Inhomogeneous Parsimonious Markov Models

Ralf Eggeling[1*], André Gohr[1*], Pierre-Yves Bourguignon[2], Edgar Wingender[3], and Ivo Grosse[1,4]

[1] Institute of Computer Science, Martin Luther University
06099 Halle, Germany
[2] Max Planck Institute for Mathematics in the Sciences
04103 Leipzig, Germany
[3] Institute of Bioinformatics, University Medical Center Göttingen
37077 Göttingen, Germany
[4] German Center of Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig
04103 Leipzig, Germany

Abstract. We introduce inhomogeneous parsimonious Markov models for modeling statistical patterns in discrete sequences. These models are based on parsimonious context trees, which are a generalization of context trees, and thus generalize variable order Markov models. We follow a Bayesian approach, consisting of structure and parameter learning. Structure learning is a challenging problem due to an overexponential number of possible tree structures, so we describe an exact and efficient dynamic programming algorithm for finding the optimal tree structures. We apply model and learning algorithm to the problem of modeling binding sites of the human transcription factor C/EBP, and find an increased prediction performance compared to fixed order and variable order Markov models. We investigate the reason for this improvement and find several instances of context-specific dependences that can be captured by parsimonious context trees but not by traditional context trees.

1 Introduction

Discrete sequential data as diverse as bit strings in computer science, DNA and polypeptide molecules in bioinformatics, or alphabetic strings in linguistics are omnipresent in todays science and technology. Despite highly diverse applications, the characterization of ensembles of sequences based on a finite sample is a common and fundamental statistical challenge raised in these different fields. Examples are data compression [1, 2], the prediction of functional sites in biological macromolecules [3–6], or the study of the structure of languages [7, 8].

While reducing a sequence to a set of independent letters may yield satisfactory results for certain tasks and certain data sets [9], one can easily name a

[*] contributed equally

wealth of other settings where this is unlikely to be the case. Examples are written texts, where the occurrence of a letter at a certain position is significantly constrained by the language [10], or DNA sequences, where the occurrence of a base at a certain position of a functional site on a chromosome influences its activity [11, 12]. Hence, there is a wealth of applications where the characterization of finite ensembles of sequences bearing statistical dependences is needed.

Inferring the probability distribution of the sequences from a finite ensemble of sequences becomes challenging already for moderate sequence lengths. In such situations, trading simplifications of the model against statistical strength has been shown to be potentially beneficial. For each model class, the joint probability of a sequence can be decomposed into a product of conditional probabilities of single symbols given all predecessors. While the class of Markov models of order d is based on the simplification that all conditional dependences except for those given the d previous symbols are dropped [2], the richer class of variable order Markov models (VOMMs) [13] makes this order context dependent. Most other approaches proposed to date also share the feature of dropping certain entries from the conditional probability distributions in a Markovian manner.

Bourguignon and Robelin [14] propose an alternative approach to the reduction of the dimension of the space of conditional distributions, where conditional independence assumptions are formed with respect to a partition of the conditions, i.e., a partition of context words, by means of a parsimonious context tree (PCT). Particular choices for the partition of the context words may result in conditional independence assumptions that coincide with those formed by a regular Markov model, as well as those formed by a variable order Markov model. The parallel with the VOMM is actually much further reaching, since PCTs can be understood as a generalization of the context trees that are used by VOMMs. However, parsimonious Markov models that use parsimonious context trees are in general not representable in a sheer Markovian manner, i.e., by dropping entries in the conditions.

Here, we aim at exploring the merits of this form of parsimony for modeling discrete sequential data of fixed length. We introduce inhomogeneous parsimonious Markov models (PMMs) based on a sequence of parsimonious context trees and follow a Bayesian approach for structure and parameter learning. Whereas parameter learning is straightforward, structure learning is challenging due to an overexponential number of possible tree structures. However, this optimization problem can be solved by an efficient dynamic programming algorithm, which generalizes the context tree maximization algorithm [1]. We apply inhomogeneous PMMs to the prediction of binding sites of the human transcription factor C/EBP [15], and investigate if the richer expressiveness of inhomogeneous PMMs might possibly lead to an improved prediction compared to inhomogeneous VOMMs.

2 Theory

In this section, we introduce inhomogeneous parsimonious Markov models in a Bayesian framework by defining likelihood and prior. We subsequently describe structure and parameter learning, and finally discuss the relation to variable order Markov models and further special cases.

We denote a single symbol by $x \in \mathcal{A}$, a sequence of length L by $\vec{x} = (x_1, \ldots, x_L)$, and a data set of N sequences of fixed length L by $\mathbf{x} = (\vec{x}_1, \ldots, \vec{x}_N)$. Further, we denote the power set of $\mathcal{A}$ by $\mathcal{P}(\mathcal{A})$, and $\mathcal{P}_{\geq 1}(\mathcal{A}) = \mathcal{P}(\mathcal{A}) \setminus \emptyset$. We call each element in $\mathcal{A}^d$ *context word* of length d.

2.1 Model

Similar to context trees, which are used by variable order Markov models for reducing and representing their parameter space, parsimonious context trees as proposed by Bourguignon and Robelin [14] are the central data structure of inhomogeneous PMMs. A PCT τ of depth d for alphabet $\mathcal{A}$ is a rooted, balanced tree. Each node of a PCT is labeled by a non-empty subset of $\mathcal{A}$, except for the root, which is labeled by the empty subset. The set of labels of all children of an arbitrary inner node forms a partition of $\mathcal{A}$.

