Sequential Prediction and Ranking in Universal Context Modeling and Data Compression

Marcelo J. Weinberger, Gadiel Seroussi
Computer Systems Laboratory

HPL-94-111 (R.1)
January, 1997

HPL-94-111 (R.1): Sequential Prediction and Ranking in Universal Context
Modeling and Data Compression

Sequential Prediction and Ranking in Universal Context Modeling and Data Compression

Marcelo J. Weinberger, Gadiel Seroussi
Computer Systems Laboratory

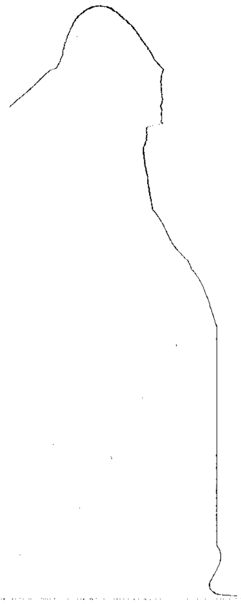
HPL-94-111 (R.1)
January, 1997

universal coding,
Context algorithm,
prediction, ranking,
image compression

Prediction is one of the oldest and most successful tools in practical data compression. It is particularly useful in applications like image compression, where the data samples represent a continuously varying physical process. The usual explanation for the beneficial effect of prediction is that it decorrelates the data, allowing the use of simple encoders for the sequence of prediction errors. It has been noted, though, both in theory and in practice, that prediction alone cannot achieve total decorrelation in general, even in the case of universal prediction. In fact, most state-of-the-art image compression schemes use prediction followed by some form of context modeling. This, however, might seem redundant at first, as the context information used for prediction is also available for building the compression model, and a universal modeler will eventually learn the "predictive" patterns of the source. Thus, the following questions arise: Why use two different modeling tools based on the same contextual information, and how do these tools interact?

In this paper, we provide answers to these questions, and we investigate the use of prediction as a means of reducing the model cost in universal lossless data compression. We provide a formal justification to the combination of this tool with a universal code based on context modeling, by showing that a combined scheme may result in faster convergence rate to the source entropy. In deriving the main result, we develop the concept of sequential ranking, which can be seen as a generalization of sequential prediction, and we study its combinatorial and probabilistic properties.

Published in and presented at the *IEEE International Symposium on Information Theory,* Trondheim, Norway, June 27-July 1, 1994.



1 Introduction

Prediction is one of the oldest and most successful tools in the data compression practitioner's toolbox. It is particularly useful in situations where the data (e.g., a digital image) originates from a natural physical process (e.g., sensed light), and the data samples (e.g., real numbers) represent a continuously varying physical magnitude (e.g., brightness). In these cases, the value of the next sample can often be accurately predicted using a simple function (e.g., a linear combination) of previously observed neighboring samples [1]. The usual interpretation of the beneficial effect of prediction is that it *decorrelates* the data samples, thus allowing the use of simple encoders for the sequence of *prediction errors*, from which the original data set can be reconstructed. In fact, in the case of natural photographic images, very simple DPCM schemes based on low order linear prediction followed by a zero-order Huffman code go a long stretch of the way towards the best practical compression ratios.

It has been noted, though, both in theory [2, 3] and in practice (starting with [4]), that prediction alone cannot achieve total decorrelation in general, even in the case of *universal prediction* [5, 6, 7, 2, 8]. In fact, most state-of-the-art lossless image compression schemes (e.g., [9, 10, 11, 12]) use prediction followed by some form of context modeling (i.e., encoding based on higher order conditional probability models), with the latter stage exploiting dependencies left in the data after the prediction stage. In particular, prediction can be followed by a *universal* encoder for a broad class of context models [13, 14, 15, 16, 10]. In this case, however, the "decorrelation" benefit of prediction, which is obvious when followed by a zero-order coder, becomes less clear, since the same contextual information that is used to predict is also available for building the compression model, and the latter will eventually learn the "predictive" patterns of the data. Yet, practical experience clearly indicates that prediction is still beneficial when used in conjunction with higher order context models (as many practitioners who have attempted to use one of the various universal schemes directly on image data and got disappointed will attest). Thus, the following questions arise, and provide the motivation for this work: Why use two different modeling tools based on the same contextual information, and how do these tools interact?

In this paper, we address these questions, and we study the interaction between universal prediction and universal coding, bridging what we perceive as a gap in the formal treatment of this successful combination, each of whose components has been thoroughly studied. We formalize and quantify the effect of prediction in the framework of *finite-memory* (or *tree*) sources over finite alphabets [13, 14, 15]. A tree source consists of an underlying α-ary context tree structure, where α is the size of the source alphabet, and a set of conditional probability distributions on the alphabet, one associated with each leaf of the tree.[1] These sources provide a reduced parametrization of Markov models (achieved through merging sibling states with identical conditional probability distributions), and can be modeled by efficient algorithms [15, 16]. In our main result, we show that prediction can provide an algorithmic way to reduce the *model cost* (i.e., the intrinsic penalty in code length due to the number of free parameters modeled) incurred by a universal modeler, thus allowing for a faster convergence to the model entropy as per Rissanen's lower bound [17, Theorem 1]. Notice that model size reduction is an appropriate refinement of the notion of decorrelation, with total decorrelation corresponding to the extreme case where the model is reduced to one node (i.e., a memoryless source).

Our starting point is the observation that the effect of prediction is often that of merging probability distributions that are similar but centered at different values. This is typical, for example, of gray-scale images, where sample values often obey a Laplacian-like (double sided exponential) probability distribution whose center is context-dependent [18, 1]. By predicting a deterministic, context-dependent value $\hat{x}$ for the "next symbol" x, and considering the (context-)conditional probability distribution of the prediction error

[1] All technical terms and concepts discussed informally in this introduction will be precisely defined in subsequent sections.

$e = x - \hat{x}$ rather than that of x itself, we allow for similar probability distributions on e, which may now be all centered at zero, to merge in situations when the original distributions on x would not do so. In the case where x is drawn from a finite alphabet of cardinality α (e.g., the set of numbers $\{0, 1, \ldots, \alpha - 1\}$), and given a prediction value $\hat{x}$ from the same alphabet, the prediction error e can take on only α different values, and one can assume without affecting the losslessness of the prediction stage that e is computed over the integers modulo α. Thus, prediction effects the alphabet permutation[2] $x \to (x - \hat{x}) \bmod \alpha$. In our framework, we will consider arbitrary permutations of the alphabet, thus freeing the analysis from any assumption of "smoothness" or even a metric on the input samples. The notion of two probability distributions that can be made similar by "centering" is replaced by that of two distributions that can be made similar by an arbitrary permutation of the alphabet. In this framework, the analog of a "predicted value" $\hat{x}$ that centers the distribution is a permutation that ranks the alphabet symbols in decreasing probability order (with appropriate provisions for ties). Thus, instead of just predicting what the most likely next symbol is, we will predict also what the second most likely symbol is, what the third most likely symbol is, and so on.[3] A *permutation-minimal tree* is defined as a tree source representation in which we have merged, to the maximum extent possible, complete sets of sibling leaves that have the same *sorted* probability vector (i.e., the same conditional probability distribution up to a permutation of the symbol "labels"). The permutation-minimal tree representation of a source may be smaller than its conventional minimal tree representation, thus allowing for potential savings in model cost and faster convergence to the source entropy.

To obtain the ranking permutations that allow for the construction of a permutation-minimal tree, we will define *universal sequential ranking* as a generalization of universal sequential prediction, to be used in conjunction with the universal coder. Universal ranking, however, also requires a model structure with an associated model cost (as does universal prediction), which should be weighted against the potential savings in compression model cost. An indication that this trade-off might be favorable is given by a comparison between the compressibility bound in [17] and the predictability bound in [8], which shows that the model cost for prediction — $O(1/n)$ — is negligible with respect to that associated with compression — $O(\log n/n)$.

Thus, the use of prediction is predicated on a "prior belief" that the source we are dealing with does indeed contain states that are permutation-equivalent (as would be the case, say, with "smooth" gray-scale images). In this case, we are willing to invest the price of modeling a predictor for the source. If our belief is correct, the permutation-minimal representation will be strictly smaller than the conventional minimal representation, and we will get the pay-off of a reduction in compression model cost and faster convergence to the entropy. If our belief is incorrect, then the permutation-minimal tree will not be smaller than the minimal tree, the compression model cost will remain the same, and we will have paid an unnecessary prediction model cost (albeit a third-order term penalty). Notice that in no case there is a penalty in the second-order term (the dominant redundancy term).

To formalize these ideas, we present a theorem which in simplified form states that for every ergodic tree source T (which includes the underlying graph and the associated conditional probabilities) a universal scheme combining sequential ranking and context modeling yields a total expected code length $E_T L(x^n)$ satisfying

$$\frac{1}{n} E_T L(x^n) \leq H_n(T) + \frac{k'(\alpha - 1)}{2n} \log n + O(n^{-1}) \tag{1}$$

where $H_n(T)$ denotes the per-symbol binary entropy of n-vectors emitted by T and k' denotes the number of leaves in the *permutation-minimal tree* T' of T. Hence, the combined scheme attains Rissanen's lower

[2] Other mappings of the error value into a set of α different values are possible, and may have compression advantages in certain applications [19]. These alternate mappings amount to more elaborate permutations determined by the predicted value $\hat{x}$.

[3] In the case of binary sources, the concepts of ranking and prediction coincide, as binary prediction is equivalent to guessing what the most likely binary value is for the next symbol, which amounts to ranking the binary alphabet.

bound in an extended source hierarchy that includes permutation-minimal trees. If T' has indeed fewer leaves than T, the combined algorithm converges faster than an optimal algorithm modeling T in the class of general tree sources. The notion of a *hierarchy* of sources here is borrowed from [20]: At the top of the hierarchy are all Markov models of order, say, m, e.g., in the binary case, tree sources with 2^m leaves and all possible parametrizations. Each sub-tree with 2^m-1 leaves defines a sub-class, consisting of all the tree sources whose parametrizations can be reduced, thru merging of two leaves, to the given sub-tree. This process is continued for all possible sub-trees of smaller sizes. Thus, the "place" of a given source in the hierarchy is determined by the underlying tree of its minimal parametrization. The universal algorithms considered here attain Rissanen's lower bound for the smallest sub-class to which a source belongs.[4]

The combined universal scheme is presented in the form of an algorithm named *P-Context*. This algorithm is based on Algorithm *Context* [13, 22, 15], which provides a practical means to estimate tree sources. However, our emphasis is on providing a formal framework for the "success story" of prediction in conjunction with context modeling, rather than on the specific algorithm described, which is presented as an example of how the gains available through prediction can be realized. In fact, similar results could be derived for other universal modeling algorithms for tree sources (e.g., *context-tree weighting* [16], or the algorithm in [14]). In particular, [16] provides an elegant scheme which recursively computes a mixture of all possible models in the class of tree sources (of any size), without explicit enumeration of the models. This "mixing" approach (as opposed to the "plug-in" or "state-estimation" approach of [15] and [14]) provides asymptotic optimality also in a non-probabilistic framework [23]. In order to maintain this property in the extended source hierarchy, it would be necessary to include additional models in the mixture. For example, consider the binary case and assume that the Markov order of the models is at most one. Optimality in the extended hierarchy would require that, besides a two-parameter model given by a Markov source of order one, and a one-parameter memoryless model, the mixture include an additional one-parameter model, in which probabilities are computed under the assumption that $p(0|0) = p(1|1)$. For larger alphabets and trees, one should include all possible permutations of the alphabet at every node. Although this approach is possible in principle, an implementation of the mixture in a way analogous to [16] is an open research problem. Such an implementation would provide a "mixing equivalent" of the practical prediction-based algorithms, which seem to follow an inherently plug-in approach. Using ranking in conjunction with the context-tree weighting method is also possible and would lead to results similar to those presented in this paper. However, for this hybrid (plug-in and mixing) approach, the pointwise (non-probabilistic) optimality of [16] may no longer hold.

