Bayesian Variable Order Markov Models

Christos Dimitrakakis
University of Amsterdam

Abstract

We present a simple, effective generalisation of variable order Markov models to full online Bayesian estimation. The mechanism used is close to that employed in context tree weighting. The main contribution is the addition of a prior, conditioned on context, on the Markov order. The resulting construction uses a simple recursion and can be updated efficiently. This allows the model to make predictions using more complex contexts, as more data is acquired, if necessary. In addition, our model can be alternatively seen as a mixture of tree experts. Experimental results show that the predictive model exhibits consistently good performance in a variety of domains.

We consider Bayesian estimation of variable order Markov models (see Begleiter et al., 2004, for an overview). Such models create a tree of partitions, where the disjoint sets of every partition correspond to different contexts. We can associate a sub-model or expert with each context in order to make predictions. The main contribution of this paper is a *conditional prior* on the Markov order—or equivalently the context depth. This is based on a recursive construction that estimates, for each context at a certain depth k, whether it makes better predictions than the predictions of contexts at depths smaller than k. This simple model defines a mixture of variable order Marko models and its parameters can be updated in closed form in time $\mathcal{O}(D)$ for trees of depth D with each new observation. For unbounded length contexts, the complexity of the algorithm is $\mathcal{O}\left(T^2\right)$ for an input sequence of length T. Furthermore, it exhibits robust performance in a variety of tasks. Finally, the model is easily extensible to controlled processes.

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

The following section presents our setup for the sequence prediction problem and introduces notation. Section 2 defines the proposed Bayesian variable order Markov model (henceforth BVMM) and derives a closed form sequential update for the model parameters. Section 3 gives a brief overview of other tree based models, as well as a standard Markov chain mixture, and discusses their relation to BVMMs. Experimental comparisons between various models are given in Section 4, on a number of sequential prediction problems. The paper concludes with Section 5, which also discusses extensions and applications to reinforcement learning.

1 INTRODUCTION

We consider sequences of observations $x_1, x_2, \ldots$, with $x_t \in \mathcal{X}$. We assume the existence of a suitable σ-algebra $\mathfrak{B}_{\mathcal{X}}$ such that $(\mathcal{X}, \mathfrak{B}_{\mathcal{X}})$ is measurable. In particular, most of the following development assumes a finite $\mathcal{X}$. Subsequences are denoted by $x_{k:t}$ for $k \leq t$ and the concatenations of sequences are denoted by $\cdot \circ \cdot$. For example $x_{k:t+1} = x_{k:t} \circ x_{t+1}$. In addition, let $\mathcal{X}^0 \triangleq \emptyset$, $\mathcal{X}^n \triangleq \times^n \mathcal{X}$, $\mathcal{X}_k^* \triangleq \bigcup_{n=k}^{\infty} \mathcal{X}^n$ and $\mathcal{X}^* \triangleq \mathcal{X}_0^*$. We use bold symbols for arbitrary-length sequences $\mathbf{x}$ and we denote the length of any $\mathbf{x} \in \mathcal{X}^*$ by $\ell(\mathbf{x})$.

Suffixes. We call $\mathbf{x}$ a suffix of $\mathbf{x}'$, and write $\mathbf{x} \prec \mathbf{x}'$ iff $\ell(\mathbf{x}) \leq \ell(\mathbf{x}')$ and $x_{\ell(\mathbf{x})+1-i} = x'_{\ell(\mathbf{x}')+1-i}$ for all $i \in \{1, \ldots, \ell(\mathbf{x})\}$. Finally, we write $\mathbf{x} \cap \mathbf{x}'$ to denote the largest common suffix of $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^*$. If $\mathbf{x}$ and $\mathbf{x}'$ have no common suffix then $\mathbf{x} \cap \mathbf{x}' = \mathbf{0}$ and $\mathbf{0} \circ \mathbf{x} = \mathbf{x} \circ \mathbf{0} = \mathbf{x}$ for any $\mathbf{x} \in \mathcal{X}^*$.

Definition 1. *A suffix set S on $\mathcal{X}$ is a set of sequences $\mathbf{c} \in \mathcal{X}^*$. S is proper iff $\mathbf{c} \cap \mathbf{c}' = \mathbf{0}$ for all $\mathbf{c}, \mathbf{c}' \in S$. We call S complete iff, $\forall \mathbf{x} \in \mathcal{X}^*$, $\exists \mathbf{c} \in S$ such that $\mathbf{c} \prec \mathbf{x}$.*

We consider a complete, but *not* proper, suffix set S, which may be infinite, constructed via a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, of depth D, with nodes $\mathcal{V}$ and edges $\mathcal{E}$. The set of nodes $\mathcal{V} = \left\{v_i : i = 0, 1, \ldots, |\mathcal{X}|^D\right\}$ is such that the node $v_i \in \mathcal{V}$ corresponds to a unique sequence $\mathbf{c}_i \in \mathcal{X}^*$. Specifically, the root node v_0 corresponds to $\mathbf{c}_0 = \mathbf{0}$, the empty sequence. All inner nodes have $|\mathcal{X}|$

children, such that if v_j is the k-th child of node v_i, then $\mathbf{c}_j = k \circ v_i$. Thus, there are $t+1$ suffixes in S for each sequence $x_{1:t} \in \mathcal{X}^t$ and the tree can be viewed as a tree of partitions of $\mathcal{X}_D^*$. Indeed, the suffixes partition $\mathcal{X}_D^*$ into sets $X_i \triangleq \{\mathbf{c} \in \mathcal{X}^* : \mathbf{c}_i \prec \mathbf{c}\}$, such that $\mathcal{P}_k \triangleq \{X_i : \ell(\mathbf{c}_i) = k, \mathbf{c}_i \in S\}$ is the partition induced by the set of nodes at depth k of the tree.