It follows that the cross product of the symbol sets encountered along each path from a leaf to the root defines a non-empty subset of $\mathcal{A}^d$, which we call *context*. Hence, a context is a set of context words, and the set of the contexts of all leaves of a PCT forms a partition of $\mathcal{A}^d$. Thus, the PCT is a data structure that represents a partition of the whole set of context words. For example, the PCT of depth two for the four-letter DNA alphabet $\mathcal{A} = \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$ shown in Figure 1 encodes the contexts $\{\mathtt{A}\}\times\{\mathtt{A}\}$,$\{\mathtt{C},\mathtt{G}\}\times\{\mathtt{A}\}$, $\{\mathtt{T}\} \times \{\mathtt{A}\}$, $\{\mathtt{A},\mathtt{G}\}\times\{\mathtt{C},\mathtt{G},\mathtt{T}\}$, and $\{\mathtt{C},\mathtt{T}\}\times\{\mathtt{C},\mathtt{G},\mathtt{T}\}$. A PCT of depth d interpolates between two extreme cases: a *minimal* tree with only one leaf, which represents the union of all context words into one set, and a *maximal* tree with $|\mathcal{A}|^d$ leaves, each of which represents a single context word.

An inhomogeneous PMM of order d for sequences of length L is based on exactly L PCTs, which we denote by $\vec{\tau} = (\tau_1, \ldots, \tau_L)$. For the ease of presentation,

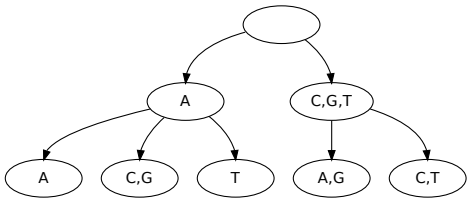

Fig. 1. Example PCT of depth 2 over DNA alphabet. It encodes the partition of all 16 possible context words into subsets {AA},{CA,GA},{TA},{AC,AG,AT,GC,GG,GT}, {CC,CG,CT,TC,TG,TT}.

we exclude the first d PCTs, which have an increasing depth of $0, \ldots, d-1$, from the following discussion. Since there is a bijective mapping from the leaves of a PCT to the corresponding contexts, we can perceive a PCT as a set of contexts as well as a set of leaf nodes. Hence, we denote a single context by c, the number of context words represented by a context by $|c|$, and the set of all contexts in a PCT by τ itself.

We denote the conditional probability of observing a symbol $a \in \mathcal{A}$ given that the concatenation of the preceding d symbols is in c by θ_{ca}. We denote the model parameters of a single position by $\Theta = \left(\tau, (\vec{\theta}_c)_{c \in \tau}\right)$ and all model parameters by $\vec{\Theta} = (\Theta_1, \ldots, \Theta_L)$. We now define the likelihood of an inhomogeneous PMM by

$$P(\mathbf{x}|\vec{\Theta}) = \prod_{\ell=1}^{L} \prod_{c \in \tau_\ell} \prod_{a \in \mathcal{A}} (\theta_{\ell ca})^{N_{\ell ca}}, \tag{1}$$

where $N_{\ell ca}$ is the number of occurrences of symbol a at position ℓ in all sequences of $\mathbf{x}$ where the concatenation of the symbols from position $\ell - d$ to position $\ell - 1$ is in c.

The likelihood of an inhomogeneous PMM is similar to that of a fixed order inhomogeneous Markov model since it is a product over all possible observations a for all possible contexts c at all possible positions ℓ. However, in contrast to fixed order inhomogeneous Markov models, where each c is a single context word, we here allow arbitrary sets of context words defined by the PCT τ_ℓ.

2.2 Prior

Assuming local and global parameter independence [16], we define the prior of an inhomogeneous PMM by

$$P(\vec{\Theta}) = P(\vec{\tau}) \prod_{\ell=1}^{L} \prod_{c \in \tau_\ell} P(\vec{\theta}_{\ell c}), \tag{2}$$

where $P(\vec{\tau})$ is the prior probability of all PCTs $\vec{\tau}$ (and could thus be referred to as structure prior) and $P(\vec{\theta}_{\ell c})$ is the prior over the probability parameters of one particular context c at position ℓ. We specify the structure prior by

$$P(\vec{\tau}) \propto \prod_{\ell=1}^{L} \kappa^{|\tau_\ell|}, \tag{3}$$

where $|\tau_\ell|$ denotes the number of leaves of τ_ℓ. It depends on one scalar hyper-parameter $\kappa \in (0, \infty)$, which can be used to influence the number of leaves and thus the complexity of the model, interpolating between the two extreme cases: When $\kappa \to +\infty$, the model that has maximal PCTs at all positions, and is thus equivalent to a fixed order Markov model, receives a prior probability of one. Conversely, when $\kappa \to 0$, the model that has minimal PCTs at all positions, and is thus equivalent to an independence model, receives full prior support. For

the local parameter priors $P(\vec{\theta}_{\ell c})$ we choose Dirichlet distributions with hyperparameters $\vec{\alpha}_{\ell c}$. In this work, we further restrict the parameter priors to symmetric Dirichlet distributions. Following the equivalent sample size concept [16], we obtain a natural computation of the pseudocounts from the equivalent sample size η that is inspired by Bayesian networks, namely $\alpha_{\ell ca} = \frac{\eta |c|}{|\mathcal{A}|^{d+1}}$.

2.3 Learning

In resonance with learning many other probabilistic graphical models, learning inhomogeneous PMMs consists of structure and parameter learning with the former being the more challenging task.