Algorithm P-Context builds two models on a common data structure: an over-estimated prediction model (i.e., a non-minimal tree) where at each (over-estimated) state we use a universal ranker based on frequency counts, and a compression model which is in turn used to encode rank indices. Notice that in the compression model built, the conditioning contexts are still determined by the original symbols, but the assigned conditional probabilities correspond to the rank indices. As explained in our discussion on "prior belief," for the sources of interest, the cost of using an over-estimated prediction model will be offset by the savings resulting from a smaller compression model. P-Context was conceived with a theoretical goal in mind, and in fact it may be too complex for some practical applications. However, some of the best published lossless image compression schemes [10, 11, 12] can be seen as simplifications of the basic concepts embodied in P-Context, including the basic paradigm of using a large model for adaptive prediction which in turn allows for a smaller model for compression, and the use of contexts determined by original symbols to encode prediction errors. Moreover, symbol ranking has been proposed in recent lossless image compression algorithms [12, 24] and in other practical applications [25], without a formal convergence proof.

The rest of the paper is organized as follows: In Section 2 we review finite-memory sources and tree

[4] This is sometimes referred to as a *twice universal* algorithm [21].

models, and we formally define the sub-hierarchy of permutation-minimal tree sources. In Section 3 we define sequential ranking of non-binary sources, and we study its combinatorial and probabilistic properties. Here, we generalize some results on universal prediction from [8] to non-binary alphabets. Finally, in Section 4 we describe the P-Context algorithm and present our main theorem.

2 Tree Models

In a finite-memory source, the conditional probability of the next emitted symbol, given all the past, depends only on a finite number of contiguous past observations. In many instances of practical interest, however, fitting a plain Markov model to the data is not the most efficient way to estimate a finite-memory source, since there exist equivalent states that yield identical conditional probabilities. Thus, the number of states, which grows exponentially with the order m in a Markov model, can be dramatically reduced by removing redundant parameters after lumping together equivalent states. The reduced models [13, 14] have been termed *tree sources* [15], since they can be represented with a simple tree structure, and are reviewed next.

Consider a sequence $x^n = x_1x_2\cdots x_n$ of length n over a finite alphabet A of α symbols. Any suffix of x^n is called a *context* in which the "next" symbol x_{n+1} occurs. A probabilistic model P, defined on A^n for all $n \geq 0$, has the finite-memory property if the conditional probability function $p(x_{n+1} = a|x^n) \triangleq P(x^na)/P(x^n)$ satisfies

$$p(x_{n+1} = a|x^n) = p(x_{n+1} = a|u\overline{s}(x^n)) \quad \forall u \in A^*,\ a \in A \tag{2}$$

where $\overline{y}$ denotes the reverse of a string y, and $s(x^n) = x_n \cdots x_{n-\ell+1}$ for some ℓ, $0 \leq \ell \leq m$, not necessarily the same for all strings (the sequence of indices is decreasing, except for the case $\ell = 0$ which is interpreted as defining the empty string λ; thus, we write the past symbols in reverse order). Such a string $s(x^n)$ is called a *state*. In a minimal representation of the model, $\overline{s}(x^n)$ is the shortest context satisfying (2). Now, consider a complete α-ary tree, where the branches are labeled by symbols of the alphabet. Each context defines a node in the tree, reached by taking the path starting at the root with the branch x_n, followed by the branch x_{n-1}, and so on. The set S of states is such that it defines a complete subtree T, with S as the set of leaves. Incurring a slight abuse of notation, we will write $p(a|s)$, $a \in A$, $s \in S$, to denote the conditional probability $p(x_{n+1} = a|u\overline{s}(x^n))$, which is independent of $u \in A^*$. Using the tree T and the associated conditional distributions $\{p(a|s) : a \in A,\ s \in S\}$, we can express the probability $P(x^n)$ as

$$P(x^n) = \prod_{t=0}^{n-1} p(x_{t+1}|s(x^t)) \tag{3}$$

for any string (a detailed description of how this is done for the very first symbols in the string, for which a leaf in the tree may not be defined, is given in [15]). For the sake of simplicity, in the sequel, T will denote the whole model, i.e., both the tree and its associated conditional probabilities.

A tree T is called *minimal* [14, 15] if for every node w in T such that all its successors wb are leaves, there exist a, b, $c \in A$ satisfying $p(a|wb) \neq p(a|wc)$. Clearly, if for some such node w the distributions $p(\cdot\,|wb)$ are equal for all b, we could lump the successors into w and have a smaller complete tree representing the same process. Thus, a minimal tree guarantees that the children of such a node w are not all equivalent to it, and hence they cannot be replaced by the parent node. Notice that even in a minimal tree, there may still be sets of equivalent leaves, not necessarily siblings, having the same associated conditional probabilities. These equivalent nodes could, in principle, be lumped together thus reducing the number of parameters in the model. However, such a reduced parameterization may no longer admit a simple tree implementation nor a practical construction.

Algorithm *Context*, introduced in [13], improved in [22], and further analyzed in [15], provides a practical means to estimate tree sources. The algorithm has two interleaved stages, the first for growing the tree and the second for selecting a distinguished context to define the state $s(x^t)$ for each $t > 0$. There are several variants of the rule for the "optimal" context selection, all of which are based on a stochastic complexity argument; that is, the context which would have assigned the largest probability for its symbol occurrences in the past string should be selected for encoding. The probability of occurrence of a symbol $a \in A$ in a context s may suitably be taken as [26]

$$p(a|s) = \frac{n_a(x^t[s]) + 1/2}{\sum_{b \in A} n_b(x^t[s]) + \alpha/2} \tag{4}$$

where $n_b(x^t[s])$ denotes the number of occurrences of $b \in A$ at context s in x^t. It is shown in [15], for a particular context-selection rule, that the total probability (3) induced from (4) attains Rissanen's lower bound.

Although the parameter reduction achieved by tree models over Markov models can be significant, it does not fully exploit some structural symmetries in the data, which, in practice, are often handled through prediction, as discussed in Section 1. The intuitive ideas there can be formalized by modifying our definition of a minimal tree as follows. Let $\vec{\mathbf{p}}(s) = [p(a_{i_1}|s)\, p(a_{i_2}|s) \cdots p(a_{i_\alpha}|s)]$ denote the *conditional probability vector* associated with a state s of a tree source, where the symbols $a_{i_j} \in A$, $1 \le j \le \alpha$, have been permuted so that $p(a_{i_1}|s) \ge p(a_{i_2}|s) \ge \cdots \ge p(a_{i_\alpha}|s)$. A tree T is called *permutation-minimal* if for every node w in T such that all its successors wb, $b \in A$, are leaves, there exist $b, c \in A$ such that $\vec{\mathbf{p}}(wb) \neq \vec{\mathbf{p}}(wc)$. In the binary case, this means that $p(0|w0) \neq p(0|w1)$ *and* $p(0|w0) \neq p(1|w1)$. Thus, a tree that is not permutation-minimal can be reduced by lumping the redundant successors wb, $b = 0, 1$, into w whenever the distributions at the siblings wb are either identical or symmetric. The common conditional probability vector is also assigned to the resulting leaf w. For example, consider the binary tree T defined by the set of leaves {0, 10, 11}, where the probabilities of 0 conditioned on the leaves are p, $(1-p)$, and p, respectively, and assume $p > 1/2$. Clearly, T is minimal in the traditional sense, but it can be reduced to the root $T' = \{\lambda\}$ in the new sense, with conditional probability vector $[p, 1-p]$ given λ.

3 Sequential Ranking for Non-Binary Tree Sources

Imagine a situation where data is observed sequentially and at each time instant t the alphabet symbols are ranked according to their frequency of occurrence in x^t. Then, after having observed x_{t+1}, we note its rank, and we keep a count of the number of times symbols with that rank occurred. For the highest rank, this would be the number of correct guesses of a sequential predictor based on counts of past occurrences. However, in our case, we are also interested in the number of times the second highest ranked symbol, the third one, and so forth, occurred. We compare these numbers with the number of times the symbol that ends being ranked first, second, and so on *after observing the entire sequence* occurred. Hence, in the latter case we keep track of regular symbol counts and we sort them to obtain a "final ranking," while in the former case we keep track of counts by index, incrementing the count corresponding to index i if x_t happens to be the symbol ranked i-th in the ranking obtained after observing x^{t-1}. In the binary case this process amounts to comparing the number of (sequential) correct predictions with the number of occurrences of the most frequent symbol in the whole sequence. This combinatorial problem on binary sequences is considered in [8, Lemma 1], where it is shown that these quantities differ by at most the number of times the sequence is balanced, i.e., contains as many zeros as ones. Note that at this point we do not distinguish between the different contexts in which the symbols occur. As the problem is essentially combinatorial, all the results still hold when occurrences are conditioned on a given context.

Table 1: Symbol counts, ranking and index counts
Example: $x^{10} = ccabbbcaac$

		Symbol counts	**Ranking**	**Sorted symbol counts**	**Index of**	**Index counts**
t	x_t	$n_\xi(x^t), \xi{=}a, b, c$	$A_i(x^t), i{=}1, 2, 3$	$N_i(x^t), i{=}1, 2, 3$	x_t **in** x^{t-1}	$M_i(x^t), i{=}1, 2, 3$
0	–	0,0,0	a, b, c	0,0,0	–	0,0,0
1	c	0,0,1	c, a, b	1,0,0	3	0,0,1
2	c	0,0,2	c, a, b	2,0,0	1	1,0,1
3	a	1,0,2	c, a, b	2,1,0	2	1,1,1
4	b	1,1,2	c, a, b	2,1,1	3	1,1,2
5	b	1,2,2	b, c, a	2,2,1	3	1,1,3
6	b	1,3,2	b, c, a	3,2,1	1	2,1,3
7	c	1,3,3	b, c, a	3,3,1	2	2,2,3
8	a	2,3,3	b, c, a	3,3,2	3	2,2,4
9	a	3,3,3	a, b, c	3,3,3	3	2,2,5
10	c	3,3,4	c, a, b	4,3,3	3	2,2,6

In order to generalize the result of [8] to any $\alpha \geq 2$, we introduce some definitions and notation. Let $A_i(x^t)$ denote the i-th most numerous symbol in x^t, $0 \leq t \leq n$, $1 \leq i \leq \alpha$ (it is assumed that there is an order defined on A and that ties are broken alphabetically; consequently, $A_i(\lambda)$ is the i-th symbol in the alphabetical order). We define

$$N_i(x^t) \triangleq |\{\ell \,:\, x_\ell = A_i(x^t),\, 1 \leq \ell \leq t\}|, \quad N_i(\lambda) \triangleq 0,\ 1 \leq i \leq \alpha\ , \tag{5}$$

and

$$M_i(x^t) \triangleq |\{\ell \,:\, x_\ell = A_i(x^{\ell-1}),\, 1 \leq \ell \leq t\}|, \quad M_i(\lambda) \triangleq 0,\ 1 \leq i \leq \alpha\ , \tag{6}$$

i.e., $N_i(x^t)$ and $M_i(x^t)$ are, respectively, the number of occurrences of the i-th most numerous symbol after observing the entire sequence x^t, and the number of occurrences of the i-th index. This is exemplified in Table 1 for the sequence $x^{10} = ccabbbcaac$ over $A = \{a, b, c\}$. Our goal is to bound the difference between $N_i(x^n)$ and $M_i(x^n)$. Later on, we consider probabilistic properties of this difference. As these properties will depend on whether the probabilities of the symbols with the i-th and $i+1$-st largest probabilities are equal, it will prove helpful to partition the alphabet into subsets of symbols with identical probabilities. This probability-induced partition provides the motivation for the following discussion, where we consider more general partitions. Specifically, consider integers $0 = j_0 < j_1 < \cdots < j_d = \alpha$, where d is a positive integer not larger than α. These integers induce a partition of the integers between 0 and α into d contiguous subsets of the form $\{j \,:\, j_{r-1} < j \leq j_r\}$, $0 < r \leq d$. This, in turn, defines a partition of A by

$$\mathcal{A}_r(x^t) = \{A_j(x^t) \,:\, j_{r-1} < j \leq j_r\},\ 0 < r \leq d. \tag{7}$$

