Context Tree Estimation for Not Necessarily Finite Memory Processes, via BIC and MDL

Imre Csiszár, *Fellow, IEEE,* and Zsolt Talata

***Abstract*— The concept of context tree, usually defined for finite memory processes, is extended to arbitrary stationary ergodic processes (with finite alphabet). These context trees are not necessarily complete, and may be of infinite depth. The familiar BIC and MDL principles are shown to provide strongly consistent estimators of the context tree, via optimization of a criterion for hypothetical context trees of finite depth, allowed to grow with the sample size n as $o(\log n)$. Algorithms are provided to compute these estimators in $O(n)$ time, and to compute them on-line for all $i \leq n$ in $o(n \log n)$ time.**

***Index Terms*— Bayesian Information Criterion (BIC), consistent estimation, context tree, Context Tree Maximization (CTM), infinite memory, Minimum Description Length (MDL), model selection.**

I. Introduction

In this paper, *process* always means a stationary ergodic stochastic process with finite alphabet. Processes are often described by the collection of the conditional probabilities of the possible symbols given the infinite pasts. When these probabilities depend on at most k previous symbols, the process is a Markov chain of order k.

The number of parameters of a general Markov chain grows exponentially with the order. A more efficient description is possible if the strings determining the conditional probabilities – referred to as *contexts* – are of variable length, sometimes substantially shorter than the order k. Models of this kind and the term *context tree* date back to Rissanen [10]. These models are also called finite memory sources or tree sources [13], [14], [16] or variable length Markov chains [3]. We note that the terms context and context tree appear in the literature in various senses. Here, the context tree of a finite memory process means, in effect, the minimal tree admitting a tree source representation of the process; the exact definition will be given in Section II.

As indicated above, the context tree model is typically used to more efficiently describe certain Markov chains (of finite order k) and, accordingly, the context tree has finite depth k. In this paper we drop the finite depth requirement, admitting also non-Markov processes. The term "infinite-depth context tree" appears in [18] in a different sense, as a tree assigned to an observed sequence, with an "indeterminate symbol" ε such that infinitely many ε's may precede a finite number of symbols of the true alphabet. A concept of generalized context tree, see [8] and references there, admits edges labeled by strings rather than single symbols. That concept is not used here, but similarly to [8] we drop the completeness requirement, often made in the literature, that each non-leaf node of the context tree has as many children as the alphabet size. If some strings have zero probability for the given process, these can not be contexts, and then the context tree need not be complete.

We address the problem of statistical estimation of the context tree in the indicated generality, based on an observed finite realization of the process, of length $n \to \infty$. This task, for finite depth context trees, has been considered, among others, in the references above. Variants of Rissanen's [10] "context" algorithm are popular. In particular, Bühlmann and Wyner [3] proved the consistency of such an algorithm not assuming a known prior bound on the depth of the context tree, but using a bound allowed to grow with n. They asserted that standard statistical methods as the Bayesian Information Criterion (BIC) of Schwarz [12] and the Minimum Description Length (MDL) principle of Rissanen [11], [2] were inappropriate for context tree estimation, due to computational infeasibility of comparing a very large number of hypothetical models. Still, Willems, Shtarkov and Tjalkens [15], [17] showed that time-consuming comparisons can be avoided by clever use of tree techniques. Their Context Tree Maximizing (CTM) algorithm computes in linear time the context tree estimator obtained by the version of MDL that uses the Krichevsky – Trofimov (KT) codelength [7], and this estimator is consistent, as they proved assuming a known upper bound on the depth of the context tree. Similar results were obtained also by Nohre [9]. Recent results on consistent context tree estimation in linear time, assuming finite depth but no known upper bound on it, appear in [1], [8]. These references use tools as the Burrows – Wheeler transform or generalized context trees.

We are not aware of prior results on context tree estimation via BIC. While BIC is commonly regarded as an approximate version of MDL, this is justified only when a finite number of model classes is considered, see [4]. We note that much of the literature of context tree models is motivated by universal source coding. In particular, CTM is a modification of the

Manuscript received September 3, 2004; revised July 15, 2005. This work was supported by the Hungarian National Foundation for Scientific Research, Grants T26041, T32323, TS40719, T046376. This paper was presented at the IEEE International Symposium on Information Theory, Adelaide, Australia, Sep 2005.

I. Csiszár is with the Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, POB 127, H-1364 Budapest, Hungary (e-mail: csiszar@renyi.hu, web: http://www.renyi.hu/˜csiszar).

Zs. Talata is with the Stochastics Research Group, Hungarian Academy of Sciences, POB 127, H-1364 Budapest, Hungary (e-mail: zstalata@renyi.hu, web: http://www.renyi.hu/˜zstalata).

celebrated Context Tree Weighting data compression algorithm of Willems, Shtarkov and Tjalkens [16].

In this paper, we prove that both MDL with KT codelength and BIC provide strongly consistent estimators of the context tree if the set of candidate context trees is suitably chosen; finiteness or completeness of the true context tree is not required. Moreover, these estimators can be implemented in linear time. The set of candidate context trees is specified by a bound on the length of the hypothetical contexts, allowed to grow as $o(\log n)$, and in one case by an additional condition on their occurrences in the observed sample. Strong consistency means in the finite depth case that the estimated context tree is equal to the true one, eventually almost surely as $n \to \infty$, while otherwise, that the estimated context tree truncated at any fixed level is equal to the true one truncated at the same level, eventually almost surely as $n \to \infty$.

For order estimation of Markov chains, it is well known that BIC and MDL, both with the KT and the Normalized Maximum Likelihood (NML) codelength, are strongly consistent when the number of candidate model classes is finite, that is, when there is a known upper bound on the order [6]. The consistency of the BIC order estimator without such prior bound has been proved by Csiszár and Shields [4]. That paper also contains a counterexample to the consistency of the KT and NML versions of MDL without any bound on the order, or with a bound depending on the sample size n, equal to a sufficiently large constant times $\log n$. The consistency of the latter order estimators with bound $o(\log n)$ resp. $O(\log n)$ on the order was proved by Csiszár [5].

Linear time implementation of our context tree estimators is achieved via the CTM method [15], [17]. This has been developed for the KT version of MDL, it appears a new observation that also the BIC estimator admits a CTM-like implementation. The same does not seem to hold for the NML version of MDL, this is why the latter is not considered in this paper.

By our consistency result, if the context tree of a process has finite depth, it can be exactly recovered, with probability 1, when the sample size is large enough; the sample size actually needed remains, however, unknown. A heuristic rule might be to stop when the estimated context tree "stabilizes", that is, it remains unchanged when the sample size n runs over a large interval. The last result in this paper shows that (slightly modified versions of) our estimators can be calculated on-line in such a way that $o(n \log n)$ time suffices to calculate them for all sample sizes $i \leq n$. This implies that the above stopping rule can be implemented with only a small increment in the order of required computations.

The structure of the paper is the following. In Section II we introduce the notation and definitions, and formulate the results for the BIC estimator and KT estimator about strong consistency and computational complexity. In Section III we prove the consistency theorems. In Section IV we introduce the algorithms for calculating the estimators, and establish their claimed computational complexity both for off-line and on-line calculations. Section V contains some remarks on the results.

II. Notation and Statement of the Main Results

For a finite set A we denote its cardinality by $|A|$. A *string* $s = a_m a_{m+1} \dots a_n$ (with $a_i \in A$, $m \leq i \leq n$) is denoted also by a_m^n; its length is $l(s) = n - m + 1$. The empty string is denoted by $\varnothing$, its length is $l(\varnothing) = 0$. The concatenation of the strings u and v is denoted by uv. We say that a string v is a *postfix* of a string s, denoted by $s \succeq v$, when there exists a string u such that $s = uv$. For a proper postfix, that is, when $s \neq v$, we write $s \succ v$. A postfix of a semiinfinite sequence $a_{-\infty}^{-1} = \dots a_{-k} \dots a_{-1}$ is defined similarly. Note that in the literature $\succ$ more often denotes the prefix relation. Also, often the term suffix is used instead of postfix.

A set $\mathcal{T}$ of strings, and perhaps also of semiinfinite sequences, is called a *tree* if no $s_1 \in \mathcal{T}$ is a postfix of any other $s_2 \in \mathcal{T}$.

Each string $s = a_1^k \in \mathcal{T}$ is visualized as a path from a leaf to the root (drawn with the root at the top), consisting of k edges labeled by the symbols $a_1 \dots a_k$. A semiinfinite sequence $a_{-\infty}^{-1} \in \mathcal{T}$ is visualized as an infinite path to the root, see Fig. 1. The strings $s \in \mathcal{T}$ are identified also with the leaves of the tree $\mathcal{T}$, *leaf* s is the leaf connected with the root by the path visualizing s as above. Similarly, the *nodes* of the tree $\mathcal{T}$ are identified with the finite postfixes of all (finite or infinite) $s \in \mathcal{T}$, the root being identified with the empty string $\varnothing$. The *children* of a node s are those strings as, $a \in A$, that are themselves nodes, that is, postfixes of some $s' \in \mathcal{T}$.