Context models. The general idea of such models is to associate an expert μ_i with a context $\mathbf{c}_i$, such that, for a given observation history, only certain experts will have matching contexts. For the problem of sequence prediction, the context $\mathbf{c}_i$ is the suffix at the node v_i of the tree and the expert μ_i defines a probability measure $\mathbb{P}(\cdot \mid \mu_i)$ on $(\mathcal{X}, \mathfrak{B}_\mathcal{X})$ that predicts the next observations.

Given any sequence $x_{1:t}$, only a subset of experts $\mathcal{M}(x_{1:t}) \triangleq (\mu_i : \mathbf{c}_i \prec x_{1:t})$ will have contexts that are suffixes of $x_{1:t}$. For any *active* expert $\mu_k \in \mathcal{M}(x_{1:t})$, we shall use:

$$p_k^t(x) \triangleq \mathbb{P}(x_{t+1} = x \mid \mu_k, x_{1:t}) \tag{1}$$

to denote the (marginal) posterior probability of possible next observations x_{t+1}, according to expert μ_k, given the history of previous t observations.

Example 1. *When $\mathcal{X} = \{1, \dots, K\}$, we can use a Dirichlet distribution $\mathcal{D}ir(\alpha_k^t)$ over multinomial parameters for each μ_k, where $\alpha_k^t \triangleq (\alpha_{k,j}^t)_{j=1}^K$ is the vector of Dirichlet parameters for expert k at time t. The corresponding marginal probability distribution is:*

$$p_k^t(x_{t+1} = x) = \frac{\alpha_{k,x}^t}{\sum_{j=1}^K \alpha_{k,j}^t}, \tag{2}$$

for all $k \in \{1, \dots, K\}$. Given a sequence $x_{1:T}$, the parameters of the expert at each context are

$$\alpha_{i,k}^T = \alpha_{i,k}^0 + \sum_{t=1}^T \mathbb{I}\{\mathbf{c}_i \prec x_{1:t} \wedge x_{t+1} = k\}, \tag{3}$$

where $\left\{\alpha_{i,k}^0\right\}$ is a set of prior parameters, typically set to values in $[0, 1]$.

For any conditional probability distribution $\mathbb{P}(\mu|x_{1:t})$ over the set of active experts $\mathcal{M}(x_{1:t})$, we obtain the marginal probability of the next observations:

$$\mathbb{P}(x_{t+1}|x_{1:t}) = \sum_{\mu_k \in \mathcal{M}(x_{1:t})} p_k^t(x_{t+1})\, \mathbb{P}(\mu_k|x_{1:t}). \tag{4}$$

This natural idea is used in the context tree weighting algorithm (Willems et al., 1995), which employs Dirichlet models for μ. It however only considers a fixed $\mathbb{P}(\mu|x_{1:t})$. We shall introduce a simple construction that allows us to update both $\mathbb{P}(\mu|x_{1:t})$ and the experts μ efficiently.

2 BAYESIAN VARIABLE ORDER MARKOV MODELS

We first define variable order Markov models as a set of experts on a suffix tree. Subsequently, we define a mixture of such models, associated with a set of weights and we give a procedure for updating the weights given new observations.

Definition 2. *A Variable order Markov Model (VMM) over $\mathcal{X}^*$ is composed of:*

1. *A complete suffix set $S = \{\mathbf{c}_i : i = 1, \dots, N\}$.*
2. *A set of experts $\mathcal{M} = \{\mu_i : i = 1, \dots, N\}$, indexing a set of probability distributions: $\{\mathbb{P}(x_{t+1} \mid \mu_i) : \mu_i \in \mathcal{M}\}$. For the discrete case in particular, each expert $\mu_i \in \mathcal{M}$ defines a multinomial distribution with parameters $\tau_i \in [0, 1]^{|\mathcal{X}|}$ s.t. $\|\tau_i\|_1 = 1$ and $\mathbb{P}(x_{t+1} = j \mid \mu_i) = \tau_{i,j}$.*

There is a one-to-one correspondence between each context $\mathbf{c}_i$ and each expert μ_i. For any history $x_{1:t} \in \mathcal{X}^t$, the we use the surjection $I : \mathcal{X}^* \to \{1, \dots, N\}$ to denote the index $I(t) \triangleq \max\{i : \mathbf{c}_i \prec x_{1:t}\}$ of the only active expert (i.e. the active expert is the one with the largest matching suffix). The distribution of next observations is:

$$\mathbb{P}(x_{t+1} \mid x_{1:t}) \triangleq \mathbb{P}(x_{t+1} \mid \mu_{I(t)}) \tag{5}$$

To obtain a distribution of VMMs that can be updated in closed form, we define the following structure.

Definition 3 (BVMM)**.** *A Bayesian variable order Markov model over $\mathcal{X}^*$ is composed of:*

1. *A suffix tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$, of depth D with a set of nodes $\mathcal{V} = \{v_i : i = 0, 1, \dots, N\}$, with $N = \mathcal{X}^D$, and a set of edges $\mathcal{E}$.*
2. *A complete suffix set $S = \{\mathbf{c}_i : i = 0, 1, \dots, N\}$.*
3. *A set of models $\mathcal{M} = \{\mu_i : i = 0, 1, \dots, N\}$. For discrete $\mathcal{X}$, these are $\mathcal{D}ir(\alpha_i)$ with $\alpha_i \in \mathbb{R}_+^{|\mathcal{X}|}$.*
4. *A set of weights $\mathcal{W} = \{w_i \in [0, 1] : i = 0, 1, \dots, N\}$.*

There is a one-to-one correspondence between each node v_i, a suffix $\mathbf{c}_i$, an expert μ_i and a weight w_i. For any observations $x_{1:t} \in \mathcal{X}^t$, the set of active experts is $\mathcal{M}(x_{1:t}) \triangleq (\mu_{I(0,t)}, \dots, \mu_{I(t,t)})$, where $I : \mathcal{X}^* \times \mathbb{N} \to \{0, \dots, N\}$ is a surjection such that $\mathbf{c}_{I(0,t)} = \mathbf{0}$ and $\mathbf{c}_{I(k+1,t)} = x_{t-k} \circ \mathbf{c}_{I(k,t)}$.