Thus, each subset $\mathcal{A}_r(x^t)$ of A contains symbols that are contiguous in the final ranking for x^t. The subsets $\mathcal{A}_r(x^t)$, $0 < r \leq d$, will be called *super-symbols*. The partitions and the induced super-symbols are depicted in Figure 1. Notice that while the partition of the integers is fixed, the partition of the alphabet may vary with t, according to the ranking. For typical sequences x^t, this ranking will correspond to the order of the probability values, and the partition of $\{1, 2, \cdots, \alpha\}$ will be defined so that super-symbols consist of symbols of equal probability, as intended. The super-symbols define new occurrence counts

$$\mathcal{N}_i(x^t) \triangleq \sum_{j=j_{i-1}+1}^{j_i} N_j(x^t) = |\{\ell \,:\, x_\ell \in \mathcal{A}_i(x^t),\, 1 \leq \ell \leq t\}|,\ 0 < i \leq d, \tag{8}$$

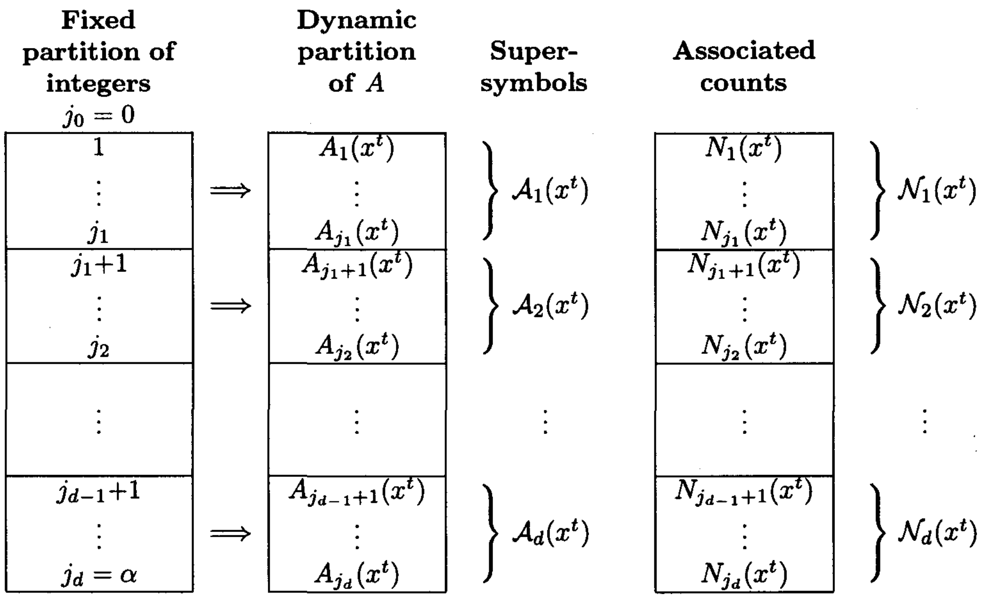


Figure 1: Partition of $\{1, 2, \ldots, \alpha\}$, induced partition of A, and associated counts

and

$$\mathcal{M}_i(x^t) \triangleq \sum_{j=j_{i-1}+1}^{j_i} M_j(x^t) = |\{\ell : x_\ell \in \mathcal{A}_i(x^{\ell-1}),\ 1 \le \ell \le t\}|,\ 0 < i \le d. \tag{9}$$

Finally, let $n_r^*(x^t)$ denote the number of times in x^t that, after having observed $x^{\ell-1}$, $1 \le \ell \le t$, there is a tie in the rankings of x_ℓ, of the first symbol of super-symbol $\mathcal{A}_{r+1}(x^{\ell-1})$, and of the last symbol of super-symbol $\mathcal{A}_r(x^{\ell-1})$, and x_ℓ comes after the latter in the alphabetical order (the notation n_r^* follows [8, Lemma 1], where n^* denotes the number of times a binary sequence contains as many zeros as ones). Specifically, $n_r^*(x^t)$ is defined by

$$\begin{array}{lll} n_0^*(x^t) = n_d^*(x^t) = 0 & & \text{for every sequence } x^t \\ n_r^*(\lambda) = 0 & & 0 < r < d \\ n_r^*(x^{t+1}) = n_r^*(x^t) + \left\{ \begin{array}{ll} 1 & \text{if } N_{j_r}(x^t) = N_{j_r+1}(x^t) = \cdots = N_{j_r+l}(x^t) \\ & \text{and } x_{t+1} = A_{j_r+l}(x^t) \text{ for some } l \ge 1, \\ 0 & \text{otherwise} \end{array} \right. & & t \ge 0,\ 0 < r < d. \end{array} \tag{10}$$

Notice that in the case $\alpha = d = 2$, $n_r^*(x^{t+1})$ is independent of x_{t+1}.

Lemma 1 *For every r, $1 \le r \le d$,*

$$\mathcal{N}_r(x^t) + n_{r-1}^*(x^t) \ge \mathcal{M}_r(x^t) \ge \mathcal{N}_r(x^t) - n_r^*(x^t)\,. \tag{11}$$

Notice that with $\alpha = d = 2$ the inequalities for $r = 1$ and $r = 2$ coincide, as $N_1(x^t) + N_2(x^t) = M_1(x^t) + M_2(x^t) = t$. Thus, the lemma reduces to

$$N_1(x^t) \ge M_1(x^t) \ge N_1(x^t) - n_1^*(x^t)\,. \tag{12}$$

In this case $M_1(x^t)$ is, clearly, the number of correct guesses with a sequential predictor based on previous occurrence counts and alphabetical tie-breaking (in fact, (11) is valid for any tie-breaking policy). Thus, (12) is essentially equivalent to [8, Lemma 1], where expectations have been taken due to a randomized tie-breaking policy. Notice also that taking $d = \alpha$, Lemma 1 bounds the differences $M_i(x^t) - N_i(x^t)$, $1 \le i \le \alpha$.

Proof of Lemma 1. The proof is by induction on t, with the case $t = 0$, for which (11) holds trivially, serving as basis. Assume that (11) holds for every ℓ, $0 \le \ell \le t$. Next, we prove it for $\ell = t + 1$. We distinguish between the cases $x_{t+1} \in \mathcal{A}_r(x^t)$ and $x_{t+1} \notin \mathcal{A}_r(x^t)$.

In the first case, by definition (9), we have $\mathcal{M}_r(x^{t+1}) = \mathcal{M}_r(x^t) + 1$. If $r = 1$ or $N_{j_{r-1}}(x^t) > N_{j_{r-1}+1}(x^t)$, then also $\mathcal{N}_r(x^{t+1}) = \mathcal{N}_r(x^t) + 1$. Otherwise, $r > 1$ and, since $N_{j_{r-1}}(x^t) \ge N_{j_{r-1}+1}(x^t)$, we must have $N_{j_{r-1}}(x^t) = N_{j_{r-1}+1}(x^t)$, in which case either $n^*_{r-1}(x^{t+1}) = n^*_{r-1}(x^t) + 1$ (if x_{t+1} is also tied with $A_{j_{r-1}+1}(x^t)$) or $\mathcal{N}_r(x^{t+1}) = \mathcal{N}_r(x^t) + 1$ (if x_{t+1} has a lower ranking than $A_{j_{r-1}+1}(x^t)$ despite being in $\mathcal{A}_r(x^t)$). In addition, for any fixed infinite sequence x having x^t as a prefix, both $\mathcal{N}_r(\cdot)$ and $n^*_{r-1}(\cdot)$ are non-decreasing functions of t. Hence, in both cases, the left-hand side of (11) holds. The right-hand side follows from the increment in $\mathcal{M}_r(x^{t+1})$ and from the trivial relations $\mathcal{N}_r(x^{t+1}) \le \mathcal{N}_r(x^t) + 1$ and $n^*_r(x^{t+1}) \ge n^*_r(x^t)$.

In the case where $x_{t+1} \notin \mathcal{A}_r(x^t)$, we have $\mathcal{M}_r(x^{t+1}) = \mathcal{M}_r(x^t)$. Thus, the left-hand side of (11) holds, as $\mathcal{N}_r(\cdot)$ and $n^*_{r-1}(\cdot)$ are non-decreasing functions of t. Moreover, if $\mathcal{N}_r(x^{t+1}) = \mathcal{N}_r(x^t)$ then also the right-hand side holds. Otherwise, we have $\mathcal{N}_r(x^{t+1}) = \mathcal{N}_r(x^t) + 1$, which means that there was an increment in the number of occurrences of the r-th ranked super-symbol even though the r-th ranked super-symbol itself did not occur (as $x_{t+1} \notin \mathcal{A}_r(x^t)$). This may happen only if symbols in $\mathcal{A}_r(x^t)$, $r < d$, were tied with symbols in $\mathcal{A}_{r+1}(x^t)$ and, possibly, subsequent super-symbols (containing symbols of lower alphabetical precedence), and one of the latter symbols occurred; namely, if

$$N_{j_r}(x^t) = N_{j_r+1}(x^t) = \cdots = N_{j_r+l}(x^t) \;\text{ and }\; x_{t+1} = A_{j_r+l}(x^t),\; l \ge 1. \tag{13}$$

Hence, by definition (10), we have $n^*_r(x^{t+1}) = n^*_r(x^t) + 1$, implying

$$\mathcal{N}_r(x^{t+1}) - n^*_r(x^{t+1}) = \mathcal{N}_r(x^t) - n^*_r(x^t) \le \mathcal{M}_r(x^t) = \mathcal{M}_r(x^{t+1}) \,, \tag{14}$$

which completes the proof. □

Hereafter, we assume that x^n is emitted by an *ergodic* tree source T with a set of states (leaf contexts) S. For the sake of simplicity, we further assume that x^n is preceded by as many zeros (or any arbitrary, fixed symbol) as needed to have a state defined also for the first symbols in x^n, for which the past string does not determine a leaf of T. For a state $s \in S$ and any time instant t, $1 \le t \le n$, let $x^t[s]$ denote the sub-sequence of x^t formed by the symbols emitted when the source is at state s. The probabilities conditioned on s are used to determine the super-alphabet of Lemma 1 for $x^t[s]$, by defining the partition given by the integers $\{j_r\}$ as follows. First, we sort the conditional probabilities $p(a|s)$, $a \in A$, so that

$$p(a_{i_1}|s) \ge p(a_{i_2}|s) \ge \cdots \ge p(a_{i_\alpha}|s) \,,\; a_{i_j} \in A,\; 1 \le j \le \alpha. \tag{15}$$

Next, we denote $p_j(s) \triangleq p(a_{i_j}|s)$ (the j-th conditional probability value at state s in decreasing order), so that the conditional probability vector associated with state s is $\vec{\mathbf{p}}(s) = [p_1(s), p_2(s), \cdots, p_\alpha(s)]$. We define $d(s) \triangleq |\{p_j(s)\}_{j=1}^{\alpha}|$, $j_0 \triangleq 0$, and for $0 \le r < d(s)$,

$$j_{r+1} = \max\{j \,:\, p_s(j) = p_s(j_r + 1)\} \,. \tag{16}$$

Hence, the partition is such that $p_{j_r}(s) > p_{j_{r+1}}(s)$ and $p_j(s)$ is constant for $j \in [j_r + 1 \,,\, j_{r+1}]$. The symbol a_{i_j}, whose probability is $p_j(s)$, is denoted $b_j(s)$.

Lemma 2 below, in conjunction with Lemma 1, shows that with very high probability the difference between $\mathcal{N}_r(x^t[s])$ and $\mathcal{M}_r(x^t[s])$ is small for the partition (16).

