changepoint detection [26].

REFERENCES

[1] R. Begleiter, R. El-yaniv, and G. Yona. On prediction using variable order Markov models. *J. Artificial Intelligence Res.*, 22:385–421, 2004.

[2] A. Berchtold. Modélisation autorégressive des chaînes de Markov: Utilisation d'une matrice différente pour chaque retard. *Revue de Statistique Appliquée*, 44(3):5–25, 1996.

[3] J.M. Bernardo and A.F.M. Smith. *Bayesian theory*. John Wiley & Sons, New York, 1994.

[4] P. Bühlmann and A.J. Wyner. Variable length Markov chains. *Ann. Statist.*, 27(2):480–513, April 1999.

[5] H. Chipman, E.I. George, R.E. McCulloch, M. Clyde, D.P. Foster, and R.A. Stine. The practical implementation of Bayesian model selection. In *Model selection*, volume 38 of *IMS Lecture Notes Monogr. Ser.*, pages 65–134. Inst. Math. Statist., Beachwood, OH, 2001. With discussion by M. Clyde, Dean P. Foster, and Robert A. Stine, and a rejoinder by the authors.

[6] I. Csiszár and P.C. Shields. The consistency of the BIC Markov order estimator. *Ann. Statist.*, 28(6):1601–1619, 2000.

[7] Y. Gao, I. Kontoyiannis, and E. Bienenstock. Estimating the entropy of binary time series: Methodology, some theory and a simulation study. *Entropy*, 10(2):71–99, June 2008.

[8] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

[9] P. Grünwald. *The minimum description length principle*. MIT Press, Cambridge, MA, 2007.

[10] V. Jääskinen, J. Xiong, J. Corander, and T. Koski. Sparse Markov chains for sequence data. *Scand. J. Stat.*, 41(3):639–655, 2014.

[11] I. Kontoyiannis, L. Mertzanis, A. Panotonoulou, I. Papageorgiou, and M. Skoularidou. Bayesian Context Trees: Modelling and exact inference for discrete time series. *ArXiv e-prints*, `2007.14900 [stat.ME]`, July 2020.

[12] L. Mertzanis, A. Panotonoulou, M. Skoularidou, and I. Kontoyiannis. Deep tree models for 'big' biological data. In *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–5, Kalamata, Greece, June 2018.

[13] I. Papageorgiou, V.M. Lungu, and I. Kontoyiannis. BCT: Bayesian Context Trees for discrete time series. *R package version 1.1*, December 2020. Availble online at: `CRAN.R-project.org/package=BCT`.

[14] F.C. Pereira and Y. Singer. An efficient extension to mixture techniques for prediction and decision trees. *Machine Learning*, 36(3):183–199, 1999.

[15] A.E. Raftery. A model for high-order Markov chains. *Journal of the Royal Statistical Society: Series B*, 47(3):528–539, 1985.

[16] A.E. Raftery and S. Tavaré. Estimation and modelling repeated patterns in high order Markov chains with the mixture transition distribution model. *Journal of the Royal Statistical Society: Series C*, 43(1):179–199, 1994.

[17] J. Rissanen. A universal data compression system. *IEEE Trans. Inform. Theory*, 29(5):656–664, September 1983.

[18] J. Rissanen. A universal prior for integers and estimation by minimum description length. *Ann. Statist.*, 11(2):416–431, June 1983.

[19] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society: Series B*, 49(3):223–239, 253–265, 1987. With discussion.

[20] J. Rissanen. *Stochastic complexity in statistical inquiry*. World Scientific, Singapore, 1989.

[21] C.P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer-Verlag, New York, second edition, 2004.

[22] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149, 1996.

[23] A. Sarkar and D.B. Dunson. Bayesian nonparametric modeling of higher order Markov chains. *J. Amer. Statist. Assoc.*, 111(516):1791–1803, 2016.

[24] T.J. Tjalkens, Y.M. Shtarkov, and F.M.J. Willems. Sequential weighting algorithms for multi-alphabet sources. In *6th Joint Swedish-Russian International Workshop on Information Theory*, pages 230–234, Mölle, Sweden, August 1993.

[25] T.J. Tjalkens, F.M.J. Willems, and Y.M. Shtarkov. Multi-alphabet universal coding using a binary decomposition context tree weighting algorithm. In *15th Symposium on Information Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 1994.

[26] J. Veness, M. White, M. Bowling, and A. György. Partition tree weighting. In *2013 Data Compression Conference*, pages 321–330, March 2013.

[27] F.M.J. Willems, A. Nowbahkt-Irani, and P.A.J. Volf. Maximum a-posteriori probability tree models. In *4th International ITG Conference on Source and Channel Coding*, Berlin, Germany, February 2002.

[28] F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens. The context tree weighting method: Basic properties. *IEEE Trans. Inform. Theory*, 41(3):653–664, May 1995.

[29] F.M.J. Willems and P.A.J. Volf. Context maximizing: Finding MDL decision trees. In *15th Symposium on Information Theory in the Benelux*, Louvain-la-Neuve, Belgium, May 1994.

[30] J. Xiong, V. Jääskinen, and J. Corander. Recursive learning for sparse Markov models. *Bayesian Analysis*, 11(1):247–263, 2016.