The tree $\mathcal{T}$ is *complete* if each node except the leaves has exactly $|A|$ children. A weaker property we shall need is *irreducibility*, which means that no $s \in \mathcal{T}$ can be replaced by a proper postfix without violating the tree property. The family of irreducible trees will be denoted by $\mathcal{I}$.

Denote $d(\mathcal{T})$ the depth of the tree $\mathcal{T}$: $d(\mathcal{T}) = \max\{\, l(s), s \in \mathcal{T} \,\}$. Let $\mathcal{T}\big|_K$ denote the tree $\mathcal{T}$ truncated at level K:

$$\mathcal{T}\big|_K = \{\, s' : s' \in \mathcal{T} \text{ with } l(s') \leq K \\ \text{or } s' \text{ is a } \lfloor K \rfloor\text{-length postfix of some } s \in \mathcal{T} \,\}. \quad (1)$$

Consider a stationary ergodic stochastic process $\{\, X_i, -\infty < i < +\infty \,\}$ with finite alphabet A. Write

$$Q(a_m^n) = \text{Prob}\{\, X_m^n = a_m^n \,\},$$

and, if $s \in A^k$ has $Q(s) > 0$, write

$$Q(a|\, s) = \text{Prob}\{\, X_0 = a \mid X_{-k}^{-1} = s \,\}.$$

A process as above will be referred to as process Q.

Definition 2.1: A string $s \in A^k$ is a *context* for a process Q if $Q(s) > 0$ and

$$\text{Prob}\{\, X_0 = a \mid X_{-\infty}^{-1} = x_{-\infty}^{-1} \,\} = Q(a|\, s), \quad \text{for all } a \in A,$$

whenever s is a postfix of the semiinfinite sequence $x_{-\infty}^{-1}$, and no proper postfix of s has this property. An *infinite context* is a semiinfinite sequence $x_{-\infty}^{-1}$ whose postfixes x_{-k}^{-1}, $k = 1, 2, \dots$ are of positive probability but none of them is a context.

Clearly, the set of all contexts is a tree. It will be called the *context tree* of the process Q, denoted by $\mathcal{T}_0$.

Remark 2.2: The context tree $\mathcal{T}_0$ has to be complete if $Q(s) > 0$ for all strings s. In general, for each node s of $\mathcal{T}_0$ which is not a leaf, exactly those as, $a \in A$, are the children of s for which $Q(as) > 0$. Moreover, Definition 2.1 implies that the context tree is always irreducible, $\mathcal{T}_0 \in \mathcal{I}$.

When the context tree has depth $d(\mathcal{T}_0) = k_0 < \infty$, the process Q is a Markov chain of order k_0. In this case the context tree leads to a parsimonious description of the process, because a collection of $(|A|-1)|\mathcal{T}_0|$ transition probabilities suffices to describe the process, instead of $(|A|-1)|A|^{k_0}$ ones. Note that the context tree of an i.i.d. process consists of the root $\varnothing$ only, thus $|\mathcal{T}_0| = 1$.

Example 2.3: (Renewal Process). Let $A = \{0, 1\}$ and suppose that the distances between the occurrences of 1's are i.i.d. Denote p_j the probability that this distance is j, that is, $p_j = Q(10^{j-1}1)/Q(1)$. Then for $k \geq 1$ we have $Q(10^{k-1}) = (1/Q(1)) \sum_{i=k}^{\infty} p_i \triangleq q_k$, $Q_k = Q(1|\, 10^{k-1}) = p_k/q_k$. Let $Q_0 = Q(1) \triangleq q_0$. Denote k_0 the smallest integer such that Q_k is constant for $k \geq k_0$ with $q_k > 0$, or $k = \infty$ if no such integer exists. Then the contexts are the strings 10^{i-1}, $i \leq k_0$, and the string 0^{k_0} (if $k_0 < \infty$) or the semiinfinite sequence 0^{∞} (if $k_0 = \infty$), see Fig. 1.

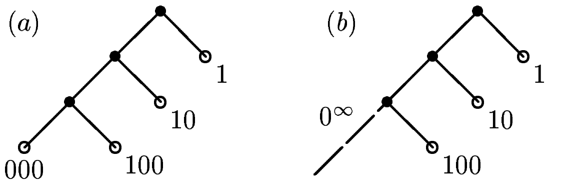

Fig. 1. Context tree of a renewal process. (a) $k_0 = 3$. (b) $k_0 = \infty$.

In this paper, we are concerned with the statistical estimation of the context tree $\mathcal{T}_0$ from the sample x_1^n, a realization of X_1^n. We demand strongly consistent estimation. We mean by this in the case $d(\mathcal{T}_0) < \infty$ that the estimated context tree equals $\mathcal{T}_0$ eventually almost surely as $n \to \infty$, while otherwise that the estimated context tree truncated at any fixed level K equals $\mathcal{T}_0\big|_K$ eventually almost surely as $n \to \infty$, see (1). Here and in the sequel, "eventually almost surely" means that with probability 1 there exists a threshold n_0 (depending on the infinite realization x_1^{∞}) such that the claim holds for all $n \geq n_0$.

Let $N_n(s, a)$ denote the number of occurrences of the string $s \in A^{l(s)}$ followed by the letter $a \in A$ in the sample x_1^n, where s is supposed to be of length at most $D(n)$, specified later, and – for technical reasons – only the letters in positions $i > D(n)$ are considered:

$$N_n(s,a) = \left| \left\{ i :\ D(n) < i \leq n,\, x_{i-l(s)}^{i-1} = s,\, x_i = a \right\} \right| .$$

The number of such occurrences of s is denoted by $N_n(s)$:

$$N_n(s) = \left| \left\{ i :\ D(n) < i \leq n,\, x_{i-l(s)}^{i-1} = s \right\} \right| .$$

Given a sample x_1^n, a *feasible tree* is any tree $\mathcal{T}$ of depth $d(\mathcal{T}) \leq D(n)$ such that $N_n(s) \geq 1$ for all $s \in \mathcal{T}$, and each string s' with $N_n(s') \geq 1$ is either a postfix of some $s \in \mathcal{T}$ or has a postfix $s \in \mathcal{T}$. A feasible tree $\mathcal{T}$ is called *r-frequent* if $N_n(s) \geq r$ for all $s \in \mathcal{T}$. The family of all feasible respectively r-frequent trees is denoted by $\mathcal{F}_1(x_1^n, D(n))$ respectively $\mathcal{F}_r(x_1^n, D(n))$.

Clearly,

$$\sum_{a \in A} N_n(s,a) = N_n(s), \quad \text{and} \quad \sum_{s \in \mathcal{T}} N_n(s) = n - D(n)$$

for any feasible tree $\mathcal{T}$. Regarding such a tree $\mathcal{T}$ as the context tree of a hypothetical process Q', the probability of the sample x_1^n can be written as

$$Q'(x_1^n) = Q'(x_1^{D(n)}) \prod_{s \in \mathcal{T},\, a \in A} Q'(a|\, s)^{N_n(s,a)}.$$

With some abuse of terminology, for a hypothetical context tree $\mathcal{T} \in \mathcal{F}_1(x_1^n, D(n))$ we define the maximum likelihood $\mathrm{ML}_{\mathcal{T}}(x_1^n)$ as the maximum in $Q'(a|\, s)$ of the second factor above, that is,

$$\mathrm{ML}_{\mathcal{T}}(x_1^n) = \prod_{s \in \mathcal{T},\, N_n(s) \geq 1} \ \prod_{a \in A} \left(\frac{N_n(s,a)}{N_n(s)} \right)^{N_n(s,a)} . \qquad (2)$$

We investigate two information criteria to estimate $\mathcal{T}_0$, both motivated by the MDL principle. An information criterion assigns a score to each hypothetical model (here, context tree) based on the sample, and the estimator will be that model whose score is minimal.

Definition 2.4: Given a sample x_1^n, the *BIC* for a feasible tree $\mathcal{T}$ is

$$\mathrm{BIC}_{\mathcal{T}}(x_1^n) = -\log \mathrm{ML}_{\mathcal{T}}(x_1^n) + \frac{(|A|-1)|\mathcal{T}|}{2} \log n.$$

Logarithms are to the base e.

Remark 2.5: Characteristic for BIC is the "penalty term" half the number of free parameters times $\log n$. Here, a process Q with context tree $\mathcal{T}$ is described by the conditional probabilities $Q(a|\, s)$, $a \in A$, $s \in \mathcal{T}$, and $(|A|-1)|\mathcal{T}|$ of these are free parameters when the tree $\mathcal{T}$ is complete. For a process with an incomplete context tree, the probabilities of certain strings must be 0, hence the number of free parameters is typically smaller than $(|A|-1)|\mathcal{T}|$ when $\mathcal{T}$ is not complete. Thus, Definition 2.4 involves a slight abuse of terminology. We note that replacing $(|A|-1)/2$ in Definition 2.4 by any $c > 0$ would not affect the results below and their proofs. In the literature, context trees are often required to be complete. This can be achieved by adding dummy edges if necessary, but this increases the penalty term in Definition 2.4, and the analog of Theorem 2.6 below appears a weaker result for completed context trees.