Lemma 2 *For every $s \in S$, every r, $0 \le r \le d(s)$, every positive integer u, and every $t > u$,*

$$\mathrm{Prob}\{n_r^*(x^t[s]) \ge u\} \le K_1 \rho^u \,, \tag{17}$$

where K_1 and ρ are positive constants that depend on T and s, and $\rho < 1$.

The proof of Lemma 2 uses the concept of *properly ordered sequence.* A sequence $x^t[s]$ is said to be properly ordered if for every a, $b \in A$, $p(a|s) > p(b|s)$ implies $n_a(x^t[s]) > n_b(x^t[s])$ (we remind that $n_a(x^t[s])$ denotes the number of occurrences of $a \in A$ at state s in x^t, i.e., the number of occurrences of a in $x^t[s]$). The sequence x^t is said to be properly ordered with respect to T if $x^t[s]$ is properly ordered for every $s \in S$. The following combinatorial lemma on the relation between the partition (16) and properly ordered sequences is used in the proof of Lemma 2.

Lemma 3 *If $N_{j_r}(x^t[s]) = N_{j_r+1}(x^t[s])$ for some $s \in S$ and some r, $0 < r < d(s)$, then x^t is not properly ordered.*

Proof. Given $s \in S$ and r, $0 < r < d(s)$, we introduce the simplified notation $y \triangleq x^t[s]$ and we define

$$B \triangleq \bigcup_{l=1}^{r} \mathcal{A}_l(y), \tag{18}$$

i.e., B is the set of symbols ranked $1, 2, \cdots, j_r$ after having observed y. We further define

$$\bar{B} \triangleq A - B = \bigcup_{l=r+1}^{d(s)} \mathcal{A}_l(y). \tag{19}$$

First, we notice that if, for some s and r, $b_j(s) \in B$ for some $j > j_r$, then we must have $b_{j'}(s) \in \bar{B}$ for some $j' \le j_r$. By (16), this implies that $p_j(s) < p_{j'}(s)$. On the other hand, by the definition of B, $b_j(s)$ is ranked higher than $b_{j'}(s)$ and, consequently, $n_{b_j(s)}(y) \ge n_{b_{j'}(s)}(y)$. Thus, y is not properly ordered, and neither is x^t.

Next, assume that $N_{j_r}(y) = N_{j_r+1}(y)$ for some $s \in S$ and some r, $0 < r < d(s)$. By the above discussion, we can further assume that $b_j(s) \notin B$ for any $j > j_r$, for otherwise there is nothing left to prove. Hence, $B = \{b_j(s) \,:\, 1 \le j \le j_r\}$. It follows that $A_{j_r}(y) = b_j(s)$ for some j, $j \le j_r$, and $A_{j_r+1}(y) = b_{j'}(s)$ for some j', $j' > j_r$. Consequently, $N_{j_r}(y) = N_{j_r+1}(y)$ is equivalent to $n_{b_j(s)}(y) = n_{b_{j'}(s)}(y)$. On the other hand, by (15) and (16), $p_j(s) > p_{j'}(s)$, implying that x^t is not properly ordered. □

Notice that, in fact, ordering the symbols according to their probabilities to define super-symbols and proper orders, is a special case of using an arbitrary ranking $\mathcal{R}$ over A (which may include ties). If $\mathcal{R}(a)$ denotes the number of symbols that are ranked *higher* than $a \in A$ (i.e., $\mathcal{R}(a) = \mathcal{R}(b)$ means that a is tied with b in $\mathcal{R}$), a sequence x^t is properly ordered *with respect to* $\mathcal{R}$ if $\mathcal{R}(a) > \mathcal{R}(b)$ implies that $n_a(x^t) > n_b(x^t)$, where $n_a(x^t)$ denotes the number of occurrences of $a \in A$ in x^t. These concepts do not require the specification of a probabilistic environment (a tree source T) and Lemma 3 applies to any ranking $\mathcal{R}$. On the other hand, the particular ranking (15) implies that the event that a sequence is not properly ordered is a large deviations event, as stated in Lemma 4 below, an essential tool in the proof of Lemma 2.

Lemma 4 *For every $t > 0$,*

$$\mathrm{Prob}\{x^t[s] \text{ is not properly ordered}\} \le K_2 \rho^t, \tag{20}$$

where K_2 and ρ are positive constants that depend on T and s, and $\rho < 1$.

Proof. If $x^t[s]$ is not properly ordered, then there exist b, $c \in A$ such that $p(b|s) > p(c|s)$ and $n_b(x^t[s]) \leq n_c(x^t[s])$. Let $p(b|s) - p(c|s) \triangleq 2\Delta > 0$. Thus, either

$$n_b(x^t[s]) \leq n(x^t[s])(p(b|s) - \Delta), \tag{21}$$

or

$$n_c(x^t[s]) \geq n(x^t[s])(p(c|s) + \Delta), \tag{22}$$

where $n(x^t[s])$ denotes the length of $x^t[s]$. In either case, there exist $\Delta > 0$ and $a \in A$ such that

$$\left| \frac{n_a(x^t[s])}{n(x^t[s])} - p(a|s) \right| \geq \Delta. \tag{23}$$

A classical bound on the probability of the event (23) is derived by applying the large deviations principle [27, Chapter 1] to the pair empirical measure of a Markov chain (see, e.g., [27, Theorem 3.1.13 and ensuing remark], or [28, Lemma 2(a)] for a combinatorial derivation). The results in [27] and [28] can be applied to any tree source by defining an equivalent Markov chain (possibly with a larger number of states [14, 15]), as shown in the proof of [15, Lemma 3]. By [28, Lemma 2(a)],

$$\limsup_{t\to\infty} \frac{1}{t} \log \text{ Prob} \left\{ \left| \frac{n_a(x^t[s])}{n(x^t[s])} - p(a|s) \right| \geq \Delta \right\} \leq -D \tag{24}$$

where D is the minimum value taken over a certain set by the Kullback-Leibler information divergence between two joint distributions over A. Furthermore, since T is assumed ergodic, the equivalent Markov chain is irreducible. It can be shown that this implies $D > 0$ and, consequently, for any ρ such that $0 < 2^{-D} < \rho < 1$, (24) implies the claim of the lemma. □

Proof of Lemma 2. By (10), the cases $r = 0$ and $r = d(s)$ are trivial. Thus, we assume $0 < r < d(s)$. We have

$$\begin{aligned} \text{Prob}\{n_r^*(x^t[s]) \geq u\} &\leq \text{Prob}\{N_{j_r}(x^\ell[s]) = N_{j_r+1}(x^\ell[s]) \text{ for some } \ell \geq u\} \\ &\leq \text{Prob}\{x^\ell[s] \text{ is not properly ordered for some } \ell \geq u\} \end{aligned} \tag{25}$$

where the first inequality follows from the definition of $n_r^*(x^t[s])$ and the second inequality follows from Lemma 3. Thus,

$$\begin{aligned} \text{Prob}\{n_r^*(x^t[s]) \geq u\} &\leq \sum_{\ell=u}^{\infty} \text{Prob}\{x^\ell[s] \text{ is not properly ordered}\} \\ &\leq \sum_{\ell=u}^{\infty} K_2 \rho^\ell = \frac{K_2}{1-\rho} \rho^u \end{aligned} \tag{26}$$

where the second inequality follows from Lemma 4, and the last equality follows from $\rho < 1$. Defining $K_1 = K_2(1-\rho)^{-1}$, the proof is complete. □

The partition (16) and the concept of properly ordered sequence are also instrumental in showing that the number $M_i(x^t[s])$ of occurrences of the i-th ranked symbol along $x^t[s]$ is close to $p_i(s)n(x^t[s])$, with high probability, as one would expect. Note that if all the entries of $\vec{p}(s)$ were different (i.e., if there were no ties in the probability ranking (15)), this would be a direct consequence of Lemmas 1, 2, and 4. However, some difficulties arise in the case where there exist tied probabilities, in which the partition (16) uses $d(s) \neq \alpha$. In this case, Lemma 2 bounds the probability that the number of ties in the sequential ranking be non-negligible, only for contiguous positions i and $i+1$ in the ranking which correspond to non-tied probabilities.

Specifically, given an arbitrary constant $\epsilon > 0$, a sequence x^t is said to be *ϵ-index-balanced* if for every $s \in S$ such that $n(x^t[s]) \neq 0$ and every i, $1 \leq i \leq \alpha$,

$$\left| \frac{M_i(x^t[s])}{n(x^t[s])} - p_i(s) \right| < \epsilon. \tag{27}$$

This means that we can look at the sequence of indices i generated by the sequential ranking as a "typical" sample of a tree source with conditional probabilities $p_i(s)$, except that an index with zero probability may have a non-zero occurrence count (whose value may reach, at most, the number of different symbols that actually occurred in x^t, depending on the alphabetical order[5]). In the same fashion, if for every $a \in A$

$$\left| \frac{n_a(x^t[s])}{n(x^t[s])} - p(a|s) \right| < \epsilon, \tag{28}$$

then the sequence is said to be *ϵ-symbol-balanced.* Lemma 5 below applies to ϵ-index-balanced sequences just as (24) applies to ϵ-symbol-balanced sequences.

Lemma 5 *Let E_1 denote the event that x^t is ϵ-index-unbalanced. Then, for any exponent $\eta > 0$ and every $t > 0$,*

$$P(E_1) < K_3 t^{-\eta} \tag{29}$$

where K_3 is a positive constant.

Proof. First, consider the special case where T is a memoryless source, i.e., there is only one state (consequently, the conditioning state s is deleted from the notation). We further assume that the zero-probability symbols, if any, are ranked in the last places of the alphabetical order. Let y^t denote the sequence of ranking indices generated by x^t, i.e., $x_\ell = A_{y_\ell}(x^{\ell-1})$, $1 \leq \ell \leq t$, and let $P'(y^t)$ denote the probability that y^t be emitted by a memoryless source with ordered probabilities $\{p_i\}_{i=1}^{\alpha}$. By the assumption on the alphabetical order, we have $M_i(x^t) = 0$ when $p_i = 0$. Thus,

$$P'(y^t) = \prod_{i=1}^{\alpha} p_i^{M_i(x^t)}. \tag{30}$$

Using the partition (16) and its related notation, and further denoting with $\beta = j_{d'}$ the number of symbols with *non-zero* probability (so that $d' = d$ if $\alpha = \beta$, $d' = d - 1$ otherwise, and p_β is the smallest non-zero probability), (30) takes the form

$$P'(y^t) = \prod_{r=1}^{d} p_{j_r}^{\mathcal{M}_r(x^t)} = p_\beta^t \prod_{r=1}^{d'-1} \left(\frac{p_{j_r}}{p_\beta} \right)^{\mathcal{M}_r(x^t)} \tag{31}$$

where the last equality follows from $\sum_{i=r}^{d} \mathcal{M}_r(x^t) = \sum_{r=1}^{d'} \mathcal{M}_r(x^t) = t$. On the other hand,

$$P(x^t) = \prod_{a \in A} p(a)^{n_a(x^t)} = \prod_{r=1}^{d} p(b_{j_r})^{\sum_{j=j_{r-1}+1}^{j_r} n_{b_j}(x^t)}. \tag{32}$$

[5] For example, consider a case were the last symbol in the alphabetical order has non-zero probability and it occurs as $x\ \ _1$ at state s. This will increase the count $M_\alpha(x^t[s])$ even though $p_\alpha(s)$ might be zero.