In order to learn the structure of the model, we intend to find the parsimonious context trees that maximize $P(\vec{\tau}|\mathbf{x})$. Since $P(\mathbf{x})$ is constant w.r.t. the tree structures, it is sufficient to maximize

$$P(\vec{\tau}, \mathbf{x}) = \int P(\mathbf{x}|\vec{\Theta}) P(\vec{\Theta}) d\vec{\Theta}_{\vec{\tau}}, \tag{4}$$

where $\vec{\Theta}$ denotes all parameters of the model, and $\vec{\Theta}_{\vec{\tau}}$ denotes here the conditional probability parameters within $\vec{\Theta}$ for given PCT structures $\vec{\tau}$. Due to global parameter independence (outer product of Eq. 2), we can decompose the structure learning problem into finding the optimal PCT for each position separately. Due to local parameter independence (inner product of Eq. 2), we can decompose the score of a PCT into a product of scores of its leaves. Solving the remaining integral, we obtain the optimization problem

$$\forall_{\ell=1}^{L} : \hat{\tau}_\ell := \operatorname*{argmax}_{\tau_\ell} \prod_{c \in \tau_\ell} \kappa \, \frac{\mathcal{B}(\vec{N}_{\ell c} + \vec{\alpha}_{\ell c})}{\mathcal{B}(\vec{\alpha}_{\ell c})}, \tag{5}$$

where $\mathcal{B}$ denotes the multinomial beta function. Hence, the target function is a product over local marginal likelihoods for all contexts multiplied by the structure prior hyperparameters for each context.

While the score for a given PCT can be computed easily, finding the optimal out of an overexponential number of possible PCTs (with respect to model order and alphabet size) without computing the score for every single PCT explicitly is challenging. This problem can be solved by a dynamic programming (DP) algorithm similar to the context tree maximization algorithm [1]. The algorithm runs on a data structure that we call the extended PCT of depth d and that we denote by $\mathcal{T}_d^{\mathcal{A}}$. In contrast to a PCT, the children of a node of an extended PCT do not form a partition of alphabet $\mathcal{A}$, but rather encompass all elements of $\mathcal{P}_{\geq 1}(\mathcal{A})$ (Figure 2). The leaves of an extended PCT are thus all possible leaves (identified by their label concatenation up to the root) that may occur in any PCT of same depth and alphabet.

Let $\mathcal{N}(\mathcal{T})$ denote the set of nodes of an extended PCT $\mathcal{T}$, n one element of $\mathcal{N}(\mathcal{T})$, and $r(\mathcal{T})$ the root of $\mathcal{T}$. Each node can be uniquely identified by the label concatenation on the path from that node up to the root of the extended PCT.

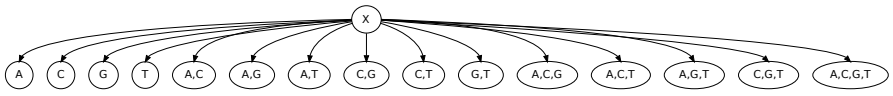

Fig. 2. Here, we show an arbitrary inner node (labeled by X) and its children in the extended PCT over the DNA alphabet. The labels of all children form $\mathcal{P}_{\geq 1}(\mathcal{A})$.

Let $s(n)$ denote the score of the optimal PCT subtree rooted at n. Let $\mathcal{C}(n)$ denote the set of all children of n in the extended PCT. Let $\mathcal{V}(\mathcal{C}(n))$ denote the set of all valid child combinations, i.e., all subsets of children whose labels form a partition of $\mathcal{A}$. Let further $\mathcal{L}(\mathcal{T})$ denote the leaves and $\mathcal{I}(\mathcal{T})$ the remaining inner nodes of $\mathcal{N}(\mathcal{T})$. Using this notation, we specify the dynamic programming approach in Algorithm 1, which consists of a single function for computing the optimal PCT subtree rooted at an arbitrary node of the extended PCT.

Algorithm 1 Dynamic programming for finding optimal PCT subtrees

`findOptimalSubtree`(n)
  if $n \in \mathcal{L}(\mathcal{T}_d^{\mathcal{A}})$ **then**
    $s(n) := \kappa \frac{\mathcal{B}(\vec{N}_{\ell n} + \vec{\alpha}_{\ell n})}{\mathcal{B}(\vec{\alpha}_{\ell n})}$
  end if
  if $n \in \mathcal{I}(\mathcal{T}_d^{\mathcal{A}})$ **then**
    for all $m \in \mathcal{C}(n)$ **do**
      `findOptimalSubtree`(m)
    end for
    for all $v \in \mathcal{V}(\mathcal{C}(n))$ **do**
      $s(v) := \prod_{m \in v} s(m)$
    end for
    $v^* := \operatorname{argmax}_{v \in \mathcal{V}(\mathcal{C}(n))} s(v)$
    $s(n) := s(v^*)$
    for all $m \in \mathcal{C}(n) \setminus v^*$ **do**
      remove m and subtree below
    end for
  end if

Applying this function to the root of the extended PCT, i.e., calling the function `findOptimalSubtree`$(r(\mathcal{T}_d^{\mathcal{A}}))$, yields the optimal PCT. The algorithm can be intuitively described as bottom-up reduction of the extended PCT towards a valid PCT by selecting at each inner node the locally optimal PCT subtree. The correctness of the algorithm follows from the property that the score of a PCT is a product of leaf scores (Eq. 5), which further implies that the score of

a PCT subtree rooted at node n depends (apart from its own structure) only on the nodes on the path from n up to the root, but is independent of the structure of the PCT subtrees rooted at siblings of n.