It is known [4] that for estimating the order of Markov chains, the BIC estimator is consistent without any restriction on the hypothetical orders. The Theorem below does need a bound on the depth of the hypothetical context trees. Still, as this bound grows with the sample size n, no a priori bound on the size of the unknown $\mathcal{T}_0$ is required, in fact, even $d(\mathcal{T}_0) = \infty$ is allowed. Note also that the presence of this bound decreases computational complexity.

Theorem 2.6: In the case $d(\mathcal{T}_0) < \infty$, the BIC estimator

$$\widehat{\mathcal{T}}_{\mathrm{BIC}}(x_1^n) = \arg \min_{\mathcal{T} \in \mathcal{F}_1(x_1^n, D(n)) \cap \mathcal{I}} \mathrm{BIC}_{\mathcal{T}}(x_1^n)$$

with $D(n) = o(\log n)$, satisfies

$$\widehat{\mathcal{T}}_{\mathrm{BIC}}(x_1^n) = \mathcal{T}_0$$

eventually almost surely as $n \to \infty$.

In general case, this estimator satisfies for any constant K

$$\widehat{\mathcal{T}}_{\mathrm{BIC}}(x_1^n)\big|_K = \mathcal{T}_0\big|_K$$

eventually almost surely as $n \to \infty$.

Proof: See Section III. □

Remark 2.7: Here and in Theorem 2.10 below, the indicated minimum is certainly attained, as the number of feasible trees is finite, but the minimizer is not necessarily unique; in that case, either minimizer can be taken as $\arg\min$.

The other information criterion we consider is the Krichevsky – Trofimov codelength [7], [16].

Definition 2.8: Given a sample x_1^n, the *KT* criterion for a feasible tree $\mathcal{T}$ is

$$\mathrm{KT}_{\mathcal{T}}(x_1^n) = -\log P_{\mathrm{KT},\mathcal{T}}(x_1^n),$$

where

$$P_{\mathrm{KT},\mathcal{T}}(x_1^n) = \frac{1}{|A|^{D(n)}}$$

$$\prod_{s\in\mathcal{T}} \frac{\prod_{a:N_n(s,a)\geq 1}\left[\left(N_n(s,a)-\frac{1}{2}\right)\left(N_n(s,a)-\frac{3}{2}\right)\cdots\left(\frac{1}{2}\right)\right]}{\left(N_n(s)-1+\frac{|A|}{2}\right)\left(N_n(s)-2+\frac{|A|}{2}\right)\cdots\left(\frac{|A|}{2}\right)}$$

is the KT-probability of x_1^n corresponding to $\mathcal{T}$.

Remark 2.9: The coding distribution $P_{\mathrm{KT},\mathcal{T}}$ is nearly optimal for the class of processes with context tree $\mathcal{T}$, in the sense that the codelengths $\lceil -\log P_{\mathrm{KT},\mathcal{T}}(x_1^n)\rceil$ (using base 2 rather than base e logarithm) minimize the worst case average redundancy for this class, up to an additive constant.

For estimating the order of Markov chains, the consistency of the KT estimator has been proved when the hypothetical orders are $o(\log n)$ [5], while without any bound on the order, or with a bound equal to a sufficiently large constant times $\log n$, a counterexample to its consistency is known [4].

Theorem 2.10: In the case $d(\mathcal{T}_0) < \infty$, the KT estimator

$$\widehat{\mathcal{T}}_{\mathrm{KT}}(x_1^n) = \arg \min_{\mathcal{T}\in\mathcal{F}_1(x_1^n, D(n))\cap\mathcal{I}} \mathrm{KT}_{\mathcal{T}}(x_1^n)$$

with $D(n) = o(\log n)$, satisfies

$$\widehat{\mathcal{T}}_{\mathrm{KT}}(x_1^n) = \mathcal{T}_0$$

eventually almost surely as $n \to \infty$.

In general case, the KT estimator

$$\widehat{\mathcal{T}}_{\mathrm{KT}}(x_1^n) = \arg \min_{\mathcal{T}\in\mathcal{F}_{n^\alpha}(x_1^n, D(n))\cap\mathcal{I}} \mathrm{KT}_{\mathcal{T}}(x_1^n)$$

with $D(n) = o(\log n)$ and arbitrary $0 < \alpha < 1$, satisfies for any constant K

$$\widehat{\mathcal{T}}_{\mathrm{KT}}(x_1^n)\big|_K = \mathcal{T}_0\big|_K$$

eventually almost surely as $n \to \infty$.

Proof: See Section III. □

Remark 2.11: Strictly speaking, the MDL principle would require to minimize the "codelength" $\mathrm{KT}_{\mathcal{T}}(x_1^n)$ incremented by an additional term, the "codelength of $\mathcal{T}$" (called the cost of $\mathcal{T}$ in [16]). This additional term is omitted, since this does not affect the consistency result.

Corollary 2.12: The vector of the empirical conditional probabilities,

$$\widehat{Q}_{\widehat{\mathcal{T}}}(a|\,s) = \frac{N_n(s,a)}{N_n(s)}, \quad a \in A,\, s \in \widehat{\mathcal{T}},$$

converges to that of the true conditional probabilities $Q(a|s)$, $a \in A$, $s \in \mathcal{T}_0$ almost surely as $n \to \infty$, where $\widehat{\mathcal{T}}$ is either the BIC estimator or the KT estimator.

Proof: Immediate from Theorems 2.6, 2.10 and the ergodic theorem. □

In practice, it is infeasible to calculate estimators via computing the value of an information criterion for each model, since the number of the hypothetical context trees is very large. However, an algorithm in Section IV admits finding the considered estimators with practical computational complexity.

We consider both off-line and on-line methods, in the latter case with a slight modification of the estimators. Note that on-line calculation of the estimator is useful when the sample size is not fixed but we keep sampling until the estimator becomes "stable", say it remains constant when the sample size is doubled.

As usual, see [1], [8], we assume that the computations are done in registers of size $O(\log n)$.

Theorem 2.13: The number of computations needed to determine the BIC estimator and the KT estimator in Theorems 2.6 and 2.10 for a given sample x_1^n is $O(n)$, and this can be achieved storing $O(n^\varepsilon)$ data, where $\varepsilon > 0$ is arbitrary.

Proof: See Section IV. □

On-line algorithms are considered with the following minor modifications of the estimators, which obviously do not affect the consistency. In the BIC penalty term, $\log n$ is replaced by $\lfloor \log_{|A|} n\rfloor \log |A|$, and in the second kind of KT estimator in Theorem 2.10 $\mathcal{F}_{n^\alpha}(x_1^n, D(n))$ is replaced by $\mathcal{F}_r(x_1^n, D(n))$ with $r = e^{\alpha\lfloor \log_{|A|} n\rfloor}$. No modification is needed in the first kind of KT estimator whose consistency has been proved for the case $d(\mathcal{T}_0) < \infty$.

Theorem 2.14: Suppose $D(n) = o(\log n)$ is a non-decreasing function of n. Adopting the above modifications, the number of computations needed to determine the BIC estimator in Theorem 2.6 or the KT estimator in Theorem 2.10, simultaneously for all subsamples x_1^i, $i \leq n$, is $o(n \log n)$, and this can be achieved storing $O(n^\varepsilon)$ data at any time, where $\varepsilon > 0$ is arbitrary.

Proof: See Section IV. □

Remark 2.15: Of course, the $O(n^\varepsilon)$ storage does not include storage of the context tree estimators for all $i \leq n$; note that for the indicated purpose of deciding when to stop sampling, it suffices to keep track of the last instance when the estimator has changed.

III. Consistency of the KT and BIC estimators

In this section we prove the consistency theorems stated in Section II.

Proof of Theorem 2.6

It suffices to prove the second assertion of the Theorem. Fix an arbitrary constant K. It suffices to show that if $\mathcal{T}\big|_K \neq \mathcal{T}_0\big|_K$ for some $\mathcal{T} \in \mathcal{F}_1(x_1^n, D(n)) \cap \mathcal{I}$ then there exists a modification $\mathcal{T}'$ of $\mathcal{T}$ also satisfying $\mathcal{T}' \in \mathcal{F}_1(x_1^n, D(n)) \cap \mathcal{I}$ such that

$$\mathrm{BIC}_{\mathcal{T}}(x_1^n) > \mathrm{BIC}_{\mathcal{T}'}(x_1^n), \tag{3}$$

simultaneously for all considered trees $\mathcal{T}$, eventually almost surely as $n \to \infty$.