In order to generate a new observation x_{t+1}, given $x_{1:t}$, we define the auxiliary indicator variable $s_t =$

$(s_{t,0} \dots, s_{t,t})$, *such that:* $x_{t+1} \sim \mathbb{P}(x_{t+1} \mid x_{1:t}, \mu_{I(k,t)})$ *if* $s_{t,k} = 1$. *The following properties hold for* s*:*

$$\begin{aligned}\mathbb{P}(s_{t,t} = 1 \mid x_{1:t}) &= w_{I(t,t)},\\ \mathbb{P}(s_{t,k} = 1 \mid s_{t,k+1} = 0, x_{1:t}) &= w_{I(k,t)},\\ \mathbb{P}(s_{t,k} = 1 \mid \exists j > k : s_j = 1) &= 0,\\ w_0 &= 1.\end{aligned}$$

The probability of the data being generated by the k*-th model, according to the ordering implied by* $x_{1:t}$*, is:*

$$\mathbb{P}(s_{t,k} = 1 \mid x_{1:t}) = w_{I(k,t)} \prod_{j=k+1}^{t} (1 - w_{I(j,t)}), \quad (6)$$

where $\prod_{j=t+1}^{t}(1-w_{I(j,t)}) = 1$ *for notational simplicity.*

Intuitively, s can be seen as a stopping variable. Given an observation $x_{1:t}$, we form the chain of active experts $\mathcal{M}(x_{1:t})$. Starting from the last model, $\mu_{I(\min(t,D),t)}$, and for every model $\mu_{I(k,t)} \in \mathcal{M}(x_{1:t})$, we stop with probability $w_{I(k,t)}$ and generate $x_{t+1} \sim p_{I(k,t)}(x_{t+1})$. Otherwise, we move to $\mu_{I(k-1,t)}$. If $k = 0$, the next observation is generated from the root model μ_0.

Remark 1. *By construction,* $\sum_k \mathbb{P}(s_{t,k} = 1) = 1$*, for any set of weights* $\left\{w_{I(k,T)} \geq 0 : k > 0\right\}$.

Proof. The proof proceeds by induction. Without loss of generality, we only consider $t \leq D$, since we can set $w_t = 0$ for $t > D$. When $t = 0$, $\sum_{k=0}^{t} \mathbb{P}(s_{t,k}) = w_0 = 1$ by definition, where we used $\mathbb{P}(s_{t,k}) \equiv \mathbb{P}(s_{t,k} = 1)$ for compactness. For other t:

$$\begin{aligned}\sum_{k=0}^{t+1} \mathbb{P}(s_{t,k}|x_{1:t}) &= \sum_{k=0}^{t+1} w_{I(k,t)} \prod_{n=k+1}^{t+1}(1 - w_{I(n,T)})\\ &= (1 - w_{I(t+1,T)}) \sum_{k=0}^{t} \mathbb{P}(s_{t,k}) + w_{I(t+1,T)},\end{aligned}$$

which obviously equals 1 if $\sum_{k=0}^{t} \mathbb{P}(s_{t,k}) = 1$. □

Remark 2. *A BVMM of depth* D *defines a distribution over VMMs.*

Proof. $\mathcal{W}$ and S define a distribution over complete context sets of maximum suffix length D. To see this, let us construct a random context set $\hat{S}$. Since $w_0 = 1$, $\hat{S}$ is always complete and the probability of each context being in $\hat{S}$ can be derived via (6) as:

$$\mathbb{P}(\mathbf{c}_i \in \hat{S}) \propto w_i \prod_{j:\mathbf{c}_i \prec \mathbf{c}_j} (1 - w_j). \quad (7)$$

Finally, for all $\mathbf{c}_i \in \hat{S}$, generate multinomial parameters $\tau_i \sim \mathcal{D}ir(\alpha_i)$. These two sets are sufficient for specifying a VMM. □

2.1 Update procedure

We now consider a recursive procedure for updating the parameters of a BVMM. For this reason, we use a superscript t to refer to the value of parameters at time t. Furthermore, we need a way to refer to the observations being generated an expert in a particular subset. This leads us to the following construction, which is central in the remaining development: Let $B_k \triangleq \mathbb{I}\{\exists i \leq k : s_{t,k} = 1\}$ denote the event that the data is generated by one of the experts with context size at most k, i.e. that $\mu \in \left\{\mu_{I(0,t)}, \dots, \mu_{I(k,t)}\right\}$. This allows us to interpret the weights as the posterior of the k-th model (where k indexes the active context experts $\mathcal{M}(x_{1:t})$), given the observation history $x_{1:t}$ and the fact that the model order is not larger than k:

$$w^t_{I(k,t)} = \mathbb{P}(\mu_{I(k,t)} \mid x_{1:t}, B_k). \quad (8)$$

Using (6), we can write the marginal predictive distribution (4) of our model at time t, in terms of the weights:

$$\mathbb{P}(x_{t+1}{=}x|x_{1:t}) = \sum_{k=0}^{t} p^t_{I(k,t)}(x) w^t_{I(k,t)} \prod_{n=k+1}^{t}(1 - w^t_{I(n,t)}).$$

The following theorem gives a procedure for updating $\mathcal{W}$ in closed form.