The complexity of the DP algorithm is given by the size of the extended PCT, which must be completely traversed, multiplied by the number of valid child combinations, for which a score must be computed in each inner node of the extended PCT. Whereas the former is exponential with the base being the number of possible subsets of $\mathcal{A}$, the latter is equivalent to the Bell number $B_{|\mathcal{A}|}$. Hence, we obtain a time complexity of roughly $\mathcal{O}\left(B_{|\mathcal{A}|}\left(2^{|\mathcal{A}|}-1\right)^d\right)$ for learning one PCT, stating that the complexity grows exponentially with model order d and overexponentially with alphabet size $\mathcal{A}$. Structure learning for an inhomogeneous PMM is linear in the sequence length as the DP algorithm is called $L-1$ times, once for each PCT of non-zero depth.

Having determined optimal PCTs, we estimate their conditional probability parameters according to the posterior mean [17]. It is in general defined by $\hat{\theta} = \int_\theta \theta P(\theta|\mathbf{x}) d\theta$ and yields for inhomogeneous PMMs

$$\forall_{\ell=1}^{L} \forall_{c\in\tau_\ell} \forall_{a\in\mathcal{A}} : \hat{\theta}_{\ell ca} := \frac{N_{\ell ca} + \alpha_{\ell ca}}{N_{\ell c\cdot} + \alpha_{\ell c\cdot}}. \tag{6}$$

A common task is prediction, i.e., computing the probability of a data point $\vec{x}_{N+1}$ after having observed N data points $(\vec{x}_1, \ldots, \vec{x}_N)$. In a Bayesian setting, this is done by integrating over the space of parameters, which is in resonance with structure learning, where the target function is the probability of the model structure given data, obtained by integrating over the space of parameters. Here, we obtain for inhomogeneous PMMs

$$P(\vec{x}_{N+1}|\mathbf{x}, \vec{\tau}) = \int P(\vec{x}_{N+1}|\vec{\Theta}) \prod_{\ell=1}^{L} P(\vec{\theta}^{\tau_\ell}|\mathbf{x}) d\vec{\Theta}_{\vec{\tau}}, \tag{7}$$

which is equivalent to computing $P(\vec{x}_{N+1}|\vec{\tau}, \hat{\Theta}_{1,\tau_1}, \ldots, \hat{\Theta}_{L,\tau_L})$, where $\hat{\Theta}_{\ell,\tau_\ell}$ is the posterior mean of the parameters (Eq. 6) of the PCT at position ℓ [16].

2.4 Special Cases

Context trees (CTs), which are used by variable order Markov models [13], are special cases of PCTs. Hence, inhomogeneous VOMMs are special cases of inhomogeneous PMMs. The differences between CTs and PCTs arise from a different concept of tree-building. Whereas the idea of building CTs is to *prune* a maximal tree by removing unimportant subtrees, the idea of PCTs is to *fuse* nodes if subtrees and corresponding conditional probability distributions are not sufficiently different. Since removing nodes can be also expressed by fusing them into one pseudo-node [18], CTs are special cases of PCTs (Figure 3). The opposite does not hold, though. There are many PCTs that represent a set of contexts that cannot be represented by CTs, since the notion of pruning yields several limitations of the possible CT structures that are relaxed by PCTs. Two structural

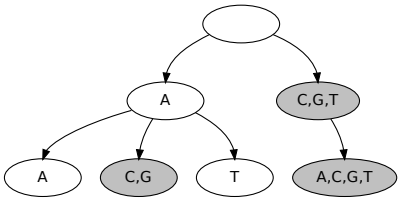

Fig. 3. Example CT of depth 2 over DNA alphabet. Pruned contexts are here shown as pseudo-nodes (displayed in gray) in order to achieve depth two for all possible contexts and thus allow a visualization of the CT in PCT-style.

features distinguish PCTs from CTs. First, an inner node in a PCT may have an arbitrary number of fused children as long as their labels form a partition of $\mathcal{A}$, whereas a CT allows at most one fused child (the pseudo-node). Second, a PCT allows arbitrary subtrees below a fused node, whereas a CT allows only a completely fused node as single child of a fused parent, which is equivalent to removing the entire subtree below the first occurrence of a fused node.

PCTs are more expressive than CTs, but this comes at the cost of a larger time complexity for structure learning, which limits the straightforward applicability of PMMs to problems with comparatively small alphabets. Even though there are plenty of such applications, with the most well known example being DNA and RNA sequence analysis, it might be desirable to benefit from more expressive tree structures also for problems where the alphabet size becomes a limiting factor.

The DP algorithm offers the possibility to reduce the allowed tree structures (and thus the space that must be searched for the optimal structure) by redefining $\mathcal{V}(\mathcal{C}(n))$, the set of allowed child combinations of an inner node n. In PCTs this is the set of all partitions of the alphabet, which yields the Bell number factor in time complexity. Restricting $\mathcal{V}(\mathcal{C}(n))$ to all partitions that include one fused node at maximum, is one of the two necessary restrictions for obtaining a CT. Enforcing it, but allowing a fused node to have more than one child, yields a data structure that lies in between CTs and PCTs in terms of complexity. Conversely, restricting $\mathcal{V}(\mathcal{C}(n))$ to only one choice – the partition that lumps all symbols together into one node – if n is already a fused node, represents the second necessary restriction for obtaining CTs, which could also be solely enforced.