If x^t is properly ordered, then the multiset of numbers $n_{b_j}(x^t)$, $j_{r-1}+1 \le j \le j_r$, is the same as the multiset $N_j(x^t)$, $j_{r-1}+1 \le j \le j_r$, possibly in a permuted order. Hence, by (8), (32) implies

$$\begin{aligned} P(x^t) &= \prod_{r=1}^{d} p_{j_r}^{\mathcal{N}_r(x^t)} = p_\beta^t \prod_{r=1}^{d'-1} \left(\frac{p_{j_r}}{p_\beta}\right)^{\mathcal{N}_r(x^t)} \\ &= P'(y^t) \prod_{r=1}^{d'-1} \left(\frac{p_{j_r}}{p_\beta}\right)^{\mathcal{N}_r(x^t)-\mathcal{M}_r(x^t)} \le P'(y^t) \prod_{r=1}^{d'-1} \left(\frac{p_{j_r}}{p_\beta}\right)^{n_r^*(x^t)} \end{aligned} \tag{33}$$

where the last inequality follows from Lemma 1 and the fact that $p_{j_r} > p_\beta$, $1 \le r < d'$. Now, let E_2 denote the event that x^t is not properly ordered, whose probability $P(E_2)$, by Lemma 4, vanishes exponentially fast with t. Since (33) holds for any properly ordered sequence, a union bound yields

$$\begin{aligned} P(E_1) &< P(E_2) + \sum_{x^t \in E_1} P'(y^t) \prod_{r=1}^{d'-1} \left(\frac{p_{j_r}}{p_\beta}\right)^{n_r^*(x^t)} \\ &\le P(E_2) + \sum_{r=1}^{d'-1} \text{Prob}\,\{n_r^*(x^t) > C\log t\} + t^{C\sum_{r=1}^{d'-1} \log \frac{p_{j_r}}{p_\beta}} \sum_{x^t \in E_1} P'(y^t) \end{aligned} \tag{34}$$

for a suitable constant C to be specified later. By definition, x^t is ϵ-index-unbalanced if and only if y^t is an ϵ-symbol-unbalanced sequence over the alphabet $\{1, 2, \cdots, \alpha\}$, with respect to the memoryless measure $P'(\cdot)$. This is an event E_3 whose probability $P'(E_3)$, by (24), vanishes exponentially fast with t. Thus, using also Lemma 2 and $d \le \alpha$,

$$\begin{aligned} P(E_1) &< P(E_2) + \alpha K_1 t^{C\log\rho} + t^{C\sum_{r=1}^{d'-1} \log \frac{p_{j_r}}{p_\beta}} P'(E_3) \\ &< P(E_2) + \alpha K_1 t^{C\log\rho} + t^{C\alpha \log \frac{p_1}{p_\beta}} P'(E_3). \end{aligned} \tag{35}$$

Choosing C sufficiently large, so that $C\log\rho < -\eta$, completes the proof of the memoryless case with the assumed alphabetical order. It is easy to see that a change in the location of the zero-probability symbols in the alphabetical order may cause a variation of, at most, β in the value of the index counts $M_i(x^t)$, $1 \le i \le \alpha$. Thus, in the memoryless case the lemma holds for any alphabetical order.

Now, consider an ergodic tree source T. We have

$$\begin{aligned} P(E_1) &< \sum_{s\in S} \text{Prob}\left\{ \left| \frac{M_i(x^t[s])}{n(x^t[s])} - p_i(s) \right| \ge \epsilon \text{ for some } i,\ 1 \le i \le \alpha \right\} \\ &< \sum_{s\in S} [P(E_4) + P(E_5) + P(E_6)] \end{aligned} \tag{36}$$

where, for a given $\delta > 0$, the three new events in (36) are defined as follows. Event E_4 consists of sequences such that

$$\left| \frac{n(x^t[s])}{t} - P^{\text{stat}}(s) \right| > \delta \tag{37}$$

where $P^{\text{stat}}(s) \ne 0$ is the stationary probability of s, and we restrict events E_5 and E_6 to sequences in $\bar{E}_4$. Event E_5 consists of sequences such that the subsequence of the first $t(P^{\text{stat}}(s) - \delta)$ emissions at state s is $\epsilon/2$-index-unbalanced (with respect to the conditional measure), and E_6 denotes the event that $x^t \notin E_5$ and

$x^t[s]$ is ϵ-index-unbalanced. Clearly, if $x^t \in E_6$ then $x^\ell[s]$, $1 \leq \ell \leq t$, turns from $\epsilon/2$-index-balanced to ϵ-index-unbalanced in, at most, $2t\delta$ occurrences of s. Taking δ sufficiently small with respect to ϵ and $P^{\text{stat}}(s)$, we can guarantee that this number of occurrences is not sufficient for E_6 to occur. In addition, by the same large deviations arguments that lead to (24) [27, Theorems 3.1.2 and 3.1.6], $P(E_4)$ vanishes exponentially fast. Thus, it suffices to prove that $P(E_5)$ vanishes as required by the lemma. By the "dissection principle" of Markov chains[6], $P(E_5)$ equals the probability that the memoryless source defined by the conditional measure at state s, emit an $\epsilon/2$-index-unbalanced sequence of length $t(P^{\text{stat}}(s) - \delta)$. By our discussion on the memoryless case, this probability vanishes as $[t(P^{\text{stat}}(s) - \delta)]^{-\eta}$, which completes the proof. □

4 The Permutation-Context Algorithm

In this section we demonstrate an algorithm that combines sequential ranking with universal context modeling, and we show that it optimally encodes any ergodic tree source T with a model cost that corresponds to the size of its permutation-minimal tree T'. The scheme will be referred to as the *Permutation-Context* algorithm (or P-Context, for short), as it is based on Algorithm Context and on the concept of permutation-minimal trees. The algorithm assumes knowledge of an upper bound m on the depth of the leaves of T, and its strong optimality is stated in Theorems 1 and 2 below. Rank indices (rather than the original symbols) are sequentially processed, with the nodes of the context tree still defined by the original past sequence over A. Symbol occurrences are ranked at each context of length m and an index is associated to x_{t+1} in context $x_t \cdots x_{t-m+1}$. An encoding node is selected by one of the usual context selection rules [15], using index counts instead of symbol counts, and the index is encoded. Finally, the *index* counts are updated at each node in the path, as well as the *symbol* counts at the nodes of depth m.

We start by describing how the data structure in the P-Context algorithm is constructed and updated. The structure consists of a growing tree $\mathcal{T}_t$, of maximum depth m, whose nodes represent the contexts, and occurrence counts $M'_i(x^t[s])$ for each node s, $1 \leq i \leq \alpha$, which are referred to as *index counts*. In addition, the nodes s_m of depth m in $\mathcal{T}_t$, which are used as *ranking contexts*, have associated counts $n_a(x^t[s_m])$ for every $a \in A$, which are referred to as *symbol counts*. The algorithm grows the contexts and updates the counts by the following rules:

Step 0. Start with the root as the initial tree $\mathcal{T}_0$, with its index counts all zero.

Step 1. Recursively, having constructed the tree $\mathcal{T}_t$ (which may be incomplete) from x^t, read the symbol x_{t+1}. Traverse the tree along the path defined by $x_t, x_{t-1}, \cdots$, until its deepest node, say $x_t \cdots x_{t-\ell+1}$, is reached. If necessary, assume that the string is preceded by zeros.

Step 2. If $\ell < m$, create new nodes corresponding to x_{t-r}, $\ell \leq r < m$, and initialize all index counts as well as the symbol counts at the node s_m of depth m to 0.

Step 3. Using the symbol counts at s_m, find the index i such that $x_{t+1} = A_i(x^t[s_m])$ (thus, x_{t+1} is the i-th most numerous symbol seen at context s_m in x^t). If $\ell < m$, i.e., if s_m has just been created, then $x^t[s_m] = \lambda$ and i is such that x_{t+1} is the i-th symbol in the alphabetical order. Increment the count of symbol x_{t+1} at s_m by one.

[6] In our case, a suitable formulation of this principle can be stated as follows (see, e.g., [29, Proposition 2.5.1] for an alternative formulation): Consider an ergodic Markov chain over a set of states S with a fixed initial state, and let $P(\cdot)$ denote the induced probability measure. For a state $s \in S$, let $P_s(\cdot)$ denote the i.i.d. measure given by the conditional probabilities at s. Let $y^{\ n}$ denote the subsequence of states visited following each of the first n occurrences of s in a semi-infinite sequence x, and let $Y^{\ n}$ denote a fixed, arbitrary n-vector over S. Then, $\text{Prob}\{x : y^{\ n} = Y^n\} = P_s(Y^n)$. The proof can be easily derived from the one in [29].

Step 4. Traverse the tree back from s_m towards the root and for every node s visited increment its index count $M'_i(x^t[s])$ by one. This completes the construction of $\mathcal{T}_{t+1}$.

Clearly, the index counts satisfy

$$M'_i(x^t[s]) = \sum_{s_m\,:\,s \text{ is a prefix of } s_m} M_i(x^t[s_m]) \tag{38}$$

where the counts $M_i(x^t[s_m])$ are defined in (6). Note that, while $M'_i(x^t[s_m]) = M_i(x^t[s_m])$, in general, $M'_i(x^t[s]) \neq M_i(x^t[s])$.

In practice, one may save storage space by limiting the creation of new nodes so that the tree grows only in directions where repeated symbol occurrences take place, as in [13] and [15]. In addition, it is convenient to delay the use of a ranking context until it accumulates a few counts, by use of a shallower node for that purpose. These modifications do not affect the asymptotic behavior of the algorithm, while the above simplified version allows for a cleaner analysis.

The selection of the distinguished context $s^*(x^t)$ that serves as an encoding node for each symbol x_{t+1} is done as in Context algorithm, but using index counts instead of symbol counts. Moreover, we encode the ranking indices rather than the symbols themselves. Thus, the contexts $s^*(x^t)$ are estimates of the leaves of a permutation-minimal tree, rather than a minimal tree in the usual sense. Clearly, as the ranking is based on x^t, which is available to the decoder, x_{t+1} can be recovered from the corresponding index. Specifically, we analyze the context selection rule of [15] but with a different "penalty term." To this end, we need the following definitions. The "empirical" probability of an index i conditioned on a context s at time t is

$$\hat{P}_t(i|s) \triangleq \frac{M'_i(x^t[s])}{\sum_{i=1}^{\alpha} M'_i(x^t[s])} = \frac{M'_i(x^t[s])}{n(x^t[s])} \tag{39}$$

where we take $0/0 \triangleq 0$. For each context sb, $b \in A$, in the tree, define

$$\Delta_t(sb) = \sum_{i=1}^{\alpha} M'_i(x^t[sb]) \log \frac{\hat{P}_t(i|sb)}{\hat{P}_t(i|s)} \tag{40}$$

where hereafter the logarithms are taken to the base 2 and we take $0 \log 0 \triangleq 0$. This is extended to the root by defining $\Delta_t(\lambda) = \infty$. Similarly to [15], $\Delta_t(sb)$ is non-negative and denotes the difference between the (ideal) code length resulting from encoding the indices in context sb with the statistics gathered at the parent s, and the code length resulting from encoding the indices in sb with its own statistics. In its simplest form, the context selection rule is given by

$$\text{find the deepest node } s^*(x^t) \text{ in } \mathcal{T}_t \text{ where } \Delta_t(s^*(x^t)) \geq f(t) \text{ holds,} \tag{41}$$

where $f(t)$ is a penalty term defined, in our case, by $f(t) = \log^{1+\gamma}(t+1)$ with $\gamma > 0$ an arbitrarily chosen constant. If no such node exists, pick $s^*(x^t) = x_t \cdots x_{t-m+1}$. In fact, a slightly more complex selection rule based on (41) is used in [15] to prove asymptotic optimality. That rule is also required in our proof. However, since its discussion would be essentially identical to the one in [15], we omit it in this paper for the sake of conciseness. Whenever properties derived from the selection rule are required we will refer to the corresponding properties in [15]. Note that the penalty term $f(t)$ differs slightly from the one used in [15].