Theorem 1. *The weight parameters* $\mathcal{W}$ *of any BVMM can be recursively updated in closed form according to:*

$$\begin{aligned}w^{t+1}_{I(k,t)} &\triangleq \mathbb{P}(\mu_{I(k,t)}|x_{1:t+1}, B_k) \quad (9)\\ &= \frac{p^t_{I(k,t)}(x_{t+1}) w^t_{I(k,t)}}{p^t_{I(k,t)}(x_{t+1}) w^t_{I(k,t)} + \mathbb{P}(x_{t+1}|x_{1:t}, B_{k-1})(1 - w^t_{I(k,t)})}\end{aligned}$$

Proof. First of all, note that B_t is trivially true at time t. For B_k with $k < t$, it is easy to see that the following recursions hold:

$$\mathbb{P}(B_{k-1}|x_{1:t}) = \mathbb{P}(B_k|x_{1:t})(1 - w^t_{I(k,t)}) \quad (10a)$$

$$\begin{aligned}\mathbb{P}(x_{t+1}|x_{1:t}, B_k) &= p^t_{I(k,t)}(x_{t+1}) w^t_{I(k,t)} \quad (10b)\\ &\quad + \mathbb{P}(x_{t+1}|x_{1:t}, B_{k-1})(1 - w^t_{I(k,t)}),\end{aligned}$$

where we used (8) and that $\mathbb{P}(x_{t+1}|\mu_{I(k,t)}, x_{1:t}, B_k) = \mathbb{P}(x_{t+1}|\mu_{I(k,t)}, x_{1:t}) = p^t_{I(k,t)}(x_{t+1})$, as given the k-th expert, the next observations do not depend on previous experts. Using (10) and Bayes' theorem, we have:

$$\begin{aligned}w^{t+1}_{I(k,t)} &\triangleq \mathbb{P}(\mu_{I(k,t)}|x_{1:t+1}, B_k)\\ &= \frac{\mathbb{P}(x_{t+1}|\mu_{I(k,t)}, x_{1:t}, B_k)\, \mathbb{P}(\mu_{I(k,t)}|x_{1:t}, B_k)}{\mathbb{P}(x_{t+1}|x_{1:t}, B_k)}\\ &= \frac{p^t_{I(k,t)}(x_{t+1}) w^t_{I(k,t)}}{p^t_{I(k,t)}(x_{t+1}) w^t_{I(k,t)} + \mathbb{P}(x_{t+1}|x_{1:t}, B_{k-1})(1 - w^t_{I(k,t)})}\end{aligned}$$

□

Updating $p_k^t(x)$ is easy if μ_k is Dirichlet with parameters $\alpha_k \triangleq \{\alpha_{k,i} : i = 1, \ldots, |\mathcal{X}|\}$, as described in Example 1. Finally, note that due to Remark 1, the weights always define a proper posterior probability distribution over the experts and additionally, a distribution over VMMs due to Remark 2.

Remark 3. *At each step t, the update complexity is $\mathcal{O}(\min\{D, t\})$, so the complexity for a sequence of length T is $\mathcal{O}\left(\min\{T^2, TD\}\right)$.*

3 RELATED MODELS

This section gives a brief overview of models related to BVMMs. Of these, the closest models are the following: (a) The Bayesian Markov chain mixture, which uses a standard prior over Markov order. (b) Dirichlet process models, which employ sampling. (c) The context tree weighting algorithm (Willems et al., 1995), which has fixed weights. We shall now discuss the related models in more detail.

Bayesian Markov chain mixture. A simpler type of Bayesian model for sequence prediction is a mixture over Markov chains (henceforth BMCM). Let a set of experts $\{\mu_k : k = 0, \ldots, D\}$, and a multinomial prior distribution with parameters $\phi^0 = (\phi_k^0)_{k=0}^D$. Each expert μ_k is a distribution over the parameters of Markov chains of order k for a discrete observation set $\mathcal{X}$. In particular, we consider a product-of-Dirichlets conjugate prior, with parameters $\alpha_k^t \triangleq \left\{\alpha_{k,z}^t : z \in \mathcal{X}^k\right\}$, updated according to $\alpha_{k,z}^T = \alpha_{k,z}^0 + \sum_{t=1}^T \mathbb{I}\{z \prec x_{1:t} \wedge x_{t+1} = k\}$ and with (marginal) predictive distribution $p_k^t(x_{t+1} = i | z \prec x_{1:t}) = \frac{\alpha_{i,z}^t}{\sum_j \alpha_{j,z}^t}$. The mixture is updated via Bayes' rule:

$$\phi_k^{t+1} \triangleq \frac{p_k^t(x_{t+1})\phi_k^t}{\sum_{j=0}^D p_j^t(x_{t+1})\phi_j^t} \tag{11}$$

This model is simpler than a BVMM. In fact, it can be seen as a BVMM with the weights of all contexts at a certain depth k being equal. Thus, a potential problem is that a large amount of data is required for the μ_{k+1} to start making globally better predictions than μ_k. Intuitively, we could do better by switching to larger order models for some contexts only. This can be achieved if we allow our belief over model order to depend on the history, something taken care of automatically in BVMMs.