Besides these two options, which are inspired by the special case CT, there are further possible modifications such as restricting the maximal number of children of n to a value smaller than $|\mathcal{A}|$ or restricting $\mathcal{V}(\mathcal{C}(n))$ based on the label and/or the location of n in the extended PCT. Hence, a plethora of model classes of almost arbitrary complexity could be defined and learned by slight modifications of Algorithm 1.

3 Experiments

In the experimental part of this work, we apply inhomogeneous PMMs to the prediction of DNA binding sites of the eukaryotic transcription factor C/EBP [15] and compare it with inhomogeneous VOMMs, both implemented within the open source Java library Jstacs [19]. For the sake of convenience, we drop the explicit reference to the inhomogeneity in the following discussion. The C/EBP data set consists of $N = 96$ DNA binding sites from human and mouse, retrieved from the TRANSFAC® database [20]. These binding sites are aligned sequences of fixed length $L = 12$ over the DNA alphabet $\mathcal{A} = \{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$.

3.1 Comparing Prediction Performance

The Bayesian learning approach for PMMs and VOMMs described above allows influencing the complexity of the model via the structure prior (Eq. 3). Since it is not immediately clear, which value of hyperparameter κ translates to which model complexity, specifying the structure prior manually is not a trivial task. While a uniform prior over all structures, which we obtain by setting $\kappa = 1$, may appear as a reasonable option in the absence of a priori knowledge, it might yield tree structures that are not optimal for prediction and related tasks.

Hence, in a first study we investigate the performance of third-order PMMs and VOMMs for different model complexities. Even though statistical models are often used for classification purposes (e.g. positive vs. negative sites), we here focus on prediction as the main challenge of many classification approaches. Evaluation by prediction has the advantage of not requiring the choice of a negative data set and a corresponding statistical model, which both may influence results heavily.

Since the C/EBP data set is rather small, we perform a leave-one-out cross validation (CV). In the i-th step, we remove the i-th sequence from the data set, learn a model (using $\eta = 1$ for the parameter prior) on the remaining 95 sequences and compute the predictive probability of the i-th sequence. We repeat this procedure for $i = 1, \ldots, 96$, compute the average number of leaves of the models, and compute the arithmetic mean of the 96 logarithmic predictive probabilities, as well as the corresponding standard error.

In Figure 4 we plot, for both model classes, the mean log predictive probability against the average model complexity, quantified by the number of leaves of all context trees in the model, which is proportional to the total number of parameters. We choose values of κ that cover the whole range of model complexity, interpolating from the minimal model with only 12 leaves (independence model) to the maximal model with 597 leaves (third-order Markov model).

We observe that for low model complexities of less than 50 leaves PMMs yield a substantially higher prediction than VOMMs. For high model complexities both approaches show a similar prediction, lower than the prediction achieved by a simple independence model, which indicates that overfitting occurs. These results are interesting in three aspects. First, PMMs are capable of utilizing statistical dependences in the data for improving prediction if the structure prior

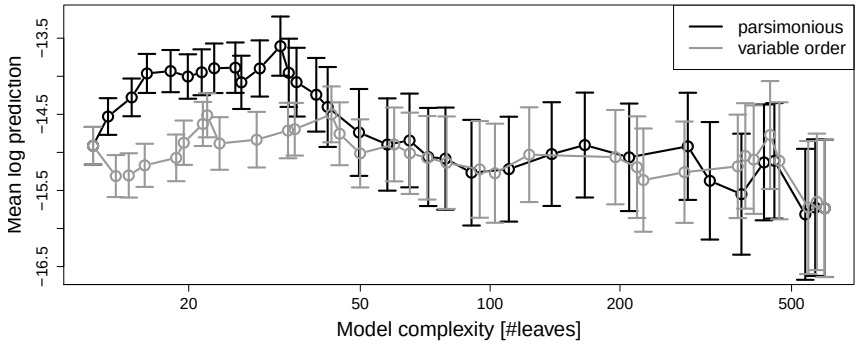

Fig. 4. We compare the prediction performance of third-order PMMs with third-order VOMMs. For both model classes, we plot the mean logarithmic prediction resulting from a leave-one-out cross validation experiment on the C/EBP data set against different model complexities (proportional to the number of parameters) obtained by varying the structure prior hyperparameter κ. Error bars depict double standard error.

is chosen well. Second, a uniform structure prior corresponds here to a model structure of approximately 110 leaves, which confirms that using it is not an optimal choice. Third, VOMMs are barely capable of benefiting from statistical dependences no matter how κ is chosen. This observation raises the question why PMMs are capable of finding a good compromise between modeling dependences and avoiding overfitting whereas VOMMs are not.

3.2 Comparing Tree Structures

In a second study, we attempt to answer that question by comparing the learned model structures of PMM and VOMM. We choose for both model classes the values of κ that yield the highest mean log prediction in the leave-one-out CV experiment of Figure 4. For the PMM, this is $\kappa = e^{-2.5}$ with an average number of leaves of 32.6 and a mean log prediction of -13.6, while for the VOMM this is $\kappa = e^{-1.8}$ with an average number of leaves of 42.8 and a mean log prediction of -14.5. We use these structure priors to learn two models on the complete C/EBP data set of 96 sequences and scrutinize the resulting models in the following.

The resulting PMM and VOMM have 32 and 43 leaves respectively, which is in resonance with the average number of leaves of the leave-one-out CV experi-

Table 1. Numbers of leaves for all trees of best third-order PMM and VOMM.