According to (2), the maximum likelihood factorizes as

$$\mathrm{ML}_{\mathcal{T}}(x_1^n) = \prod_{s \in \mathcal{T}} \widetilde{P}_{\mathrm{ML},\, s}(x_1^n), \tag{4}$$

where

$$\widetilde{P}_{\mathrm{ML},\, s}(x_1^n) = \begin{cases} \prod_{a \in A} \left[\frac{N_n(s,a)}{N_n(s)} \right]^{N_n(s,a)} & \text{if } N_n(s) \geq 1, \\ 1 & \text{if } N_n(s) = 0. \end{cases} \tag{5}$$

Using this and the definition of BIC, see Definition 2.4, (3) is equivalent to

$$\sum_{s \in \mathcal{T}} \log \widetilde{P}_{\mathrm{ML},\, s}(x_1^n) - \sum_{s' \in \mathcal{T}'} \log \widetilde{P}_{\mathrm{ML},\, s'}(x_1^n) < \frac{(|A|-1)}{2} \left(|\mathcal{T}| - |\mathcal{T}'| \right) \log n. \tag{6}$$

Since $\mathcal{T}$ is a feasible tree by assumption, so is also $\mathcal{T}\big|_K$ defined by (1). For n sufficiently large, so that $N_n(s) \geq 1$ for all s with $l(s) \leq K$, $Q(s) > 0$, it follows by Remark 2.2 that $\mathcal{T}_0\big|_K$ is feasible, as well. Hence, the indirect assumption $\mathcal{T}\big|_K \neq \mathcal{T}_0\big|_K$ implies that there exist strings $\tilde{s} \in \mathcal{T}\big|_K$ and $\tilde{s_0} \in \mathcal{T}_0\big|_K$ such that either $\tilde{s} \prec \tilde{s_0}$ (underestimation) or $\tilde{s} \succ \tilde{s_0}$ (overestimation). Equivalently, there exist $s \in \mathcal{T}$ and $s_0 \in \mathcal{T}_0$ such that either (a) $l(s) < K$, $s \prec s_0$ or (b) $l(s_0) < K$, $s_0 \prec s$.

We claim that a modification $\mathcal{T}'$ of $\mathcal{T}$ with the required properties is

$$\mathcal{T}' = (\,\mathcal{T} \backslash \{s\}\,) \cup \widetilde{\mathcal{T}} \tag{7}$$

in case (a), with $\widetilde{\mathcal{T}}$ as in Lemma 3.1 below, and

$$\mathcal{T}' = (\,\mathcal{T} \backslash \widetilde{\mathcal{T}}\,) \cup \{w\} \tag{8}$$

in case (b), with $\widetilde{\mathcal{T}}$ and w as in Lemma 3.2 below. The properties of $\widetilde{\mathcal{T}}$ in Lemmas 3.1, 3.2 immediately imply that the condition $\mathcal{T}' \in \mathcal{F}_1(x_1^n, D(n)) \cap \mathcal{I}$ is satisfied in both cases (a), (b) (in case (a), $\mathcal{T}' \in \mathcal{F}_1(x_1^n, D(n))$ holds by the ergodic theorem, eventually almost surely as $n \to \infty$). Thus, it remains to check (6) for this choice of $\mathcal{T}'$.

In case (a), for $\mathcal{T}'$ given by (7) we have $|\mathcal{T}| - |\mathcal{T}'| = 1 - |\widetilde{\mathcal{T}}|$, and the left hand side of (6) is equal to that of (9) below. By Lemma 3.1, the latter is less than $-c\,n$, eventually almost surely as $n \to \infty$, and thus (6) certainly holds. Regarding simultaneity for all considered trees $\mathcal{T}$, note that $\widetilde{\mathcal{T}}$ corresponding to a particular $\mathcal{T}$ may be chosen depending on s only, and the number of strings s with $l(s) \leq K$ is finite.

In case (b), for $\mathcal{T}'$ given by (8) we have $|\mathcal{T}| - |\mathcal{T}'| = |\widetilde{\mathcal{T}}| - 1$, and the left hand side of (6) is equal to that of (10) below. Hence by Lemma 3.2, (6) is satisfied also in this case, eventually almost surely as $n \to \infty$ for all considered $\mathcal{T}$. □

Lemma 3.1: For any proper postfix s of some $s_0 \in \mathcal{T}_0$, there exists an irreducible tree $\widetilde{\mathcal{T}}$ with $d(\widetilde{\mathcal{T}}) < \infty$ such that $u \succ s$ and $Q(u) > 0$ for each $u \in \widetilde{\mathcal{T}}$, each $v \succeq s$ with $Q(v) > 0$ has a postfix in $\widetilde{\mathcal{T}}$, and

$$\log \widetilde{P}_{\mathrm{ML},\, s}(x_1^n) - \sum_{u \in \widetilde{\mathcal{T}}} \log \widetilde{P}_{\mathrm{ML},\, u}(x_1^n) < -c\,n, \tag{9}$$

eventually almost surely as $n \to \infty$, where $c > 0$ is a sufficiently small constant.

Proof: Given $s \prec s_0 \in \mathcal{T}_0$, denote by s_{0l} the l-length postfix of s_0. Let

$$\widetilde{\mathcal{T}} = \{\, s_{0L+1} \,\} \cup \{\, as_{0l} : l(s) \leq l \leq L,\, a \in A,\, as_{0l} \neq s_{0l+1},\, Q(as_{0l}) > 0 \,\}.$$

We show that if $L = l(s_0) - 1$ when $l(s_0) < \infty$, or L is sufficiently large with the property $Q(s_{0L+1}) < Q(s_{0L})$ when $l(s_0) = \infty$, this tree $\widetilde{\mathcal{T}}$ satisfies the assertions of the Lemma.

Now, using (5), the inequality (9) can be written as

$$\sum_{u \in \widetilde{\mathcal{T}},\, a \in A} N_n(u,a) \log \frac{N_n(u,a)}{N_n(u)} - \sum_{a \in A} N_n(s,a) \log \frac{N_n(s,a)}{N_n(s)} > c\,n.$$

Due to the ergodic theorem, $N_n(v,a)/n \to Q(va)$ for any string v, almost surely as $n \to \infty$. Hence, it is enough to show that

$$\sum_{u \in \widetilde{\mathcal{T}},\, a \in A} Q(ua) \log \frac{Q(ua)}{Q(u)} - \sum_{a \in A} Q(sa) \log \frac{Q(sa)}{Q(s)} > 0.$$

Jensen's inequality implies

$$Q(s) \sum_{u \in \widetilde{\mathcal{T}}} \frac{Q(u)}{Q(s)} \left(\frac{Q(ua)}{Q(u)} \log \frac{Q(ua)}{Q(u)} \right) \geq Q(sa) \log \frac{Q(sa)}{Q(s)}\,, \quad a \in A,$$

where the strict inequality holds for some $a \in A$, unless $Q(a|\,s) = Q(a|\,u)$ for each $a \in A$ and $u \in \widetilde{\mathcal{T}}$, in particular, for $u = s_{0L+1}$. In the case $l(s_0) < \infty$ we have $s_{0L+1} = s_0$, hence the last contingency is ruled out by $s \prec s_0 \in \mathcal{T}_0$ and the definition of context tree $\mathcal{T}_0$. In the case $l(s_0) = \infty$, if $Q(a|s)$ were equal to $Q(a|s_{0L+1})$ for each $a \in A$ and all L satisfying $Q(s_{0L+1}) < Q(s_{0L})$, letting $L \to \infty$ would give $Q(a|s) = Q(a|s_0)$, again contradicting $s \prec s_0 \in \mathcal{T}_0$.

The irreducibility of $\widetilde{\mathcal{T}}$ is obvious when $l(s_0) = \infty$, and in the case $l(s_0) < \infty$ it only requires checking that for $L = l(s_0) - 1$ there exists $a \in A$ with $as_{0L} \neq s_0$, $Q(as_{0L}) > 0$; this follows from $s_0 \in \mathcal{T}_0$ by Definition 2.1.

Moreover, we have $Q(u) > 0$ for each $u \in \widetilde{\mathcal{T}}$, and each $v \succeq s$ with $Q(v) > 0$ has a postfix in $\widetilde{\mathcal{T}}$ by construction. □

Lemma 3.2: For any irreducible tree $\mathcal{T}$ with $d(\mathcal{T}) \leq D(n)$, $D(n) = o(\log n)$, and $s \in \mathcal{T}$ that has a proper postfix $s_0 \in \mathcal{T}_0$

with $l(s_0) \le K$, there exists w satisfying $s \succ w \succeq s_0$ such that, for $\widetilde{\mathcal{T}} = \{\, u \in \mathcal{T} : u \succ w \,\}$ and arbitrary $\nu > 0$,

$$\sum_{u \in \widetilde{\mathcal{T}}} \log \widetilde{P}_{\mathrm{ML},\, u}(x_1^n) - \log \widetilde{P}_{\mathrm{ML},\, w}(x_1^n) < \nu \, |\widetilde{\mathcal{T}}| \log n, \tag{10}$$

holds simultaneously for all $\mathcal{T}$ and s as above, eventually almost surely as $n \to \infty$. Moreover, here $w = a_{-k}a_{-k+1}\dots a_{-1}$ can be chosen such that $a_{-k+1}\dots a_{-1}$ is a proper postfix of some $u \in \mathcal{T} \backslash \widetilde{\mathcal{T}}$.