Other variable length Markov chains models. One closely related model is the context tree weighting (henceforth CTW) algorithm (Willems et al., 1995). CTW employs smooth maximum likelihood estimation at each context and so is equivalent to BVMMs with respect to the adaptation of the experts μ, when those are Dirichlet-multinomial. While CTW uses a closed-form update, the weights used in CTW are fixed. The prediction by partial matching (PPM) algorithm (Cleary and Witten, 1984), includes a closed-form weight update, which is however ad-hoc (Begleiter et al., 2004, p.392). Other variants are examined in (Begleiter et al., 2004) which in addition supplies an experimental comparison between methods. A final related model is Variable length Markov chains (Bühlmann and Wyner, 1999) (henceforth VMC), which however utilises growing and subsequent pruning of the context tree. It is thus a batch (offline) algorithm.

Tree experts. A tree expert is a collection of a finite number of experts, with each expert being a predictive tree, whose nodes are of two types: leaf nodes (which have no children) and inner nodes (which all have n children). Decisions at inner nodes concern only which child to proceed to, while the decisions at leaf nodes concern a prediction of the next observation. A VMM can be seen as a particular type of tree expert and a BVMM as a mixture of tree experts. A low regret prediction algorithm for such models is given in (Cesa-Bianchi and Lugosi, 2006, ch. 5.3).

Dirichlet process models. An important class of priors over distriutions are **Polya trees** (Ferguson, 1974). Just as in BVMMs, a distribution is defined over a partition tree. However, there is only one set of parameters for each node, which relates to the child node which will be next visited. This allows closed form updates, but lacks the additional expressiveness possible with BVMMs. Dirichlet processes are also used in the **infinite hidden Markov model** (IHMM, Beal et al., 2001) and the **infinite Markov model** (IMM, Mochihashi and Sumita, 2008). In particular, the IMM uses a similar structure, with the difference that a Beta prior on the stopping variable s is used. Inference in both of these models requires sampling instead. Thus, as long as one is only interested in prediction, rather than state estimation, BVMMs amount to a significant improvement over I(H)MMs in terms of computation. However, another approach close to the BVMM is the **stochastic memoizer** (SM), proposed by Wood et al. (2009), which, although it also employs sampling, it is much more efficient in terms of computation.

Other models. In a different setting, learning mixtures of trees has been explored using EM in (Meila and Jordan, 2001), with a similar construction to the BVMM and the IMM, while (Friedman and Koller, 2003) extended the work to the more general problem

of learning network structure, for which they employed Markov chain Monte Carlo approaches. Finally, there are some parallels with the autoregressive literature, where the prediction problem is the same, but $\mathcal{X} \subset \mathbb{R}^n$, such as the work of Mena and Walker (2005), which utilised a Dirichlet process mixture latent variable.

4 EXPERIMENTS

We evaluated the method on a number of sequence prediction domains.[1] The objective was to compare its performance against commonly used approaches in the target domain. Our evaluation criterion was the total error made by each algorithm in *online* prediction.

For comparative purposes, we employed two domains. The first was stochastically generated hidden Markov models, for which there exist approximate inference methods in the target model class. The second domain was fixed sequences of text and gene data.

4.1 HIDDEN MARKOV MODEL

In this domain, data is generated from a hidden Markov model (hence forth HMM) ν^*, with a discrete set of states $s \in \mathcal{S}$ and observations $x \in \mathcal{X}$. Experiments were performed for $|\mathcal{S}| \in \{2, 4, 8\}$ and $|\mathcal{X}| \in \{2, 4, 8\}$. There were 100 runs performed for each choice of $\mathcal{X}, \mathcal{S}$. For each run, a HMM was stochastically generated in order to ensure that: (a) There would be some stationarity in every state, so as to make sequences more predictable. (b) States would be sufficiently differentiable from each other.

For the n-th run of each experiment, a HMM ν_n^* with transition and observation matrices P and Q, such that $P_{ij} \triangleq \nu_n^*(s_{t+1} = j | s_t = i)$ and $Q_{ij} \triangleq \nu_n^*(x_t = j | s_t = i)$, was used to generate the observation sequence $x_{1:t}$ by setting $s_0 = 1$ and sampling $s_{t+1} \sim \nu_n^*(s_{t+1}|s_t)$, and $x_t \sim \nu_n^*(x_t|s_t)$. In order to ensure that the HMMs had the desired structure, the matrices were generated as follows. First, a stationarity parameter β was generated uniformly in the interval $[\frac{1}{2}, 1)$. The entries of P were set to $P_{ij} = \hat{p}_{ij} / \sum_k \hat{p}_{ik}$, with $\hat{p}_{i,j} = \mathbb{I}\{i = j\}\beta + \exp(z_{i,j})$, with $z_{i,j} \sim \mathcal{U}ni(0, 1)$. The entries of Q were set to $Q_{ij} = \hat{q}_{ij} / \sum_k \hat{q}_{ik}$, with $\hat{q}_{i,j} = \mathbb{I}\{i = j\} + \zeta_{i,j}$, with $\zeta_{i,j} \sim \mathcal{U}ni(0, 1/10)$.