Position	1	2	3	4	5	6	7	8	9	10	11	12	Σ
PMM	1	1	4	3	2	3	1	2	6	5	3	2	32
VOMM	1	1	4	1	12	2	1	3	8	2	7	1	43

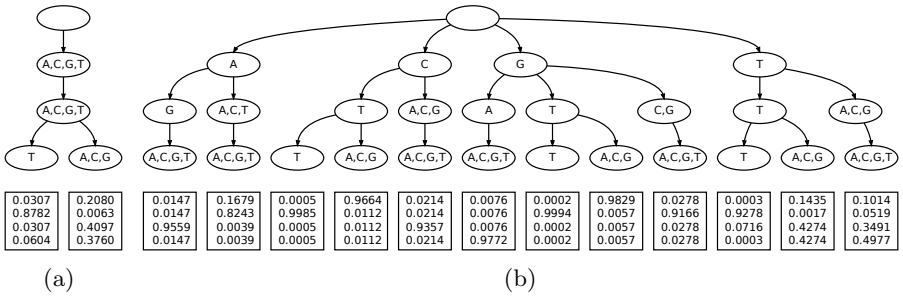

Fig. 5. We compare the PCT and the CT at position 5. We choose for both the PMM and the VOMM the optimal structure prior hyperparameter κ with respect to the leave-one-out experiment of Figure 4. Next, we learn both models using their respective optimal structure prior on the complete data set of 96 sequences and depict both the PCT of the PMM (a) and the CT of the VOMM (b) at position 5.

ment. First, we analyze how the total numbers of leaves of both models distribute over the 12 trees (Table 1). Even though the VOMM has more leaves than the PMM in total, this does not apply for each of the 12 individual trees. Whereas in some cases (positions 5, 8, 9, and 11), the CT of the VOMM is indeed more complex than the PCT of the PMM, in other cases (positions 4, 6, 10, and 12) the opposite holds, even though the absolute difference in complexity is here generally smaller, which is the reason of the overall higher complexity of the VOMM. Hence, it might be worthwhile to compare PCT and CT structures for both groups in detail. To this end, we choose position 5 and position 4, both representing extreme cases.

In Figure 5, we show the PCT of the PMM and the CT of the VOMM at position 5. Since tree structures can be only partially interpreted without knowing the underlying conditional probability distributions, we plot the conditional probabilities for each context, estimated according to Eq. 6, in rectangular boxes below the corresponding leaf node in lexicographical order of the observations.

The PCT in Figure 5(a) has only two leaves, so it partitions all context words into only two sets. The first and the second layer of the tree are completely fused, so the first and second predecessor symbol does not influence the probability distribution at position 5. At the third layer, however, the context words are partitioned into two subsets according to the observed symbol at the third predecessor (position 2). Observing a `T` at position 2 yields a high conditional probability of 0.8782 for finding a `C` at position 5, whereas any other symbol at position 2 yields a low conditional probability of 0.063 for a `C` at the fifth position. Conversely, for the second context, the conditional probability of finding `A`, `G`, and `T` is highly increased. This shows that there is a strong statistical dependence among positions 5 and 2, and a PCT is capable of exploiting it with only two parameters sets, which can be estimated comparatively robustly from 96 data points (partitioned into two sets of sizes 67 and 29 respectively).

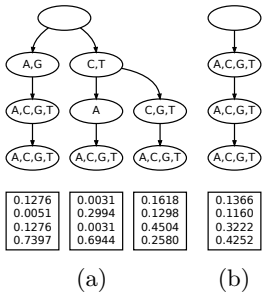

Fig. 6. We compare learned PCT (a) and CT (b) at position 4 of the C/EBP data set. The experimental setup is identical to that of Figure 5.

The CT in Figure 5(b) has twelve leaves, but many of the contexts represent only few occurrences of context words in the data set. For example, the first, fourth, and ninth leaf represent only a single sequence in the data set each. Hence, the reliability of the corresponding parameter estimates is highly questionable. The reason why such a context tree is learned despite the indication of overfitting is the strong statistical dependence among positions 5 and 2. Leaves number two, three, seven, and ten represent most of the context words that are combined in the first leaf of the PCT in Figure 5(a). But since a CT does not allow a split in the tree structure below a fused node, the only possibility to learn this third-order dependence is a broad tree with many dispensable parameter sets.

We conclude that one reason for the inability of the VOMM to effectively capture dependences in this data set is its structural limitation of not being capable of "skipping" a position, which may lead to strong overfitting if skipping positions were actually required.

In Figure 6, we display the PCT and CT at position 4. The CT of Figure 6(b) is completely pruned, resulting in a minimal tree corresponding to full statistical independence. At this position the VOMM does not suffer from overfitting, but it may neglect existing dependences.

The PCT at the same position has three leaves, resulting in three different parameter sets. Each leaf represents a substantial amount of sequences from the data set (24, 10, 62) so that the parameter estimates may not be completely unrealiable. We observe that the first leaf yields a high conditional probability of 0.7397 for a `T`, given the symbol of the preceding position being either `A` or `G`. The second and third leaf represent the other contexts that have either `C` or `T` at the previous position and differ in the second predecessor. The second leaf represents the subset of context words that have an `A` at position 2. The corresponding conditional probability of a `T` is 0.6944, whereas `A` and `C` rarely occur. However, if the symbol at the second predecessor is not `A`, then `G` has the highest probability at position 4 (third leaf).

