[7] D. Gitchell and N. Tran, "A utility for detecting similarity in computer programs," in *Proc. 30th ACM Special Interest Group on Computer Science Education Tech. Symp.*, New Orleans, LA, 1998, pp. 266–270.
[8] M. Li, J. Badger, X. Chen, S. Kwong, P. Kearney, and H. Zhang, "An information-based sequence distance and its application to whole mitochondrial genome phylogeny," *Bioinformatics*, vol. 17, no. 2, pp. 149–154, 2001.
[9] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*, 2nd ed. New York: Springer-Verlag, 1997.
[10] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, "The similarity metric," in *Proc. 14th Annu. ACM-SIAM Symp. Discrete Algorithms*, Baltimore, MD, 2003, pp. 863–872.
[11] M. Li and P. Vitanyi, "Reversibility and adiabatic computation: Trading time and space for energy," *Proc. Roy. Soc. London*, ser. A, vol. 452, pp. 769–789, 1996.
[12] T. Łuczak and W. Szpankowski, "A suboptimal lossy data compression based on approximate pattern matching," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1439–1451, Sept. 1997.
[13] G. Malpohl. JPlag: Detecting Software Plagiarism. [Online]. Available: http://www.ipd.uka.de:2222/index.html
[14] K. Ottenstein, "An algorithmic approach to the detection and prevention of plagiarism," *SIGCSE Bull.*, vol. 8, no. 4, pp. 30–41, 1977.
[15] A. Parker and J. Hamblen, "Computer algorithms for plagiarism detection," *IEEE Trans. Education*, vol. 32, pp. 94–99, May 1989.
[16] S. C. Sahinalp, M. Tasan, J. Macker, and Z. M. Ozsoyoglu, "Distance based indexing for string proximity search," in *Proc. Int. Conf. Data EngineeringICDE'2003*, Bangalore, India, Mar. 2003, pp. 125–138.
[17] S. Schleimer, D. S. Wilkerson, and A. Aiken, "Winnowing: Local algorithms for document fingerprinting," in *Proc. ACM SIGMOD Conf.*, San Diego, CA, June 9–12, 2003, pp. 76–85.
[18] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, July and Oct. 1948.
[19] W. Weaver and C. E. Shannon, *The Mathematical Theory of Communication*. Chicago, IL: Univ. Illinois Press, 1949.
[20] G. Whale, "Plague: Plagiarism Detection Using Program Structure," Dept. Comput. Sci., Univ. New South Wales, Kensington, Australia, Tech. Rep. 8805, 1988.
[21] ——, "Identification of program similarity in large populations," *Computer J.*, vol. 33, no. 2, pp. 140–146, 1990.
[22] M. Wise, "Running Karp–Rabin Matching and Greedy String Tiling," Dept. Comput. Sci., Sydney Univ., Sydney, Australia, Tech. Rep., 1994.
[23] ——, "YAP3: Improved detection of similarities in computer program and other texts," in *Proc. 27th SCGCSE Tech. Symp.*, Philadelphia, PA, 1996, pp. 130–134.
[24] E.-h. Yang and J. C. Kieffer, "On the performance of data compression algorithms based upon string matching," *IEEE Trans. Inform. Theory*, vol. 44, pp. 47–65, Jan. 1998.
[25] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May 1977.

# Universal Entropy Estimation Via Block Sorting

Haixiao Cai, *Student Member, IEEE*,
Sanjeev R. Kulkarni, *Fellow, IEEE*, and Sergio Verdú, *Fellow, IEEE*

***Abstract*—In this correspondence, we present a new universal entropy estimator for stationary ergodic sources, prove almost sure convergence, and establish an upper bound on the convergence rate for finite-alphabet finite memory sources. The algorithm is motivated by data compression using the Burrows–Wheeler block sorting transform (BWT). By exploiting the property that the BWT output sequence is close to a piecewise stationary memoryless source, we can segment the output sequence and estimate probabilities in each segment. Experimental results show that our algorithm outperforms Lempel–Ziv (LZ) string-matching-based algorithms.**

***Index Terms*—Block sorting, Burrows-Wheeler transform (BWT), entropy estimation, piecewise stationary memoryless source, tree source.**

## I. Introduction

Ever since Shannon's initial work on the entropy of English text [26], there has been significant interest in the problem of how to estimate the entropy of sources whose statistical characterization is unknown.