For each run, we calculated the cumulative loss ℓ of each model ν assigning probabilities $\nu^t(x_{t+1})$ to outcomes given the history $x_{1:t}$, generated by ν_n^*.

$$\ell_T(\nu) \triangleq \sum_{t=1}^{T} \mathbb{I}\left\{ x_{t+1} \neq \arg\max_{i \in \mathcal{X}} \nu^t(x_{t+1} = i) \right\}. \quad (12)$$

In this case, we interested in the *expected* loss $\mathbb{E}\,\ell_t(\nu)$. We compared the following models: (a) The **HMM oracle** model, which is a HMM with the same parameters as the true HMM ν, and uses the observations $x_{1:t}$ to form a belief over s_t and predict the next observations. (b) The **HMM particle filter** model, which is a grid filter (see Doucet et al., 2001) with n_p particles on the parameter space. We additionally employed a variant which resampled grid points when particle weights dropped below the threshold $(n_p)^{-1/2}$. (c) The **HMM EM** model, which performs expectation maximisation on all the data to estimate a model, which it then uses to predict (it thus should have a performance very close to that of the oracle). In addition, we used an *incremental EM* variant; this simply performs one iteration of expectation maximisation for each new observation, starting from the result of the previous iteration.(d) The **BMCM**. (e) The **BVMM**. We utilised a prior $\alpha_i^0 = \frac{1}{2}$ for the Dirichlet models, $w_k^0 = 2^{-k}$ for the BVMM, and $\phi_k^0 \propto 2^{-k}$ for the BMCM. For the HMM algorithms, the initial parameters were sampled from the distribution used to generate ν^*. For the particle filters, $n_p = 128$ particles were used.

Figure 1 shows the run-average regret of each model ν with respect to the HMM oracle ν^*, $\rho_T(\nu) \triangleq \frac{1}{100} \sum_{n=1}^{100} (\ell_T(\nu_n) - \ell_T(\nu_n^*))$. Naturally, the EM approach almost always has the best performance, as it makes predictions after it has been trained on the complete sequence. The naive incremental EM method sometimes matches full EM, but both have problems with local optima. The particle filter methods performed well for a small number of states and observations only. Both the BMCM and the BVMM performance track the performance of the other methods well, even though they are not in the correct model class. In particular, BMCM is consistently close to or better than the best HMM estimation method. This is perhaps due to the stationarity and easy identifiability of the states for this particular set of hidden Markov models. The BVMM exhibits slower convergence, but it eventually catches up (Figure 1b) and sometimes matches BMCM (Figures 1c, 1d).

4.2 FIXED SEQUENCES

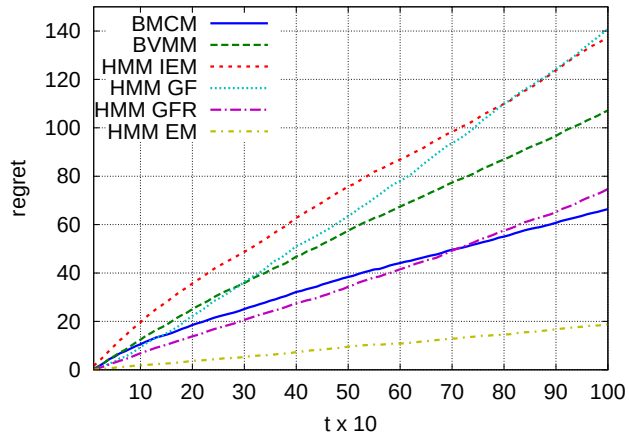
We performed experiments on a number of fixed sequences $x_{1:T}$. In this case, for each model ν making predictions ν^t, we measure the average log loss:

$$\mathcal{L}(\nu \mid x_{1:T}) \triangleq \frac{1}{T} \sum_{t=0}^{T} \log_2 \nu^t(x_{t+1}). \quad (13)$$

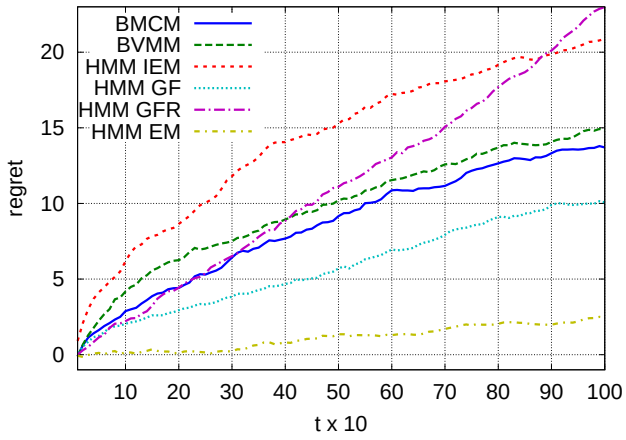
We performed experiments on the *large* and *calgary* corpuses[2], where we compared the following three

[1] Code available at `http://code.google.com/p/beliefbox/`

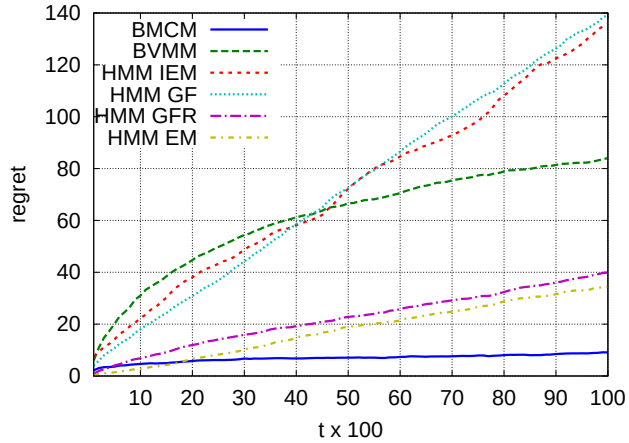
[2] `http://corpus.canterbury.ac.nz/`



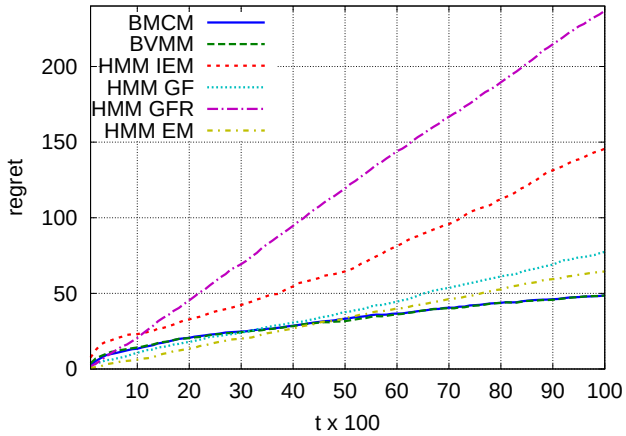
(a) $D = 8$, $|\mathcal{S}| = 8$, $|\mathcal{X}| = 8$