Finally, following [26] and (4), the probability assigned to a symbol $x_{t+1} = a$ whose associated index is i, is

$$p_t(a|s^*(x^t)) = \frac{M'_i(x^t[s^*(x^t)]) + 1/2}{n(x^t[s^*(x^t)]) + \alpha/2}\,. \tag{42}$$

The total probability assigned to the string x^n is derived as in (3), and the corresponding code length assigned by an arithmetic code is

$$L(x^n) = -\sum_{t=0}^{n} \log p_t(x_{t+1}|s^*(x^t)). \tag{43}$$

Notice that in the binary case, the P-Context algorithm reduces to predicting symbol x_{t+1} as $\hat{x}_{t+1} = \arg\max_{a \in A} n_a(x^t[x_t \cdots x_{t-m+1}])$ and applying Algorithm Context to the sequence of prediction errors $x_{t+1} \oplus \hat{x}_{t+1}$, with the conditioning states still defined by the original past sequence x^t. Theorem 1 below establishes the asymptotic optimality of the P-Context algorithm in a strong sense for the case where all the conditional probabilities are non-zero. Later on, we present a modification of the algorithm that covers the general ergodic case. Although the changes to be introduced are relatively minor, we postpone their discussion since it might obscure some of the main issues addressed in Theorem 1.

Theorem 1 *Let T be an arbitrary tree source whose conditional probabilities satisfy $p(a|s) > 0$ for all $a \in A$ and $s \in S$. Then, the expected code length $E_T L(x^n)$ assigned by the P-Context algorithm to sequences x^n emitted by T satisfies*

$$\frac{1}{n} E_T L(x^n) \leq H_n(T) + \frac{k'(\alpha-1)}{2n} \log n + O(n^{-1}), \tag{44}$$

where $H_n(T)$ denotes the per-symbol binary entropy of n-vectors emitted by T, k' denotes the number of leaves in the permutation-minimal tree T' of T, and the $O(n^{-1})$ term depends on T.

Notice that the assumption on the conditional probabilities implies that the tree source is ergodic. Theorem 1 says that P-Context attains Rissanen's lower bound in the extended source hierarchy that includes permutation-minimal trees. This does not contradict the corresponding lower bound for conventional minimal tree sources, as for each given level in the hierarchy of minimal tree sources, the sub-class of sources for which the permutation-minimal tree representation is strictly smaller than the minimal tree representation has Lebesgue measure zero at that level of the parameter space (in the same way that reducible tree sources have measure zero in the class of all Markov sources of a given order). However, the sub-class does capture those sources for which prediction has a beneficial effect, which are interesting in practice. For those sources, the reduction in model size yields a potential reduction in model cost, which is realized by P-Context.

The proof of Theorem 1 uses a key lemma which states that the probability that $s^*(x^t)$ is not a leaf of T' vanishes at a suitable rate when t tends to infinity. This result, stated in Lemma 6 below, parallels [15, Lemma 1]. Its proof, which is given in Appendix A, extends the one in [15] by use of the tools developed in Section 3.

Lemma 6 *Let T be as defined in Theorem 1 and let E^t denote the event that $s^*(x^t)$ is not a leaf of T'. Then, the probability $P(E^t)$ of E^t satisfies*

$$\sum_{t=1}^{\infty} P(E^t) \log t < \infty. \tag{45}$$

Lemma 6 means that the cost of ranking the symbols sequentially, based on an over-estimated model, does not affect the rate at which the probability of the error event vanishes.

Proof of Theorem 1. Let y^n denote the sequence of indices derived from x^n by ranking the symbols sequentially at the nodes of depth m in the tree, as in the P-Context algorithm. Thus, y_t, $0 < t \leq n$, takes values over the integers between 1 and α. Let $\hat{H}(y^n|T')$ denote the conditional entropy with respect to T' of the empirical measure defined in (39), namely

$$\hat{H}(y^n|T') \triangleq -\sum_{s\in S'}\sum_{i=1}^{\alpha}\frac{M_i'(x^n[s])}{n}\log\frac{M_i'(x^n[s])}{n(x^n[s])} \tag{46}$$

where S' denotes the set of leaves of T'. Had the probability assignment (42) been computed using the true (unknown) permutation-minimal tree T' instead of the sequence of contexts derived with the context selection rule, we would have obtained for every sequence x^n a code length $L'(x^n)$ satisfying, [26],

$$\frac{L'(x^n)}{n} \leq \hat{H}(y^n|T') + \frac{k'(\alpha-1)}{2n}\log n + O(n^{-1})\,. \tag{47}$$

In addition, by the arguments in [14, Theorem 4(a)], Lemma 6 implies

$$\frac{1}{n}E_T[L(x^n) - L'(x^n)] = O(n^{-1})\,. \tag{48}$$

Hence, it suffices to prove that

$$E_T\hat{H}(y^n|T') \leq H_n(T) + O(n^{-1})\,. \tag{49}$$

Now, by the definition of T', all the descendants $sv \in S$ of $s \in S'$ have the same associated conditional probability vector, as defined after (15), which is independent of the string v and is denoted by $\vec{\mathbf{p}}(s) = [p_1(s), p_2(s), \cdots, p_\alpha(s)]$. Note that, in fact, this constitutes an abuse of notation since s may not be in S, so that the conditional distribution $p(\cdot|s)$ may not be defined. Now, applying Jensen's inequality to (46) and then using (38), we obtain

$$\hat{H}(y^n|T') \leq -n^{-1}\sum_{s\in S'}\ \sum_{w\,:\,|sw|=m}\ \sum_{i=1}^{\alpha} M_i(x^n[sw])\log p_i(s) \tag{50}$$

where $|sw| = m$ means that sw is a ranking context that has s as a prefix. Note that if we allowed zero-valued conditional probabilities, there might be cases where some $M_i(x^n[sw]) \neq 0$ even though the corresponding probability $p_i(s) = 0$, as noted in the discussion preceding Lemma 5. Consequently, the application of Jensen's inequality in (50) relies on the assumption that all conditional probabilities are non-zero. Since sw also has a prefix which is a state of T we can treat it as a state in a possibly non-minimal representation of the source. Thus,

$$\begin{aligned} E_T\hat{H}(y^n|T') &\leq -n^{-1}\sum_{s,w,i}\log p_i(s)E_T[M_i(x^n[sw])] && (51)\\ &= -n^{-1}\sum_{s,w}\sum_{r=0}^{d(s)-1}\sum_{j=j_r+1}^{j_{r+1}}\log p_j(s)E_T[M_j(x^n[sw])] && (52)\end{aligned}$$

where the j_r's are defined by (16) and, hence, depend on s (however, for the sake of clarity, our notation does not reflect this dependency). The summation ranges of s, w, and i in (51) are as in (50). By the definition of the partition boundaries, (52) takes the form

$$\begin{aligned} E_T\hat{H}(y^n|T') &\leq -n^{-1}\sum_{s,w}\sum_{r=0}^{d(s)-1}\log p_{j_{r+1}}(s)E_T\left[\sum_{j=j_r+1}^{j_{r+1}} M_j(x^n[sw])\right]\\ &= -n^{-1}\sum_{s,w}\sum_{r=1}^{d(s)}\log p_{j_r}(s)E_T\left[\mathcal{M}_r(x^n[sw])\right] && (53)\end{aligned}$$

where the last equality follows from the definition (9). Thus, by Lemma 1,

$$E_T\hat{H}(y^n|T') \leq -n^{-1}\sum_{s,w,r} \log p_{j_r}(s) E_T[\mathcal{N}_r(x^n[sw]) + n^*_{r-1}(x^n[sw])]. \tag{54}$$

To compute $E_T[\mathcal{N}_r(x^n[sw])]$, we partition the set of n-sequences A^n as follows. For each ranking context sw, let $E_1(sw)$ denote the event that $x^n[sw]$ is not properly ordered. By Lemma 4, this is a large deviations event whose probability is upper-bounded by $K_2\rho^n$, where we choose K_2 and ρ as the maximum of the corresponding constants over the set of m-tuples sw. If $x^n \notin E_1(sw)$ then, clearly,

$$\mathcal{N}_r(x^n[sw]) = \sum_{j=j_{r-1}+1}^{j_r} n_{b_j(sw)}(x^n[sw]) \tag{55}$$

for every m-tuple sw and every r, $1 \leq r \leq d(s)$. Here, we recall that $n_{b_j(z)}(x^n[z])$ denotes the number of occurrences in $x^n[z]$ of the symbol with the j-th largest conditional probability at state z. Thus,

$$\begin{aligned} E_T\left[\mathcal{N}_r(x^n[sw])\right] &\leq \sum_{x^n\in E_1(sw)} P(x^n)\mathcal{N}_r(x^n[sw]) + \sum_{j=j_{r-1}+1}^{j_r} E_T[n_{b_j(sw)}(x^n[sw])] \\ &\leq n(x^n[sw])K_2\rho^n + \sum_{j=j_{r-1}+1}^{j_r} p_j(s)\sum_{t=0}^{n-1} P_t(sw) \end{aligned} \tag{56}$$

where $P_t(sw)$ denotes the probability that the state at time t (in a possibly non-minimal representation of T) be sw. Again, by the definition of the j_r's, (56) yields

$$E_T\left[\mathcal{N}_r(x^n[sw])\right] \leq n(x^n[sw])K_2\rho^n + (j_r - j_{r-1})p_{j_r}(s)\sum_{t=0}^{n-1} P_t(sw). \tag{57}$$

In (54) we also need to bound $E_T[n^*_{r-1}(x^n[sw])]$. To this end, we have

$$E_T[n^*_{r-1}(x^n[sw])] = \sum_{u=0}^{\infty} u \text{ Prob}(n^*_{r-1}(x^n[sw]) = u) < \sum_{u=0}^{\infty} u \text{ Prob}(n^*_{r-1}(x^n[sw]) \geq u) \tag{58}$$

for every sw and every r, $1 < r \leq d(s)$. Thus, by Lemma 2,

$$E_T[n^*_{r-1}(x^n[sw])] \leq K_1\sum_{u=0}^{\infty} u\rho^u < \infty, \tag{59}$$

implying

$$E_T\left[\frac{n^*_{r-1}(x^n[sw])}{n}\right] = O\left(\frac{1}{n}\right). \tag{60}$$

Defining the positive constant $Q \triangleq -\min_{a\in A,\, s\in S}\log p(a|s)$, (54), (57), and (60) yield

$$\begin{aligned} E_T\hat{H}(y^n|T') &\leq \alpha Q K_2\rho^n - n^{-1}\sum_{s,w,r}\left[(j_r - j_{r-1})p_{j_r}(s)\log p_{j_r}(s)\sum_{t=0}^{n-1} P_t(sw)\right] \\ &+ Q\sum_{s,w,r} E_T\left[\frac{n^*_{r-1}(x^n[sw])}{n}\right] \\ &= n^{-1}\sum_{s\in S'}\left[h_\alpha(\vec{\mathbf{p}}(s))\sum_{t=0}^{n-1}\sum_{w} P_t(sw)\right] + O\left(\frac{1}{n}\right) \end{aligned} \tag{61}$$

where $h_\alpha(\cdot)$ denotes the binary entropy function over an α-ary alphabet. By the definitions of S and S', (61) can be rewritten as

$$\begin{aligned} E_T\hat{H}(y^n|T') &\leq n^{-1}\sum_{s\in S}\left[h_\alpha\left(\{p(a|s)\}_{a\in A}\right)\sum_{t=0}^{n-1}P_t(s)\right]+O\left(\frac{1}{n}\right) \\ &= -n^{-1}\sum_{s\in S}\sum_{a\in A}\left[\log p(a|s)\sum_{t=0}^{n-1}\sum_{x^n:\, s(x^t)=s, x_{t+1}=a}P(x^n)\right]+O\left(\frac{1}{n}\right) \\ &= -n^{-1}\sum_{x^n\in A^n}P(x^n)\left[\sum_{t=0}^{n-1}\log p(x_{t+1}|s(x^t))\right]+O\left(\frac{1}{n}\right) \\ &= -n^{-1}\sum_{x^n\in A^n}P(x^n)\log P(x^n)+O\left(\frac{1}{n}\right). \end{aligned} \tag{62}$$

By the discussion preceding (49), this completes the proof of Theorem 1. □

Theorem 1 relies on the positivity of all the conditional probabilities since otherwise, as noted in its proof, some value M_i might be non-zero even when the corresponding entry $p_i(s)$ in the probability vector is zero. This is caused by the use of a predetermined alphabetical order in breaking ranking ties and by the fact that the encoder does not have prior knowledge of the effective alphabet. A value M_i corresponding to $p_i(s) = 0$ is upper-bounded by the constant α which implies that the empirical probability $\hat{P}_t(i|s)$ is $O(n^{-1})$. However, its contribution to the empirical entropy (46) is $O(\log n \,/\, n)$, which would void our proof of (49). The problem would be eliminated if the tie-breaking order were such that zero-probability symbols were placed last in the order. To remove the restriction on the conditional probabilities we modify the P-Context algorithm to use a dynamically determined order in each ranking context s_m which is given by the *order of first occurrence* of the symbols in $x^t[s_m]$. Thus, an $M_i \neq 0$ always corresponds to a $p_i(s) \neq 0$. This amounts, in fact, to using a possibly different alphabet (of variable size) in each state, for which all conditional probabilities are positive. The algorithm will guarantee that the decoder will be able to reconstruct the order.