Proof: Let $w = a_{-k}a_{-k+1}\dots a_{-1}$ be the longest postfix of s with $k < l(s)$ for which there exists a string in $\mathcal{T}$ not equal to w but having the postfix $a_{-k+1}\dots a_{-1}$. Then $\mathcal{T}_0 \in \mathcal{I}$ implies that $w \succeq s_0$, and hence $a_{-k+1}\dots a_{-1} \prec u$ for some $u \in \mathcal{T} \backslash \widetilde{\mathcal{T}}$.

Since

$$\prod_{a \in A} \left[\frac{N_n(w,a)}{N_n(w)} \right]^{N_n(w,a)} \ge \prod_{a \in A} Q(a|\, w)^{N_n(w,a)},$$

the left hand side of the claimed inequality can be bounded above by

$$\begin{aligned}
&\sum_{u \in \widetilde{\mathcal{T}},\, a \in A} N_n(u,a) \log \frac{N_n(u,a)}{N_n(u)} \; - \sum_{a \in A} N_n(w,a) \log Q(a|\, w) \\
&\overset{(i)}{=} \sum_{u \in \widetilde{\mathcal{T}},\, a \in A} N_n(u,a) \log \frac{N_n(u,a)}{N_n(u)} \\
&\qquad - \sum_{u \in \widetilde{\mathcal{T}},\, a \in A} N_n(u,a) \log Q(a|\, u) \\
&= \sum_{u \in \widetilde{\mathcal{T}}} N_n(u) \sum_{a \in A} \frac{N_n(u,a)}{N_n(u)} \log \frac{N_n(u,a)/N_n(u)}{Q(a|\, u)} \\
&= \sum_{u \in \widetilde{\mathcal{T}}} N_n(u)\; D\left(\frac{N_n(u,\cdot)}{N_n(u)} \,\Big\|\, Q(\,\cdot\,|u) \right)
\end{aligned}$$

Here (i) follows as $u \succ w \succeq s_0 \in \mathcal{T}_0$ implies $Q(a|\, u) = Q(a|\, w) = Q(a|\, s_0)$ by Definition 2.1. Using Lemmas 6.2 and 6.3 in the Appendix, this can be further bounded above, eventually almost surely simultaneously for all considered $\mathcal{T}$ and s, by

$$\begin{aligned}
&\sum_{u \in \widetilde{\mathcal{T}}} N_n(u) \frac{1}{q_{\min}} \sum_{a \in A} \left[\frac{N_n(u,a)}{N_n(u)} - Q(a|\, u) \right]^2 \\
&\quad < \sum_{u \in \widetilde{\mathcal{T}}} N_n(u) \frac{1}{q_{\min}} |A| \frac{\delta \log n}{N_n(u)} \le \frac{\delta \, |A|}{q_{\min}} |\widetilde{\mathcal{T}}| \log n,
\end{aligned}$$

where $q_{\min}$ is the minimum of the nonzero conditional probabilities $Q(a|\, s_0)$, $a \in A$, $s_0 \in \mathcal{T}_0$, $l(s_0) \le K$, and $\delta > 0$ is arbitrary small. □

Proof of Theorem 2.10

If $d(\mathcal{T}_0) < \infty$, the assumptions $\mathcal{T} \in \mathcal{F}_1(x_1^n, D(n))$, $D(n) = o(\log n)$, imply that $\mathcal{T} \in \mathcal{F}_{n^\alpha}(x_1^n, D(n))$ eventually almost surely as $n \to \infty$, by Lemma 6.1 in the Appendix. Hence it suffices to prove the second assertion of the Theorem.

The proof is similar to that of Theorem 2.6. It has to be checked that if $\mathcal{T}\big|_K \ne \mathcal{T}_0\big|_K$ for some $\mathcal{T} \in \mathcal{F}_{n^\alpha}(x_1^n, D(n)) \cap \mathcal{I}$ with $d(\mathcal{T}) \le D(n)$, then the modification $\mathcal{T}'$ of $\mathcal{T}$ defined by (7) or (8) satisfies $\mathcal{T}' \in \mathcal{F}_{n^\alpha}(x_1^n, D(n)) \cap \mathcal{I}$ and

$$\mathrm{KT}_{\mathcal{T}}(x_1^n) > \mathrm{KT}_{\mathcal{T}'}(x_1^n), \tag{11}$$

simultaneously for all considered trees $\mathcal{T}$, eventually almost surely as $n \to \infty$.

Let $\widetilde{P}_{\mathrm{KT},\, s}(x_1^n)$ denote

$$\frac{\prod_{a: N_n(s,a) \ge 1} \left[\left(N_n(s,a) - \frac{1}{2}\right) \left(N_n(s,a) - \frac{3}{2}\right) \cdots \left(\frac{1}{2}\right) \right]}{\left(N_n(s) - 1 + \frac{|A|}{2}\right) \left(N_n(s) - 2 + \frac{|A|}{2}\right) \cdots \left(\frac{|A|}{2}\right)} \tag{12}$$

if $N_n(s) \ge 1$, and 1 if $N_n(s) = 0$. Then the KT probability $P_{\mathrm{KT},\mathcal{T}}(x_1^n)$ in Definition 2.8 factorizes as

$$P_{\mathrm{KT},\mathcal{T}}(x_1^n) = \frac{1}{|A|^{D(n)}} \prod_{s \in \mathcal{T}} \widetilde{P}_{\mathrm{KT},\, s}(x_1^n). \tag{13}$$

It follows that (11) is equivalent to

$$\sum_{s \in \mathcal{T}} \log \widetilde{P}_{\mathrm{KT},\, s}(x_1^n) - \sum_{s' \in \mathcal{T}'} \log \widetilde{P}_{\mathrm{KT},\, s'}(x_1^n) < 0. \tag{14}$$

Substituting $\mathcal{T}'$ given by (7) or (8), this reduces to

$$\log \widetilde{P}_{\mathrm{KT},\, s}(x_1^n) - \sum_{u \in \widetilde{\mathcal{T}}} \log \widetilde{P}_{\mathrm{KT},\, u}(x_1^n) < 0 \tag{15}$$

in case (a), where $\widetilde{\mathcal{T}}$ is as in Lemma 3.1, respectively to

$$\sum_{u \in \widetilde{\mathcal{T}}} \log \widetilde{P}_{\mathrm{KT},\, u}(x_1^n) - \log \widetilde{P}_{\mathrm{ML},\, w}(x_1^n) < 0 \tag{16}$$

in case (b), where $\widetilde{\mathcal{T}}$ and w are as in Lemma 3.2.

To deduce (15) and (16) from Lemmas 3.1 and 3.2 (in the required eventually almost sure sense), we use the standard bound (see, e.g., [4] eq. (2.12))

$$\left| \log \widetilde{P}_{\mathrm{KT},\, u}(x_1^n) - \sum_{a \in A} N_n(u,a) \log \frac{N_n(u,a)}{N_n(u)} + \frac{|A| - 1}{2} \log N_n(u) \right| < C$$

for any string u with $N_n(u) \ge 1$, where C is a constant depending only on the alphabet size $|A|$ with the notation (5). The last bound can be equivalently written as

$$\left| \log \widetilde{P}_{\mathrm{KT},\, u}(x_1^n) - \log \widetilde{P}_{\mathrm{ML},\, u}(x_1^n) + \frac{|A| - 1}{2} \log N_n(u) \right| < C\,. \tag{17}$$

The claim (15) immediately follows from (9) by (17) and the trivial bounds $0 \le \log N_n(u) \le \log n$.