This implies that a certain amount of statistical dependences exists among the fourth position of the C/EBP data set and its predecessors, and that these

dependences can be modeled – at least to some degree – by a PCT. A PCT is capable of splitting the contexts at any layer so that there is more than one fused child node per parent. This feature may be required to properly represent statistical dependences at position 4. Apparently a CT is not capable of representing these splits, so it here neglects statistical dependences completely. This indicates that VOMMs are not necessarily always overfitted compared to PMMs, but also the opposite, underfitting due to structural limitations, may occur.

We may conclude that, compared to the third-order PMM, the third-order VOMM is both over- and underfitted. The PMM is capable of using the full potential of the inhomogeneity of the model better than a VOMM, since it yields – on average over all positions – a better tradeoff between capturing dependences and reducing the parameter space.

3.3 Model Validation

The previous two experiments show that a PMM is capable of modeling dependences in a small real-world data set and how it finds a reasonable balance in avoiding both over- and underfitting due to its structural flexibility. However, as we have seen in Figure 4, the prediction performance depends on the choice of the structure prior, for which real a priori knowledge is rarely available.

Hence, we must devise a method that can automatically provide us with an adequate choice for κ in order to validate the model class against other alternatives. To this end, we perform in each step of the CV described in Section 3.1 another internal CV on the 95 training sequences. We then choose in the i-th step that κ that yields the highest mean log prediction in the CV on the 95 training data sequences, learn a model on that data set, and compute the predictive probability of the i-th sequence. Finally, we average all logarithmic predictive probabilities and use that single number for evaluating the performance of the model class.

In Figure 7, we compare PMMs, VOMMs, and inhomogeneous Markov models of orders 1-3. In addition, we consider the independence model, which neglects all dependences. Despite its simplicity, it is the most popular choice for modeling DNA binding sites in bioinformatics and in that field known as position weight matrix model [21, 22]. For the independence model and the fixed order Markov models, there is no internal cross validation.

We find that the independence model yields a mean log prediction value of -14.91. A first-order Markov model improves it to a value of -14.56, showing that taking into account first-order dependences is reasonable and beneficial. Second- and third-order Markov models yield a lower prediction than the independence model. This is not surprising since we expect overfitting for complex models when sample size is as small as 96 data points.

First- and second-order VOMM yield a prediction accuracy that is comparable with that of an independence model. Despite reducing the parameter space, they are – at least on this data set – not capable of utilizing statistical dependences effectively. The third-order VOMM yields an even lower prediction, comparable to a third-order Markov model, indicating that overfitting occurs.

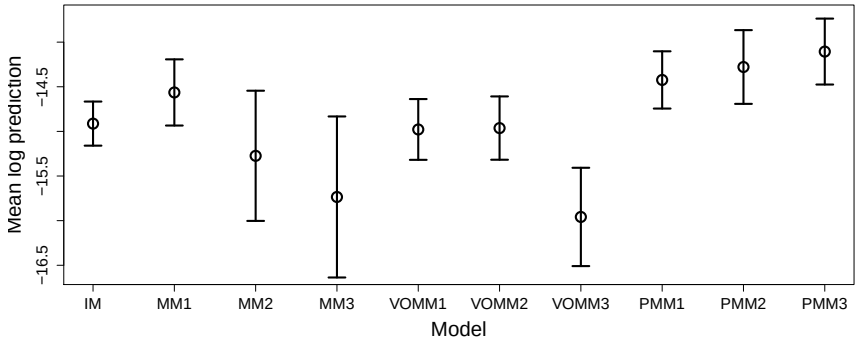

Fig. 7. We show the prediction performance of the independence model (IM), fixed order MMs, VOMMs, and PMMs of orders 1-3. The experimental setup is identical to that of Figure 4. For the parsimonious and variable order models, we perform an additional internal cross validation on the $N-1$ training sequences for determining the optimal structure prior hyperparameter κ.

This also shows that the internal CV fails in this case: At more than one position it selects in some iterations very complex and thus poorly generalizing tree structures, comparable to that in Figure 5(b).

The first-order PMM yields a mean log prediction value of -14.42, which is comparable to that of the first-order Markov model. Apparently, overfitting is not a serious problem for the first-order Markov model, so the potential reduction of the parameter space yields only a small improvement. However, in contrast to fixed order MMs, PMMs of second- and third-order continue to increase the prediction performance. The overall best prediction is achieved by a third-order PMM with a mean log prediction value of -14.1, which is slightly lower than the best prediction in Figure 4, but close to the average prediction within the range of reasonable complexities (15 to 40 leaves).

We summarize that PMMs yield a higher prediction of C/EBP binding sites than the independence model, than fixed order Markov models, and than variable order Markov models. Among the three PMMs, the third-order PMM yields the overall highest prediction. Hence, PMMs are capable of exploiting dependences in the small data set of only 96 sequences effectively, whereas the effectivity of VOMMs is harmed by their structural limitations. This makes it tempting to speculate that PMMs might be a useful model class for other types of sequential data as well, especially when certain dependences among non-neighboring positions exist and when the sample size is comparatively small.

4 Conclusions

In this work, we have introduced a new model class for sequential data in a discrete state space. Inhomogeneous parsimonious Markov models are capable of learning position-dependent statistical dependences from limited data by using parsimonious context trees for reducing the parameter space. Parsimony achieved by grouping context words is shown here to be promising from theoretical point of view as it generalizes the idea of context word pruning. However, the presented approach has an acceptable time complexity only for small alphabets, so additional constraints on the tree structures must be imposed when sequences of large alphabets are to be modeled. We have discussed how the learning algorithm can be adapted to incorporate these constraints, admitting an acceptable time complexity while retaining specific merits of parsimonious context trees.