Specifically, the dynamic tie-breaking order used in determining $A_i(x^t[s_m])$ in Step 3 of the P-Context algorithm is derived as follows: If this is the first occurrence of x_{t+1} at context s_m and β different symbols have occurred at this context, then x_{t+1} is assigned the place $\beta + 1$ in the dynamic alphabetical order. Otherwise, if this is not the first occurrence of x_{t+1}, it has already been placed in the order. Notice that this order guarantees that in the first occurrence of a symbol x_{t+1} in a ranking context s_m the assigned index is the smallest i such that $M_i(x^t[s_m]) = 0$ and, therefore, the count is incremented from 0 to 1. The context selection rule does not need any modification, but the formula (42) for the probability assigned to a symbol has to be modified so that the decoder can recover symbols that occurred for the first time in a ranking context, for which the index is not sufficient, since the dynamic order is still unavailable. To this end, we first notice that for a ranking context s_m where $\beta(s_m)$ different symbols occurred, with $\beta(s_m) < \alpha$, the α possible events to be encoded at a selected encoding node $s^*(x^t)$ are of two kinds: on the one hand, the indices 1, 2, $\cdots$, $\beta(s_m)$, from which the decoder can recover the corresponding symbol and, on the other hand, the index $\beta(s_m) + 1$ in conjunction with one of the $\alpha - \beta(s_m)$ possible new symbols at s_m. For events of the first kind, the probability assigned to a symbol $x_{t+1} = a$ whose associated index is $i \leq \beta(s_m)$, is

$$p'_t(a|s^*(x^t)) = \frac{M'_i(x^t[s^*(x^t)]) + 1/2}{n(x^t[s^*(x^t)]) + (\beta(s_m) + 1)/2}. \tag{63}$$

For each event of the second kind, we assign the probability

$$p'_t(a|s^*(x^t)) = \frac{\sum_{i=\beta(s_m)+1}^{\alpha} M'_i(x^t[s^*(x^t)]) + 1/2}{n(x^t[s^*(x^t)]) + (\beta(s_m)+1)/2} \cdot \frac{1}{\alpha - \beta(s_m)}, \tag{64}$$

i.e., the remaining probability mass is distributed uniformly among the $\alpha - \beta(s_m)$ events of this kind. If $\beta(s_m) = \alpha$, then we use the probability assignment (42). Note that with $\beta(s_m) = \alpha - 1$ both assignments coincide.

For the P-Context algorithm, modified as described above, we have:

Theorem 2 *Let T be an arbitrary, ergodic tree source. Then, the expected code length $E_T L(x^n)$ assigned by the modified P-Context algorithm to sequences x^n emitted by T satisfies*

$$\frac{1}{n} E_T L(x^n) \le H_n(T) + \frac{\sum_{s \in S'} \alpha(s)}{2n} \log n + O(n^{-1}), \tag{65}$$

where $H_n(T)$ denotes the per-symbol binary entropy of n-vectors emitted by T, S' denotes the set of leaves of the permutation-minimal tree T' of T, $\alpha(s)$ is the minimum between $\alpha - 1$ and the number of non-zero entries in the probability vector $\vec{\mathbf{p}}(s)$, and the $O(n^{-1})$ term depends on T.

Note that if all the conditional probabilities are positive, (65) reduces to (44) of Theorem 1. Otherwise, there are savings in the asymptotic model cost. However, these savings are not enough to attain the optimum $0.5n^{-1} \sum_{s \in S'} (\alpha(s) - 1)$ that could have been achieved if the fact that the source belonged to a subclass with a reduced number of free parameters was known *a priori* by the coder. The k' additional parameters is the cost that is paid for accomplishing the task *sequentially.*

The proof of Theorem 2 follows that of Theorem 1 almost *verbatim* if we notice that since $p_i(s) = 0$ implies $M_i(x^t[s]) = 0$, Jensen's inequality can be safely used in deriving (50) and (A.4). The use of the results in [26] to obtain an analogous to (47) is based on the fact that the assignment (64) is employed a finitely bounded number of times and, therefore, the total contribution of the factors $1/(\alpha - \beta(s_m))$ to the per-symbol code length is $O(n^{-1})$. Disregarding this factor, since $\beta(s_m) + 1$ in (63) is upper-bounded by $\alpha(s)$, where s is the unique prefix of s_m in S', the assignment $p'_t(x_{t+1}|s^*(x^t))$ is clearly at least as large as $p_t(x_{t+1}|s^*(x^t))$ even if the latter were computed with an alphabet size of $\alpha(s) + 1$. This is the origin of the model cost in Theorem 2. Finally, in (A.3), the divisor $p_\alpha(s)$, which may be zero, is replaced by $p_{\alpha(s)+1}(s)$, similarly to the proof of Lemma 5 (eq. (31)).

A Appendix: Proof of Lemma 6

By the context selection rule and proceeding as in the proofs of Lemmas 1, 2, and 3 of [15], it can be shown that there exist only two non-trivial situations in which a sequence x^t can lead to the selection of a context $s^*(x^t)$ which is not a leaf of T':

a. There exists a leaf s of T' such that a longer context swc, $c \in A$, $|s| \le |sw| < m$, for which s is a prefix, satisfies $\Delta_t(swc) \ge f(t)$. In this case, there is an over-estimation error.

b. There exists a node z such that all its successors are leaves of T', for which $\sum_{b \in A} \Delta_t(zb) < \alpha f(t)$. This case may lead to an under-estimation error.

First, we consider the over-estimation case, for which we introduce some additional notation. For a context z and a tree $\mathcal{T}$, let $\mathcal{T}_z$ denote the minimal complete tree containing $\mathcal{T}$ and z. With s, w, and c defined as above and fixed, let S_{sw} denote the set of leaves of $\mathcal{T}_{sw}$. For a node v in $\mathcal{T}_{sw}$, let $S(v)$ denote the set of leaves of $\mathcal{T}_{sw}$ having v as a prefix. Clearly,

$$\log P(x^t) = \sum_{z\in S_{sw}-S(sw)} \sum_{a\in A} n_a(x^t[z]) \log p(a|z) + P_{sw} \tag{A.1}$$

where

$$P_{sw} \triangleq \sum_{z\in S(sw)} \sum_{a\in A} n_a(x^t[z]) \log p(a|z). \tag{A.2}$$

Since s is a leaf of $\mathcal{T}'$, all the contexts in $S(sw)$ share a common probability vector independent of w, which by abuse of notation is denoted $\vec{\mathbf{p}}(s) = [p_1(s), \cdots, p_\alpha(s)]$, as defined after (15). We use $n_j(x^n[z])$ as a simplified notation for the number $n_{b_j(z)}(x^n[z])$ of occurrences in $x^n[z]$ of the symbol with the j-th largest conditional probability at state z. Thus,

$$\begin{aligned}
P_{sw} &= \sum_{z\in S(sw)} \sum_{j=1}^{\alpha} n_j(x^t[z]) \log p_j(s) = \sum_{j=1}^{\alpha} \left[\log p_j(s) \sum_{z\in S(sw)} n_j(x^t[z]) \right] \\
&= \sum_{j=1}^{\alpha} \left[\log p_j(s) \sum_{z\in S(sw)} M'_j(x^t[z]) \right] + \sum_{j=1}^{\alpha} \left[\log p_j(s) \sum_{z\in S(sw)} [n_j(x^t[z]) - M'_j(x^t[z])] \right] \\
&= \sum_{j=1}^{\alpha} M'_j(x^t[sw]) \log p_j(s) + \sum_{j=1}^{\alpha-1} \left[\log \frac{p_j(s)}{p_\alpha(s)} \sum_{z\in S(sw)} [n_j(x^t[z]) - M'_j(x^t[z])] \right] \\
&\triangleq P_{sw}^{(1)} + P_{sw}^{(2)}
\end{aligned} \tag{A.3}$$

where the fourth equality follows from (38) and from the fact that $\sum_{j=1}^{\alpha} [n_j(x^t[z]) - M'_j(x^t[z])] = 0$ for every z. By (39) and Jensen's inequality, and then using (38) and (40), we obtain

$$P_{sw}^{(1)} \le \sum_{j=1}^{\alpha} \left[\sum_{b\in A,\, b\neq c} M'_j(x^t[swb]) \log \hat{P}_t(j|sw) + M'_j(x^t[swc]) \log \hat{P}_t(j|swc) \right] - \Delta_t(swc). \tag{A.4}$$

Next, we modify an argument used in the proof of [15, Lemma 2] by defining, for a fixed sequence x^t, a new process $Q_{swc}(\cdot\,; x^t)$ by the tree $\mathcal{T}_{swc}$, which is used to assign conditional probabilities as follows. For $z \in S_{sw} - S(sw)$, we associate the original conditional probabilities $p(\cdot|z)$, while symbols $y_{\ell+1}$ occurring at nodes swb, $b \in A$, in a sequence y^t, are mapped to indices j such that $y_{\ell+1} = A_j(y^\ell[s_m])$, where s_m denotes the corresponding ranking context of length m. Then, we assign the conditional probabilities $\hat{P}_t(j|sw)$ for occurrences at swb, $b \neq c$, and $\hat{P}_t(j|swc)$ in case $b = c$, where the empirical probabilities, defined by (39), correspond to the fixed sequence x^t. Clearly, given the entire sequence x^t, the probability $Q_{swc}(\cdot\,; x^t)$ can be assigned *sequentially* to any sequence y^t and, consequently,

$$\sum_{y^t\in A^t} Q_{swc}(y^t; x^t) = 1. \tag{A.5}$$

Thus, we can define equivalence classes on A^t in a way similar to [15, Lemma 2] (but using index counts instead of symbol counts) to show that

$$\sum_{x^t\in A^t} Q_{swc}(x^t; x^t) \le (t+1)^{2\alpha} \tag{A.6}$$

where the right hand side is a bound on the number of classes. On the other hand, (A.1), (A.3), (A.4), and the definition of the process $Q_{swc}(\cdot\,; x^t)$, imply that

$$\log P(x^t) \leq \log Q_{swc}(x^t; x^t) - \Delta_t(swc) + P^{(2)}_{sw}\,. \tag{A.7}$$

Since over-estimation occurs whenever $\Delta_t(swc) \geq f(t) = \log^{1+\gamma}(t+1)$, the probability $P_t^{\mathrm{over}}(swc)$ of over-estimation at time t for the node swc can be upper-bounded as

$$P_t^{\mathrm{over}}(swc) \leq \sum_{x^t:\, \Delta_t(swc) \geq f(t)} \frac{Q_{swc}(x^t; x^t) 2^{P^{(2)}_{sw}}}{2^{\Delta_t(swc)}} \leq \frac{1}{(t+1)^{\log^\gamma(t+1)}} \sum_{x^t \in A^t} Q_{swc}(x^t; x^t) 2^{P^{(2)}_{sw}}\,. \tag{A.8}$$