Also, (17) gives for the left hand side of (16) the upper bound

$$\begin{aligned}
&\sum_{u \in \widetilde{\mathcal{T}}} \left(\log \widetilde{P}_{\mathrm{ML},\, u}(x_1^n) - \frac{|A| - 1}{2} \log N_n(u) + C \right) \\
&\qquad - \left(\log \widetilde{P}_{\mathrm{ML},\, w}(x_1^n) - \frac{|A| - 1}{2} \log N_n(w) - C \right).
\end{aligned}$$

For $\widetilde{\mathcal{T}}$ in Lemma 3.2, the assumption $\mathcal{T} \in \mathcal{F}_{n^\alpha}(x_1^n, D(n))$ implies $N_n(u) \geq n^\alpha$ for each $u \in \widetilde{\mathcal{T}}$, and since the sum of $N_n(u)$ for $u \in \widetilde{\mathcal{T}}$ is equal to $N_n(w)$, we have $N_n(u) \geq N_n(w)/|\widetilde{\mathcal{T}}|$ for at least one $u \in \widetilde{\mathcal{T}}$. Using these facts in the last bound, and denoting the left hand side of (10) in Lemma 3.2 by Δ, it follows that the left hand side of (16) is bounded above by

$$\Delta - (|\widetilde{\mathcal{T}}|-1)\frac{|A|-1}{2}\,\alpha\log n \;-\; \frac{|A|-1}{2}\log\frac{N_n(w)}{|\widetilde{\mathcal{T}}|} + \frac{|A|-1}{2}\log N_n(w) + (|\widetilde{\mathcal{T}}|+1)\,C.$$

By Lemma 3.2, here $\Delta < \nu\,|\widetilde{\mathcal{T}}|\log n$ eventually almost surely as $n \to \infty$, for arbitrary $\nu > 0$, simultaneously for all considered $\mathcal{T}$ and s, and thus the claim (16) follows. □

IV. Computation of the KT and BIC estimators

The estimators $\widehat{\mathcal{T}}_{\mathrm{BIC}}(x_1^n)$ and $\widehat{\mathcal{T}}_{\mathrm{KT}}(x_1^n)$ in Theorems 2.6 and 2.10, the latter for the case $d(\mathcal{T}_0) < \infty$, can be represented as

$$\widehat{\mathcal{T}}(x_1^n) = \arg\max_{\mathcal{T}\in\mathcal{F}_1(x_1^n,D(n))\cap\mathcal{I}} \prod_{s\in\mathcal{T}} \widetilde{P}_s(x_1^n), \tag{18}$$

where $\widetilde{P}_s(x_1^n) = \widetilde{P}_{\mathrm{KT},\,s}(x_1^n)$ in the KT case, and $\widetilde{P}_s(x_1^n) = n^{-\frac{|A|-1}{2}}\widetilde{P}_{\mathrm{ML},\,s}(x_1^n)$ in the BIC case, see (13), Definition 2.8, (4), Definition 2.4.

These facts admit a joint treatment of the computations of the BIC and KT estimators, via an extension of the CTM algorithm of [15], [17] developed for the KT case. This algorithm has the following construction.

Consider the full tree A^D, where $D = D(n) = o(\log n)$, and let $\mathcal{S}_D$ denote the set of its nodes, i.e., the set of all strings of length at most D. Based on the sample x_1^n we assign to each node a value and a binary indicator. This assignment is recursive, that is, the value and the indicator assigned to a node are calculated from the values assigned to the children of this node. The desired estimator will be the subtree determined by the indicators as specified below.

In the sequel, $\widetilde{P}_s(x_1^n)$ denotes either possibility in the first passage of this section.

Definition 4.1: Given a sample x_1^n, to each string $s \in \mathcal{S}_D$ with $N_n(s) \geq 1$, $D = D(n)$ we assign recursively, starting from the leaves of the full tree A^D, the value

$$V_s^D(x_1^n) = \begin{cases} \max\left\{ \widetilde{P}_s(x_1^n)\,,\; \displaystyle\prod_{a\in A:\,N_n(as)\geq 1} V_{as}^D(x_1^n) \right\} & \\ \qquad\qquad\qquad\qquad \text{if } \; 0 \leq l(s) < D, & \\ \widetilde{P}_s(x_1^n) \qquad\qquad\qquad \text{if } \; l(s) = D, & \end{cases}$$

and the indicator

$$\chi_s^D(x_1^n) = \begin{cases} 1 \;\text{ if }\; 0 \leq l(s) < D \text{ and} \\ \qquad \prod_{a\in A:\,N_n(as)\geq 1} V_{as}^D(x_1^n) > \widetilde{P}_s(x_1^n), \\ 0 \;\text{ if }\; 0 \leq l(s) < D \text{ and} \\ \qquad \prod_{a\in A:\,N_n(as)\geq 1} V_{as}^D(x_1^n) \leq \widetilde{P}_s(x_1^n), \\ 0 \;\text{ if }\; l(s) = D. \end{cases}$$

Using these indicators, we assign to each $s \in \mathcal{S}_D$, $D = D(n)$ a maximizing tree $\mathcal{T}_s^D(x_1^n)$ consisting of strings $u \succeq s$. The term “maximizing” is justified by Lemma 4.4 below.

Definition 4.2: Given $s \in \mathcal{S}_D$, let $\mathcal{T}_s^D(x_1^n)$ equal to

$$\left\{\, u \in \mathcal{S}_D : \chi_u^D(x_1^n) = 0,\ \chi_v^D(x_1^n) = 1 \text{ for all } s \preceq v \prec u \,\right\}$$

if $\chi_s^D(x_1^n) = 1$, and to $\{s\}$ if $\chi_s^D(x_1^n) = 0$.

The maximizing tree $\mathcal{T}_s^D(x_1^n)$ is irreducible unless it equals $\{s\}$. Indeed, if $N_n(s) = N_n(as)$ holds for a string $s \in \mathcal{S}_{D-1}$ and a letter a (and thus $N_n(a_1 s) = 0$ for all $a_1 \neq a$, $a_1 \in A$) then $\chi_s^D(x_1^n) = 1$ implies $\chi_{as}^D(x_1^n) = 1$.

Proposition 4.3: The context tree estimator $\widehat{\mathcal{T}}(x_1^n)$ in (18) equals the maximizing tree assigned to the root, that is,

$$\widehat{\mathcal{T}}(x_1^n) = \mathcal{T}_\varnothing^D(x_1^n).$$

Proof: The claimed equality follows from the next lemma by substituting $s = \varnothing$, on account of (18) and the fact that $\mathcal{T}_\varnothing^D(x_1^n)$ is irreducible. □

For any $s \in \mathcal{S}_D$ with $N_n(s) \geq 1$, define $\mathcal{F}_1(x_1^n|s)$ as the family of all trees $\mathcal{T}$ of depth $d(\mathcal{T}) \leq D$, consisting of strings $u \succeq s$ with $N_n(u) \geq 1$, such that each $s' \succ s$ with $N_n(s') \geq 1$ is either a postfix of some $u \in \mathcal{T}$ or has a postfix in $\mathcal{T}$.

Lemma 4.4: For any $s \in \mathcal{S}_D$ with $N_n(s) \geq 1$

$$V_s^D(x_1^n) = \max_{\mathcal{T}\in\mathcal{F}_1(x_1^n|s)} \prod_{u\in\mathcal{T}} \widetilde{P}_u(x_1^n) = \prod_{u\in\mathcal{T}_s^D(x_1^n)} \widetilde{P}_u(x_1^n).$$

Proof: By induction on the length of the string s, similarly to [15]. For $l(s) = D$ the statement is obvious.

Supposing the assertion holds for all strings of length d, we have for any s with $l(s) = d-1$

$$\begin{aligned} \prod_{a\in A:\,N_n(as)\geq 1} & V_{as}^D(x_1^n) \\ &= \prod_{a\in A:\,N_n(as)\geq 1}\left(\max_{\mathcal{T}_a\in\mathcal{F}_1(x_1^n|as)} \prod_{u\in\mathcal{T}_a} \widetilde{P}_u(x_1^n)\right) \\ &= \max_{\mathcal{T}\in\mathcal{F}_1(x_1^n|s):\,d(\mathcal{T})\geq 1} \prod_{u\in\mathcal{T}} \widetilde{P}_u(x_1^n). \end{aligned}$$

Here the second equality holds since any family of trees $\mathcal{T}_a$, $a \in A$, $N_n(as) \geq 1$, satisfying the indicated constraints, uniquely corresponds to a tree $\mathcal{T} \in \mathcal{F}_1(x_1^n|s)$ with $d(\mathcal{T}) \geq 1$ via $\mathcal{T} = \cup_a \mathcal{T}_a$.

It follows by Definition 4.1 that

$$\begin{aligned} V_s^D(x_1^n) &= \max\left\{ \widetilde{P}_s(x_1^n)\,,\; \max_{\mathcal{T}\in\mathcal{F}_1(x_1^n|s):\,d(\mathcal{T})\geq 1} \prod_{u\in\mathcal{T}} \widetilde{P}_u(x_1^n) \right\} \\ &= \max_{\mathcal{T}\in\mathcal{F}_1(x_1^n|s)} \prod_{u\in\mathcal{T}} \widetilde{P}_u(x_1^n), \end{aligned}$$

proving the first equality in the Lemma. The second equality also follows from the last identity, by the induction hypothesis and Definitions 4.1 and 4.2. □

Remark 4.5: For the KT case, Lemma 4.4 above with the condition $\mathcal{T} \in \mathcal{F}_1(x_1^n|s)$ replaced by the condition that $\mathcal{T}$ is complete, is a result of [15], [17] (with the minor difference that the trees there also had “costs”), and the above proof is similar to theirs.