Predicting functional DNA sequences is an important application where this model class can be used in a straightforward manner. In a case study on binding sites of the human transcription factor C/EBP, we have observed that inhomogeneous parsimonious Markov models yield more accurate predictions than the corresponding variable order Markov models. Scrutinizing the structural differences between the best models of both model classes, we found that strong third-order dependences but comparatively weak first- and second-order dependences exist at several positions. These are features that a parsimonious context tree can take into account with very few parameters, whereas a traditional context tree is limited by its structural constraints, either requiring substantially more parameters, yielding unreliable parameter estimates, or neglecting those dependences completely. We conclude that inhomogeneous parsimonious Markov models are a promising alternative to inhomogeneous Markov models and inhomogeneous variable order Markov models. The adaptation to different applications might possibly require additional algorithmic work, but taking such challenges might be worth the effort.

Acknowledgments. This work was funded by *Reisestipendium des allg. Stiftungsfonds der MLU Halle–Wittenberg*, DFG (grant no. GR 3526/1-2), and *CNRS/MPG GDRE in Systems Biology*.

References

1. P. Volf and F. Willems, "Context maximizing: Finding MDL decision trees," in *15th Symp. Inform. Theory Benelux*, pp. 192–200, May 1994.
2. T. Cover and J. Thomas, *Elements of Information Theory*. Wiley Interscience, 2 ed., 2006.
3. Y. Ding, "Statistical and Bayesian approaches to RNA secondary structure prediction," *RNA*, vol. 12, no. 3, pp. 323–331, 2006.
4. X. Xu, Y. Ji, and G. D. Stormo, "RNA sampler: a new sampling based algorithm for common RNA secondary structure prediction and structural alignment," *Bioinformatics*, vol. 23, no. 15, pp. 1883–1891, 2007.

5. J. R. Busch, P. A. Ferrari, A. G. Flesia, R. Fraiman, S. P. Grynberg, and F. Leonardi, "Testing statistical hypothesis on random trees and applications to the protein classification problem," *The Annals of Applied Statistics*, vol. 3, no. 2, pp. 542–563, 2009.
6. K.-J. Won, B. Ren, and W. Wang, "Genome-wide prediction of transcription factor binding sites using an integrated model," *Genome Biology*, vol. 11, no. 1, p. R7, 2010.
7. F. Ramus, M. Nespor, and J. Mehler, "Correlates of linguistic rhythm in the speech signal," *Cognition*, vol. 73, pp. 265–292, 1999.
8. A. Kolmogorov and N. Rychkova, "Analysis of russian verse rhythm, and probability theory," *Theory Probab. Appl.*, vol. 44, pp. 375–385, 2000.
9. J. Rissanen and G. Langdon, "Arithmetic coding," *IBM Journal of research and development*, vol. 23, pp. 149–162, 1979.
10. A. Galves, C. Galves, J. Garcia, N. Garcia, and F. Leonardi, "Context tree selection and linguistic rhythm retrieval from written texts," *Ann. Appl. Stat.*, vol. 6, no. 1, pp. 186–209, 2012.
11. G. D. Stormo, "DNA binding sites: representation and discovery," *Bioinformatics*, vol. 16, no. 1, pp. 16–23, 2000.
12. G. Bejerano and G. Yona, "Variations on probabilistic suffix trees: statistical modeling and prediction of protein families," *Bioinformatics*, vol. 17, no. 1, pp. 23–43, 2001.
13. J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory*, vol. 29, no. 5, pp. 656–664, 1983.
14. P. Bourguignon and D. Robelin, "Modèles de Markov parcimonieux," in *Proceedings of JOBIM*, 2004.
15. D. Ramji and P. Foka, "CCAAT/enhancer-binding proteins : structure, function and regulation," *Biochem. J.*, vol. 365, pp. 561–575, 2002.
16. G. Heckerman, D. Geiger, and D. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.
17. E. T. Jaynes, *Probability Theory: The Logic of Science*. Cambridge University Press, 2003.
18. P. Bühlmann and A. Wyner, "Variable length Markov chains," *Annals of Statistics*, vol. 27, pp. 480–513, 1999.
19. J. Grau, J. Keilwagen, A. Gohr, B. Haldemann, S. Posch, and I. Grosse, "Jstacs: A Java Framework for Statistical Analysis and Classification of Biological Sequences," *Journal of Machine Learning Research*, vol. 13, pp. 1967–1971, 2012.
20. V. Matys, E. Fricke, R. Geffers, E. Gling, M. Haubrock, R. Hehl, K. Hornischer, D. Karas, A. Kel, O. Kel-Margoulis, D. Kloos, S. Land, B. Lewicki-Potapov, H. Michael, R. Münch, I. Reuter, S. Rotert, H. Saxel, M. Scheer, S. Thiele, and E. Wingender, "TRANSFAC: transcriptional regulation, from patterns to profiles," *Nucleic Acids Research*, vol. 33, pp. 374–378, 2003.
21. G. Stormo, T. Schneider, and L. Gold, "Characterization of translational initiation sites in e.coli," *Nucleic Acids Research*, vol. 10, no. 2, pp. 2971–2996, 1982.
22. R. Staden, "Computer methods to locate signals in nucleic acid sequences," *Nucleic Acids Research*, vol. 12, pp. 505–519, 1984.