Thus, we can use (A.6) to upper-bound the over-estimation probability, provided we find a uniform upper bound on $P^{(2)}_{sw}$ for all sequences x^t except a suitably small set. To this end, we apply the tools developed in Section 3. By (A.3), the definition of the set $S(sw)$, and (38), we have

$$P^{(2)}_{sw} = \sum_{j=1}^{\alpha-1} \left[\log \frac{p_j(s)}{p_\alpha(s)} \sum_{v\,:\,|swv|=m} [n_j(x^t[swv]) - M_j(x^t[swv])] \right] , \tag{A.9}$$

which, using the partition (16) and its related notation, can be written as

$$P^{(2)}_{sw} = \sum_{r=1}^{d(s)-1} \left[\log \frac{p_{j_r}(s)}{p_\alpha(s)} \sum_{j=j_{r-1}+1}^{j_r} \sum_{v\,:\,|swv|=m} [n_j(x^t[swv]) - M_j(x^t[swv])] \right] . \tag{A.10}$$

Now, if $x^t[swv]$ is properly ordered, then (A.10) takes the form

$$\begin{aligned} P^{(2)}_{sw} &= \sum_{r=1}^{d(s)-1} \left[\log \frac{p_{j_r}(s)}{p_\alpha(s)} \sum_{v\,:\,|swv|=m} [\mathcal{N}_r(x^t[swv]) - \mathcal{M}_r(x^t[swv])] \right] \\ &\leq \sum_{r=1}^{d(s)-1} \left[\log \frac{p_{j_r}(s)}{p_\alpha(s)} \sum_{v\,:\,|swv|=m} n^*_r(x^t[swv]) \right] \end{aligned} \tag{A.11}$$

where the last inequality follows from Lemma 1. Proceeding as in the proof of Lemma 5, we first use Lemma 2 to show that

$$\mathrm{Prob}\left\{ x^t : \sum_{v\,:\,|swv|=m} n^*_r(x^t[swv]) > 2^m C \log(t+1) \text{ for some } r,\ 1 \leq r < d(s) \right\} < \alpha 2^m K_1 (t+1)^{C \log \rho} \tag{A.12}$$

for an arbitrary constant C. This leads to the desired uniform bound on $P^{(2)}_{sw}$ for sequences in the complementary event. By (A.8), Lemma 4, (A.12), and (A.11), it then follows that

$$P_t^{\mathrm{over}}(swc) \leq K_2 \rho^t + \alpha 2^m K_1 (t+1)^{C \log \rho} + \frac{1}{(t+1)^{\log^\gamma(t+1)}} \sum_{x^t \in A^t} Q_{swc}(x^t; x^t)(t+1)^{2^m C \sum_{r=1}^{d(s)-1} \log \frac{p_{j_r}(s)}{p_\alpha(s)}}\,. \tag{A.13}$$

By (A.6) and with $R(s) \triangleq \sum_{r=1}^{d(s)-1} \log \frac{p_{j_r}(s)}{p_\alpha(s)}$, (A.13) takes the form

$$P_t^{\mathrm{over}}(swc) \leq K_2 \rho^t + \alpha 2^m K_1 t^{C \log \rho} + (t+1)^{-\log^\gamma(t+1) + 2\alpha + 2^m C R(s)}\,. \tag{A.14}$$

Thus, for an appropriate choice of C, the over-estimation probability is summable as desired[7].

Next, we turn to the under-estimation probability $P_t^{\text{under}}(z)$ associated with a node z such that all its successors are leaves of T', as stated in the definition of the under-estimation case. Clearly, it suffices to show that this probability is summable as desired. We have

$$P_t^{\text{under}}(z) \leq \text{Prob}\left\{ x^t : \sum_{b \in A} \Delta_t(zb) < \alpha f(t) \right\}. \tag{A.15}$$

Now, by (40),

$$\sum_{b \in A} \Delta_t(zb) = n(x^t[z]) h_\alpha \left(\left\{ \frac{M_j'(x^t[z])}{n(x^t[z])} \right\}_{j=1}^{\alpha} \right) - \sum_{b \in A} n(x^t[zb]) h_\alpha \left(\left\{ \frac{M_j'(x^t[zb])}{n(x^t[zb])} \right\}_{j=1}^{\alpha} \right). \tag{A.16}$$

By Lemma 5, we can assume that x^t is ϵ_1-index-balanced for some $\epsilon_1 > 0$, as the probability of the complementary event is summable as desired. In this case, for every m-tuple s_m and every j, $1 \leq j \leq \alpha$, we have

$$\left| \frac{M_j(x^t[s_m])}{n(x^t[s_m])} - p_j(s_m) \right| < \epsilon_1. \tag{A.17}$$

If s_m is a descendant of the leaf $zb \in S'$, we have $p_j(s_m) = p_j(zb)$. Consequently, summing (A.17) over the m-tuples that are descendants of zb we get

$$\left| \frac{M_j'(x^t[zb])}{n(x^t[zb])} - p_j(zb) \right| < \epsilon_1. \tag{A.18}$$

By the continuity of the function $h_\alpha(\cdot)$, (A.16) and (A.18) yield

$$\sum_{b \in A} \Delta_t(zb) \geq n(x^t[z]) h_\alpha \left(\left\{ \sum_{b \in A} p_j(zb) \frac{n(x^t[zb])}{n(x^t[z])} \right\}_{j=1}^{\alpha} \right) - \sum_{b \in A} n(x^t[zb]) h_\alpha(\vec{\mathbf{p}}(zb)) - t\epsilon_2 \tag{A.19}$$

for some $\epsilon_2 > 0$, which can be made arbitrarily small by letting ϵ_1 approach 0. Since $t^{-1} f(t) \to 0$ as $t \to \infty$, it follows from (A.15) that it suffices to prove that

$$\text{Prob}\left\{ x^t : \frac{n(x^t[z])}{t} h_\alpha \left(\left\{ \sum_{b \in A} p_j(zb) \frac{n(x^t[zb])}{n(x^t[z])} \right\}_{j=1}^{\alpha} \right) - \sum_{b \in A} \frac{n(x^t[zb])}{t} h_\alpha(\vec{\mathbf{p}}(zb)) < \epsilon \right\} \tag{A.20}$$

is summable as desired for some $\epsilon > 0$. By applying the large deviations result of [28, Lemma 2(a)] (see also [27, Theorem 3.1.13]) in a way similar to the proof of [15, Lemma 3], it can be shown that this holds provided that

$$h_\alpha \left(\left\{ \sum_{b \in A} p_j(zb) \frac{P^{\text{stat}}(zb)}{P^{\text{stat}}(z)} \right\}_{j=1}^{\alpha} \right) - \sum_{b \in A} \frac{P^{\text{stat}}(zb)}{P^{\text{stat}}(z)} h_\alpha(\vec{\mathbf{p}}(zb)) > 0, \tag{A.21}$$

where for a node s in T

$$P^{\text{stat}}(s) \triangleq \sum_{u \,:\, su \in S} P^{\text{stat}}(su), \tag{A.22}$$

[7] Note that any $o(t)$ penalty term of the form $g(t)\log(t+1)$, where $g(t)$ is an arbitrary, unbounded, increasing function of t, would suffice to make $P_t^{\text{over}}(swc)$ summable. In (A.14), we have $g(t) = \log^{\gamma}(t+1)$.

and $P^{\mathbf{stat}}(su)$ denotes the (unique) stationary distribution defined on S by the tree source. Note that, as in [15, Lemma 3], we can assume that the process generated by T is a unifilar Markov chain (possibly with a number of states larger than $|S|$). By Jensen's inequality, the strict inequality (A.21) holds, for otherwise $\vec{\mathbf{p}}(zb)$ would be independent of b, which would contradict the permutation-minimality of T'. □

References

[1] A. Netravali and J. O. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, pp. 366–406, 1980.

[2] M. Feder, N. Merhav, and M. Gutman, "Universal prediction of individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-38, pp. 1258–1270, July 1992.

[3] M. Feder and N. Merhav, "Relations between entropy and error probability," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 259–266, Jan. 1994.

[4] S. Todd, G. G. Langdon, Jr., and J. Rissanen, "Parameter reduction and context selection for compression of the gray-scale images," *IBM Jl. Res. Develop.*, vol. 29 (2), pp. 188–193, Mar. 1985.

[5] J. F. Hannan, "Approximation to Bayes risk in repeated plays," in *Contributions to the Theory of Games, Volume III, Annals of Mathematics Studies*, pp. 97–139, Princeton, NJ, 1957.

[6] T. M. Cover, "Behavior of sequential predictors of binary sequences," in *Proc. 4th Prague Conf. Inform. Theory, Statistical Decision Functions, Random Processes*, (Prague), pp. 263–272, Publishing House of the Czechoslovak Academy of Sciences, 1967.

[7] T. M. Cover and A. Shenhar, "Compound Bayes predictors for sequences with apparent Markov structure," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-7, pp. 421–424, May/June 1977.

[8] N. Merhav, M. Feder, and M. Gutman, "Some properties of sequential predictors for binary Markov sources," *IEEE Trans. Inform. Theory*, vol. IT-39, pp. 887–892, May 1993.

[9] J. L. Mitchell and W. B. Pennebaker, *JPEG Still Image Data Compression Standard.* Van Nostrand Reinhold, 1993.

[10] M. J. Weinberger, J. Rissanen, and R. Arps, "Applications of universal context modeling to lossless compression of gray-scale images," *IEEE Trans. Image Processing*, vol. 5, pp. 575–586, Apr. 1996.

[11] X. Wu, "An algorithmic study on lossless image compression," in *Proc. of the 1996 Data Compression Conference*, (Snowbird, Utah, USA), pp. 150–159, Mar. 1996.

[12] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based, lossless image compression algorithm," in *Proc. of the 1996 Data Compression Conference*, (Snowbird, Utah, USA), pp. 140–149, Mar. 1996.

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

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

[15] 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.

[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] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 629–636, July 1984.

[18] J. O'Neal, "Predictive quantizing differential pulse code modulation for the transmission of television signals," *Bell Syst. Tech. J.*, vol. 45, pp. 689–722, May 1966.

[19] R. F. Rice, "Some practical universal noiseless coding techniques - Part III," Tech. Rep. JPL-91-3, Jet Propulsion Laboratory, Pasadena, CA, Nov. 1991.

[20] M. Feder and N. Merhav, "Hierarchical universal coding," *IEEE Trans. Inform. Theory*, 1996. To appear.

[21] B. Ryabko, "Twice-universal coding," *Problems of Information Transmission*, vol. 20, pp. 173–177, July/September 1984.

[22] G. Furlan, *Contribution a l'Etude et au Développement d'Algorithmes de Traitement du Signal en Compression de Donnëes et d'Images.* PhD thesis, l'Université de Nice, Sophia Antipolis, France, 1990. (In French).

[23] M. J. Weinberger, N. Merhav, and M. Feder, "Optimal sequential probability assignment for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-40, pp. 384–396, Mar. 1994.

[24] P. G. Howard and J. S. Vitter, "Fast and efficient lossless image compression," in *Proc. of the 1993 Data Compression Conference*, (Snowbird, Utah, USA), pp. 351–360, Mar. 1993.

[25] L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, and M. A. Picheny, "Robust methods for using context-dependent features and models in a continuous speech recognizer," in *Proceedings IEEE ICASSP-94*, (Adelaide, South Australia), pp. I533–I536, 1994.

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

[27] A. Dembo and O. Zeitouni, *Large Deviations Techniques and Applications.* Boston, London: Jones and Bartlett, 1993.

[28] I. Csiszar, T. M. Cover, and B.-S. Choi, "Conditional limit theorems under Markov conditioning," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 788–801, Nov. 1987.

[29] S. I. Resnick, *Adventures in Stochastic Processes.* Boston: Birkhäuser, 1992.