The KT estimator in Theorem 2.10 for the general case can still be represented as in (18), with $\widetilde{P}_s(x_1^n) = \widetilde{P}_{\text{KT},s}(x_1^n)$, the only difference is that $\mathcal{F}_1(x_1^n, D(n))$ in (18) is replaced by $\mathcal{F}_r(x_1^n, D(n))$ with $r = n^\alpha$. For this case, Definition 4.1 is modified by setting $V_s^D(x_1^n) = 0$ for all $s \in \mathcal{S}_D$ with $N_n(s) < r$. The definition remains unchanged for $s \in \mathcal{S}_D$ with $N_n(s) \geq r$, but of course the values $V_s^D(x_1^n)$ may change also for these strings s. In particular, if $N_n(s) \geq r$ but $1 \leq N_n(as) < r$ for some $a \in A$, the modified definition gives $V_s^D(x_1^n) = \widetilde{P}_s(x_1^n)$ and $\chi_s^D(x_1^n) = 0$.

Adopting this modified Definition 4.1, it is easy to see that Proposition 4.3 still holds, that is, the maximizing tree of Definition 4.2 assigned to the root equals the KT estimator in Theorem 2.10 for the general case.

Next we show that the computation of the estimators in Theorems 2.6 and 2.10 via the above method has the asserted complexity in the off-line case.

Proof of Theorem 2.13

Since $D(n) = o(\log n)$, we may write $D(n) = \varepsilon_n \log n$, where $\varepsilon_n \to 0$.

For each string $s \in \mathcal{S}_D$, $D = D(n) = \varepsilon_n \log n$, the counts $N_n(s,a)$, $a \in A$, as well as $\widetilde{P}_s(x_1^n)$, $V_s^D(x_1^n)$, $\chi_s^D(x_1^n)$ are stored. The number of stored data is proportional to the cardinality of $\mathcal{S}_D$, which is

$$\sum_{j=0}^{D} |A|^j = \frac{|A|^{D+1} - 1}{|A| - 1} \leq 2|A|^D = O(n^\varepsilon). \tag{19}$$

To get the indicators $\chi_s^D(x_1^n)$, $s \in \mathcal{S}_D$ which give rise to the trees $\mathcal{T}_s^D(x_1^n)$ according to Definition 4.2, first we need the counts $N_n(s,a)$, $s \in \mathcal{S}_D$, $a \in A$.

The counts $N_n(s,a)$ for $s \in A^D$, $a \in A$ can be determined successively processing the sample x_1^n from position $j = D(n)$ to $j = n$, and at instance j incrementing the count $N_n\left(x_{j-D(n)}^{j-1}, x_j\right)$ by 1 (the starting values of all counts being 0). This is $O(n)$ calculations. The other counts $N_n(s,a)$, $s \in \mathcal{S}_{D-1}$, $a \in A$ can be determined recursively, as $N_n(s,a) = \sum_{b \in A} N_n(bs, a)$. This is $|A|\,|\mathcal{S}_{D-1}| = o(n)$ calculations.

Then, from these counts the values $\widetilde{P}_s(x_1^n)$ are determined by $O(n)$ multiplications. The calculation of the values $V_s^D(x_1^n)$ and $\chi_s^D(x_1^n)$ requires calculations proportional to the cardinality of $\mathcal{S}_D$, which is less than $2|A|^D = o(n)$. □

Consider next the on-line versions of the estimators, with the modifications described in the passage before Theorem 2.14. In the BIC case, the representation (18) holds with $\widetilde{P}_s(x_1^n) = e^{-\frac{|A|-1}{2}\lfloor \log_{|A|} n \rfloor \log|A|}\, \widetilde{P}_{\text{ML},s}(x_1^n)$. In the KT case, the same estimator is used as for the off-line computation, when $d(\mathcal{T}_0) < \infty$. The on-line version of the KT estimator for the general case is analogous to the off-line version, with $r = e^{\alpha\lfloor \log_{|A|} n \rfloor}$ instead of $r = n^\alpha$.

Finally, we show that these algorithms have the asserted computational complexity in the on-line case.

Proof of Theorem 2.14

The calculations required by the algorithm in Definition 4.1 can be performed recursively in the sample size n.

Suppose at instant i, for each string $s \in \mathcal{S}_{D(i)}$, the counts $N_i(s,a)$, $a \in A$, as well as $\widetilde{P}_s(x_1^i)$, $V_s^D(x_1^i)$, $\chi_s^D(x_1^i)$ are stored, where $D = D(i)$. The number of stored data is proportional to the cardinality of $\mathcal{S}_{D(i)}$, which is $O(i^\varepsilon)$, see (19).

Consider first those instances i when the sample size increases from $i-1$ to i but $\lfloor \log_{|A|}(i-1) \rfloor = \lfloor \log_{|A|} i \rfloor$, and the depth does not change, $D(i) = D(i-1)$. If $\widetilde{P}_s(x_1^{i-1})$ at a node s is known, $\widetilde{P}_s(x_1^i)$ can be calculated using, for the KT case, that

$$\widetilde{P}_{\text{KT},s}(x_1^i) = \frac{N_i(s,x_i) + 1/2}{N_i(s) + |A|/2}\ \widetilde{P}_{\text{KT},s}(x_1^{i-1}),$$

and in the BIC case that in the expression of $\widetilde{P}_{\text{ML},s}(x_1^{i-1})$ only the counts $N_i(s,x_i)$ and $N_i(s)$ were incremented to obtain $\widetilde{P}_{\text{ML},s}(x_1^i)$. From $\widetilde{P}_s(x_1^i)$ the values $V_s^D(x_1^i)$ and $\chi_s^D(x_1^i)$ can be computed in constant number of steps. These values are different for x_1^{i-1} and x_1^i only when s is a postfix of x_1^{i-1}, hence updating is needed at $D(i)$ nodes only. Thus the number of required computations is proportional to $D(i)$.

Consider those instances i when the sample size increases from $i-1$ to i such that $\lfloor \log_{|A|} i \rfloor = \lfloor \log_{|A|}(i-1) \rfloor + 1$ but the depth does not change. The additional task compared to the previous case is that recalculation of $V_s^D(x_1^i)$ and $\chi_s^D(x_1^i)$ is needed for all nodes $s \in \mathcal{S}_{D(i)}$, which requires calculations proportional to the cardinality of $\mathcal{S}_{D(i)}$.

Consider next those instances i when the depth increases, $D(i) = D(i-1)+1$. In this case we have three tasks. We have to update $\widetilde{P}_s(x_1^{i-1})$ at those nodes s that already existed at instance $i-1$, namely where $l(s) < D(i)$. In addition, we have to calculate them for the new terminal nodes s, $l(s) = D(i)$, and recalculate $V_s^D(x_1^i)$ and $\chi_s^D(x_1^i)$ at all nodes s of the new full tree. The former needs $O(i)$ calculations. Indeed, the counts $N_i(s,a)$, $l(s) = D(i)$, can be determined successively processing the sample x_1^i from position $j = D(i)$ to $j = i$, and at instance j incrementing the count $N_i\left(x_{j-D(i)}^{j-1}, x_j\right)$ by 1 (the starting values of all counts being 0), and from these counts $\widetilde{P}_s(x_1^i)$ are determined by $O(i)$ multiplications. The recalculation of the values $V_{D,s}(x_1^i)$ and $\chi_{D,s}(x_1^i)$ requires calculations proportional to the cardinality of $\mathcal{S}_{D(i)}$.

Finally, the total number of computations performed on a sample x_1^n is bounded as follows. The number of computations needed for the updating at all instances $i \leq n$ is proportional to

$$\sum_{i=1}^{n} D(i) = \sum_{i=1}^{n} \lfloor \varepsilon_i \log i \rfloor = o(n \log n).$$

The number of computations to recalculate $V_{D,s}$, $\chi_{D,s}$ for all nodes in the full tree $A^{D(i)}$ at the instances when $\lfloor \log_{|A|} i \rfloor$ increases is of order

$$\sum_{D=0}^{\lfloor \log_{|A|} n \rfloor} 2|A|^D = O\left(|A|^{\log_{|A|} n}\right) = O(n).$$

The number of computations to calculate $\widetilde{P}_s$ for the new terminal nodes at the instances when $D(i)$ increases is proportional to

$$\sum_{D=0}^{\lfloor \varepsilon_n \log n \rfloor} \min\{\, i : D \le \varepsilon_i \log i \,\}$$
$$= \sum_{D=0}^{\lfloor \varepsilon_n \log n \rfloor} \min\{\, i : e^{D/\varepsilon_i} \le i \,\} \le \sum_{D=0}^{\lfloor \varepsilon_n \log n \rfloor} e^{D/\varepsilon_n} + 1$$
$$\le O\left(e^{\frac{1}{\varepsilon_n} \varepsilon_n \log n} \right) + \varepsilon_n \log n = O(n).$$

The number of computations to recalculate $V_{D,\,s},\ \chi_{D,\,s}$ for all nodes in the full tree $A^{D(i)}$ at the instances when $D(i)$ increases is of order

$$\sum_{D=0}^{\lfloor \varepsilon_n \log n \rfloor} 2|A|^D = O\left(|A|^{\varepsilon_n \log n} \right) = o(n).$$

□

V. Discussion

We have proved the strong consistency of the BIC estimator and the KT version of MDL estimator of the context tree of any (stationary ergodic) process, when the depth of the hypothetical context trees is allowed to grow with the sample size n as $o(\log n)$. This context tree may have infinite depth, and it is not necessarily complete. These consistency results are generalizations of similar results for estimation of the order of Markov chains [4], [5].