(b) $D = 8$, $|\mathcal{S}| = 4$, $|\mathcal{X}| = 2$



(c) $D = 8$, $|\mathcal{S}| = 2$, $|\mathcal{X}| = 2$



(d) $D = 8$, $|\mathcal{S}| = 4$, $|\mathcal{X}| = 2$

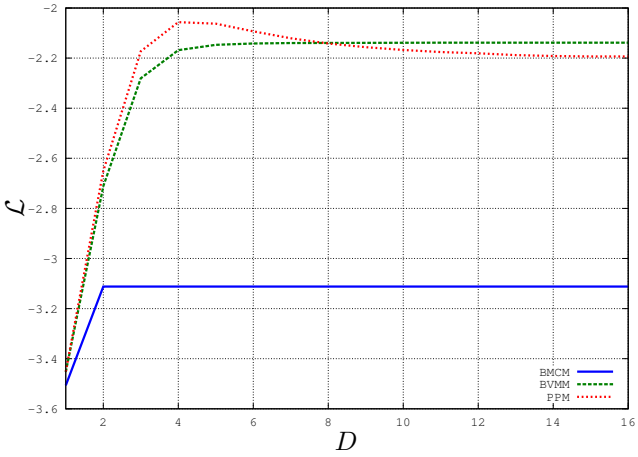
Figure 1: The average regret ρ_T of different models with respect to the HMM oracle. **BMCM** is a Markov chain mixture, while **BVMM** is the proposed model, both with depth D. **HMM EM** is an HMM trained with EM on all data, while **HMM IEM** uses an incremental EM method, **HMM GF** is a grid filter and **HMM GFR** is a grid filter with replacement.

models: (a) The **PPM** algorithm (Cleary and Witten, 1984), and in particular, the PPM-C variant which is known to have particularly good performance in such tasks (see Begleiter et al., 2004), (b) the **BMCM**, (c) the **BVMM**, with experts using the generalisation of the Dirichlet for large alphabets, similar to that used in PPM. The BVMM used a prior $w_k^0 = 2^{-k}$ and for the BMCM a matching prior $\phi_k^0 \propto 2^{-k}$. To limit memory use for very large sequences, the context tree was grown dynamically, adding leaves only to contexts with at least two observations.

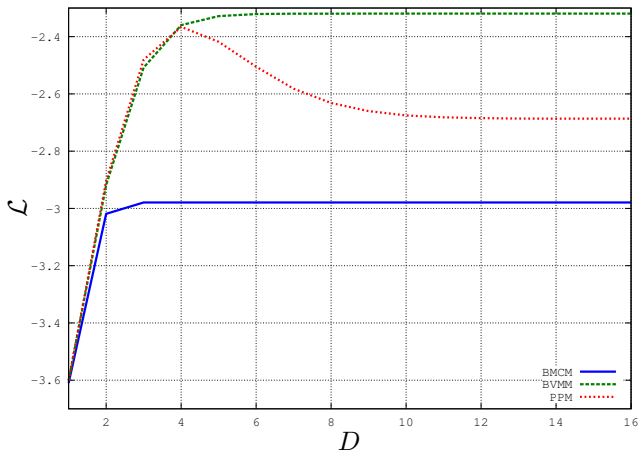
Figure 2 shows the log loss of these algorithms for increasing maximum depth D. The BMCM is usually underfitting, making no use of depths higher than 3. The BVMM approach does not appear to overfit for those choices when depth increases. The PPM approach may have a slight overall advantage, as long as it is tuned appropriately. The unique behaviour of the *E. Coli* dataset (Fig. 2c), is particularly interesting. PPM shows a sharp reduction in performance, while BMCM has a pronounced increase in performance around depth 20, which the BVMM fails to replicate. This may be due to the fact that weights of contexts at each depth are independent of each other, but this is something that will require future investigation.

A summary of the results on the *calgary* corpus is shown in Table 1, which in addition shows results for CTW and SM[3]. Both the CTW and SM algorithm enjoy an advantage of $0.15 - 0.25$ bits/symbol on average

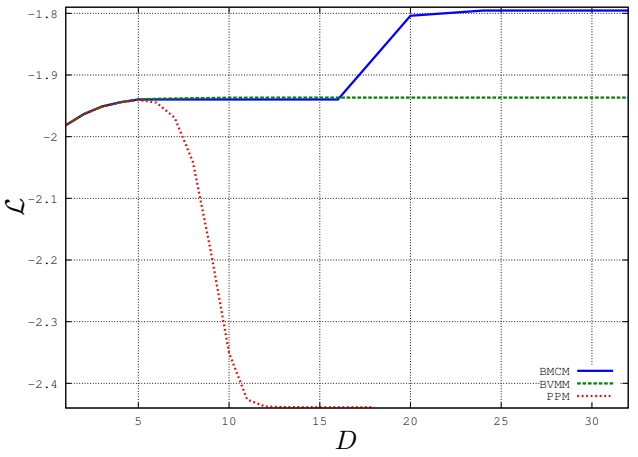
[3]Results obtained from (Gasthaus et al., 2010)