We have considered processes with time domains equal to the set of all integers, but as long as stationarity and ergodicity are insisted upon, any process with one-sided time domain $\mathbb{N}$ can be obtained by restricting the time domain of a process of the former kind. When dealing with Markov chain order estimation in the one-sided case, dropping the stationarity assumption causes no additional difficulty, see [4]. For context tree estimation of tree sources, non-stationarity may cause technical problems in dealing with transient phenomena, but does not appear to significantly change the picture, see [8].

While the BIC Markov order estimator is consistent without any bound on the hypothetical orders [4], it remains open whether the BIC context tree estimator remains consistent when dropping the depth bound $o(\log n)$, or replacing it by a bound $c \log n$. For the KT context tree estimator it also remains open whether the depth bound could be increased; it certainly can not be dropped or replaced by a large constant times $\log n$, since then consistency fails even for Markov order estimation [4].

With KT, we have considered two kinds of estimators, the second kind admitting only "r-frequent" hypothetical trees with $r = n^\alpha$. The latter conforms with the intuitive idea that the estimation should be based on those strings that "frequently" appeared in the sample, see [3]. When the context tree has finite depth, the restriction to n^α-frequent hypothetical trees was not necessary since all feasible trees (of depth $D(n) = o(\log n)$) satisfied it automatically, eventually almost surely. It remains open whether the mentioned restriction is necessary for consistency when the context tree has infinite depth.

A consequence of the consistency theorems is that when a process is not a Markov chain of any (finite) order, the estimated order, produced by either of the BIC or KT estimators, tends to infinity almost surely.

We have also shown that the BIC and KT context tree estimators can be computed in linear time, via suitable modifications of the CTM method [15], [17]. An on-line procedure was also considered that calculates the estimators for all sample sizes $i \le n$ in $o(n \log n)$ time. This result may be useful, for example, to implement context tree estimation with a stopping rule based on "stabilizing" of the estimator.

The NML version of MDL was not considered for the context tree estimation problem (though it was for Markov order estimation in [5]), because the structure of the NML criterion, unlike BIC and KT, appears unsuitable for CTM implementation.

Finally we note that in the definition of BIC (Definition 2.4), the factor $(|A|-1)|\mathcal{T}|/2$ in the penalty term could be replaced by $c|\mathcal{T}|$, with any positive constant c, without affecting our results. The question of what other penalty terms might be appropriate is not in the scope of this paper.

VI. Appendix

Lemma 6.1: Given a process Q with context tree of finite depth, for any $0 < \alpha < 1$ there exists $\kappa > 0$ such that, eventually almost surely as $n \to \infty$,

$$N_n(s) \ge n^\alpha,$$

simultaneously for all strings s with $Q(s) > 0$, $l(s) \le \kappa \log n$.

Proof: This bound has been used in [5], proof of Theorem 5. It is a consequence of the typicality theorem in [4], see also [5], remark after Th. 1. Indeed, the latter implies the existence of $\kappa > 0$ such that $N_n(s)/n \ge Q(s)/2$ simultaneously for all s with $l(s) < \kappa \log n$, eventually almost surely as $n \to \infty$. The assertion of the lemma follows, since $Q(s)$, when positive, is bounded below by $\xi^{l(s)}$ for a constant $\xi > 0$. □

Lemma 6.2: Given a process Q, to any $\delta > 0$ there exists $\kappa > 0$ such that, eventually almost surely as $n \to \infty$,

$$\left| \frac{N_n(s,a)}{N_n(s)} - Q(a|\, s) \right| < \sqrt{\frac{\delta \log n}{N_n(s)}}$$

simultaneously for all strings s with $l(s) \le \kappa \log n$ and $N_n(s) \ge 1$ which have a postfix in the context tree of Q.

Proof: By Theorem 2 of [5], for $\xi > (\log |A|)/2$ there exist $\eta > 0$ and $c > 0$ such that, eventually almost surely as $n \to \infty$,

$$\left| \frac{N_n(s,a)}{N_n(s)} - Q(a|\, s) \right| < \sqrt{\frac{\max\{\, \xi\, l(s),\, \eta \log \log N_n(s) \,\}}{N_n(s)}} \tag{20}$$

simultaneously for all strings s with $N_n(s) \ge c\, l(s)$ which have a postfix in the context tree of Q. While Theorem 2 of [5] is stated for Markov processes only, the proof relies upon the martingale property of the sequence Z_n of [5], eq. (10), and $Z_n = N_n(s,a) - Q(a|s)\,N_{n-1}(s)$ defines a martingale whenever s has a postfix in the context tree of the process Q. Thus the mentioned proof applies literally.

Then the choice $\kappa = \delta/\max\{\xi, c/4\}$ is suitable for Lemma 6.2. Indeed, if $N_n(s) \geq c\,l(s)$, the bound (20) holds and gives the assertion, while in the opposite case $N_n(s) < c\,l(s) \leq c\,\kappa \log n$ we have $\sqrt{(\delta \log n)/N_n(s)} \geq \sqrt{\delta/(c\,\kappa)} \geq 2$ and the assertion holds trivially. □

Lemma 6.3: For probability distributions P_1 and P_2 on A

$$D(P_1\|P_2) \leq \sum_{a\in A} \frac{(P_1(a) - P_2(a))^2}{P_2(a)}.$$

Proof:

$$D(P_1\|P_2) = \sum_{a\in A} P_1(a) \log \frac{P_1(a)}{P_2(a)}$$
$$\leq \sum_{a\in A} P_1(a) \left(\frac{P_1(a)}{P_2(a)} - 1 \right) = \sum_{a\in A} \frac{(P_1(a) - P_2(a))^2}{P_2(a)}\,.$$

□

Acknowledgement

The authors are indebted to László Györfi for pointing out Lemma 6.3, and to anonymous reviewers for suggestions to improve presentation.

References

[1] D. Baron and Y. Bresler, "An $O(N)$ semipredictive universal encoder via the BWT," *IEEE Trans. Inform. Theory,* vol. IT-50, pp. 928–937, May 2004.

[2] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Inform. Theory,* vol. IT-44, pp. 2743–2760, Oct. 1998.

[3] P. Bühlmann and A. J. Wyner, "Variable length Markov chains," *Ann. Statist.,* vol. 27, pp. 480–513, 1999.

[4] I. Csiszár and P. C. Shields, "The consistency of the BIC Markov order estimator," *Ann. Statist.,* vol. 28, pp. 1601–1619, 2000.

[5] I. Csiszár, "Large-scale typicality of Markov sample paths and consistency of MDL order estimators," *IEEE Trans. Inform. Theory,* vol. IT-48, pp. 1616–1628, June 2002.

[6] L. Finesso, "Estimation of the order of a finite Markov chain," In *Recent Advances in Mathematical Theory of Systems, Control, Networks and Signal Processing, I* (H. Kimura and S. Kodama, eds.) pp. 643–645. Tokyo: Mita Press, 1992.

[7] R. E. Krichevsky and V. K. Trofimov, "The performance of universal encoding," *IEEE Trans. Inform. Theory,* vol. IT-27, pp. 199–207, Mar. 1981.

[8] A. Martín, G. Seroussi, and M. J. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inform. Theory,* vol. IT-50, pp. 1442–1468, July 2004.

[9] R. Nohre, "Some topics in descriptive complexity," EE Dept., Linköping University, Ph.D. Thesis, 1994.

[10] J. Rissanen, "A universal data compression system," *IEEE Trans. Inform. Theory,* vol. IT-29, pp. 656–664, Sept. 1983.

[11] J. Rissanen, *Stochastic Complexity in Statistical Inquiry.* Singapore: World Scientific, 1989.

[12] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.,* vol. 6, pp. 461–464, 1978.

[13] M. J. Weinberger, A. Lempel, and J. Ziv, "A sequentional algorithm for the universal coding of finite memory sources," *IEEE Trans. Inform. Theory,* vol. IT-38, pp. 1002–1014, May 1992.

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

[15] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," EE Dept., Eindhoven University, Tech. Rep., 1993. (An earlier unabridged version of [16]).

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

[17] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context-tree maximizing," in *Proc. 2000 Conf. Information Sciences and Systems,* Princeton, NJ, pp. TP6-7–TP6-12, Mar. 2000.

[18] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inform. Theory,* vol. IT-44, pp. 792–798, Mar. 1998.