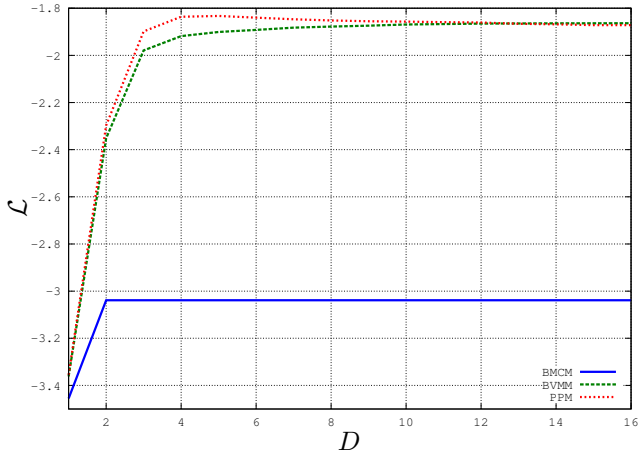
(a) The *bib* dataset



(b) The *book1* dataset



(c) The *E. Coli* dataset



(d) The *progp* dataset

Figure 2: Results for four datasets from the *calgary* and *large* corpuses, chosen to show representative behaviours for all methods **BMCM**, **BVMM** and **PPM**. Each graph shows the mean log loss $\mathcal{L}$ for varying D.

	BMCM	BVMM	PPM	SM^3	CTW^3
mean	3.41	2.46	2.54	2.12	2.24
w. m.	2.93	2.14	2.39	1.89	1.99

Table 1: Calgary corpus result summary. The mean and weighted mean bits/symbol of each method on the 14 files tested in (Gasthaus et al., 2010) are shown.

compared to BVMM, and in fact, they performed better in each and every one of these datasets. Thus, the BVMM, although close to the state of the art, cannot be recommended for compression in this form.

Additional experiments (not shown) indicated that the BMCM and BVMM are robust to the choice of prior weights, with a slight drop-off in performance at higher depths if large weights were chosen. This implies that adjusting the weights of the trees in a manner that depends on the data is actually effective.

5 CONCLUSION

We have introduced a Bayesian version of variable order Markov models, that can be efficiently updated in *closed form*. This is possible due to a recursive construction. Furthermore, we outlined its relations to other models such as the context tree weighting algorithm, and its equivalence to a mixture of tree experts.

In future theoretical work, we shall further investigate links between BVMMs and other similar models. It would also be particularly interesting to derive performance bounds on the predictions of BVMMs.

The experimental results indicate that BVMMs have a consistently good predictive performance, while at the same time being relatively insensitive to the choice of

prior. It seems reasonable to conclude that BVMMs should be a robust choice when little is known about the data and when the problem is mainly prediction, rather than tracking of a hidden state, or when low computational complexity is an issue.

Extensions. In the above development, we only considered finite $\mathcal{X}$. A naive extension to countable $\mathcal{X}$ is straightforward, but robustness may require additional machinery. The model can be further extended so that the weights of each context are not independent, in the manner of SM. This may result in improved performance, especially as evidenced by the results in the *E. Coli* dataset. Finally, the model can be extended to controlled processes. In particular, it may be an effective Bayesian model for near-optimal decision making in unknown partially observable Markov decision processes (i.e. Ross et al., 2008). Since BVMMs are able to provide good predictions, as well as easily computable closed-form posteriors, they are an excellent candidate for planning under uncertainty in such domains (Dimitrakakis, 2009, 2010). It is of the author's opinion that this is a better application for such models than sequence prediction.

Acknowledgements

Part of this work was performed while at the university of Leoben. Many thanks to P. Auer and R. Ortner who have helped with the preparation of this paper, as well as to P. Grünwald, F. Oliehoek and N. Vlassis for comments and discussions and the anonymous reviewers for their extremely detailed suggestions. This work was partially supported by the ICIS project, under the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

Matthew J. Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. The infinite hidden Markov model. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *NIPS*, pages 577–584. MIT Press, 2001.

Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, pages 385–421, 2004.

Peter Bühlmann and Abraham J. Wyner. Variable length Markov chains. *The Annals of Statistics*, 27 (2):480–513, 1999.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning and Games*. Cambridge University press, Cambridge, UK, 2006.

J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):394–402, 1984.

Christos Dimitrakakis. Variable order Markov decision processes: Exact Bayesian inference with an application to POMDPs. Submitted, 2010.

Christos Dimitrakakis. Bayesian variable order Markov models: Towards bayesian predictive state representations. Technical Report IAS-UVA-09-04, University of Amsterdam, June 2009.

Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

Thomas S. Ferguson. Prior distributions on spaces of probability measures. *The Annals of Statistics*, 2(4): 615–629, 1974. ISSN 00905364.

N. Friedman and D. Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1):95–125, 2003.

J. Gasthaus, F. Wood, and Y. W. Teh. Lossless compression based on the sequence memoizer. In *Data Compression Conference*, 2010.

M. Meila and M.I. Jordan. Learning with mixtures of trees. *The Journal of Machine Learning Research*, 1:1–48, 2001.

Ramsés H. Mena and Stephen G. Walker. Stationary autoregressive models via a bayesian nonparametric approach. *Journal of Time Series Analysis*, 26(6): 789–805, 2005.

D. Mochihashi and E. Sumita. The infinite Markov model. In *Advances in Neural Information Processing Systems*, pages 1017–1024. MIT Press, 2008.

Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive POMDPs. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, Cambridge, MA, 2008. MIT Press.

F.M.J. Willems, Y.M. Shtarkov, and T.J. Tjalkens. The context tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41 (3):653–664, 1995.

F. Wood, C. Archambeau, J. Gasthaus, L. James, and Y.W. Teh. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM New York, NY, USA, 2009.