In the area of entropy estimation for independent and identically distributed (i.i.d.) sources, the extension to countably infinite alphabet has been considered in [1], [36]. It is shown in [36] that, for the class of discrete memoryless sources with a countable alphabet and finite entropy variance, i.e., $E[(-\log_2 P(Z))^2] < \infty$, there can be no universal estimator that converges at a rate faster than $O(1/(\log_2 n)^{1+\epsilon})$ for all sources and any $\epsilon > 0$. Furthermore, both the Lempel–Ziv (LZ) string matching estimator and the plug-in estimator achieve the convergence rate $O(1/\log_2 n)$. In [21], it is pointed out that in the undersampled regime, i.e., the alphabet size is comparable to $n$, the plug-in estimator can be contaminated by the bias, although the variance converges to zero fast. Hence, an entropy estimator aimed at simultaneously minimizing bias and variance is proposed and its application in neural science shown in [21]. In the setting of high-dimensional i.i.d. sources, the minimal spanning tree approach is proposed in [11] as an alternative entropy estimator.

Entropy estimators for sources with memory are often related to universal data compression algorithms. The entropy estimator introduced in [35] was motivated by the LZ data compression algorithm. The basic idea is to find the longest string pattern that has occurred previously in the sequence of length $n$. The length of this longest string is denoted by $L_n$. The corresponding estimator is $H_n = \log_2 n / L_n$. This universal estimator converges to the entropy in probability for all stationary ergodic processes [35].[1] Faster convergence can be achieved by an averaging procedure [15], [22], [27], which basically averages longest-

Manuscript received May 5, 2003; revised March 31, 2004. This work was supported in part by ARL MURI under Grant DAAD19-00-1-0466, Draper Laboratory under IR&D 6002 Grant DL-H-546263, and the National Science Foundation under Grant CCR-0312413. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002.

The authors are with the Electrical Engineering Department, Princeton University, Princeton, NJ 08544 USA.

Communicated by E.-h. Yang, Guest Editor.
Digital Object Identifier 10.1109/TIT.2004.830771

[1]Almost-sure convergence may fail depending on the details of how $\boldsymbol{L}_n$ is defined (see [30]), although in the setting of double-sided sequences and static model, almost-sure convergence was proved in [20].

string-matching lengths attained from different starting points. Convergence of the averaging algorithms is proved under an additional condition on the memory structure of the source, namely, the Doeblin condition (see [15], [22], [27]). A context tree weighing- (CTW)-based estimator is used in [9]. Other universal data compression methods, such as *bzip* [24] or irreducible grammar-based codes and their grammar entropies [14], [37] can also be used for entropy estimation.

Reference [28] analyzes an empirical entropy estimator, which computes the empirical distribution $\hat{q}_{k(n)}$ of $k(n)$-blocks and then takes $H(\hat{q}_{k(n)})/k(n)$ as an estimate of the entropy. It is proved in [28 (Theorem II.3.5)] that if $k(n) \to \infty$ as $n \to \infty$ and $k(n) \leq \log_{|\chi|} n$, then the empirical entropy estimator converges to the entropy almost surely for any stationary ergodic processes.

Recent applications of entropy estimation include DNA sequences [16], [23], neural spike trains [9], [29], [19], and image processing [12].

Our new entropy estimation algorithm is motivated by data compression using the Burrows–Wheeler (or block-sorting) transform (BWT) as a preprocessor [5]. A variety of compression algorithms with linear complexity (by using a suffix tree data structure [18]) have been proposed recently using the BWT as a front end followed by modules such as move-to-front, run-length coding, and adaptive Huffman coding. The properties of BWT-based compression algorithms have been analyzed in [8], where a proof of universal optimality is given. The actual performance on real data has been shown to be quite competitive compared with the LZ algorithm, and embodiments such as *bzip* are quickly becoming popular. As shown in [8], the BWT output sequence of a tree source is close to a piecewise stationary memoryless source. This property is exploited in our algorithm to allow estimation of the entropy without the need to know the memory structure of the sources.

We establish an upper bound on the convergence rate for finite memory sources with the property that the conditional probabilities of a symbol, given all the past observations, depend only on no more than a fixed number $d$ of contiguous past observations. Hence such sources can be viewed as $d$th-order Markov sources. However, a Markov model with $|\chi|^d$ states (where $|\chi|$ is the alphabet size) may not be particularly efficient as there may be equivalent states having identical conditional probabilities. This problem has been addressed in [33], where tree sources were introduced. Tree sources are finite-alphabet, finite-memory sources, defined by: i) an alphabet set $\chi$; ii) a finite set of states $S$, that is, a complete and suffix-free subset of sequences of maximal length $d$; and iii) conditional probabilities for each state. The current state is a unique suffix of the last $d$ symbols that is in $S$. In a tree representation, each branch is labeled with a symbol from the alphabet set; the sequence from a leaf to the root is the past observations (with the symbol closest to the root being the last observation). Therefore, there is a one-to-one correspondence from the states to leaf nodes, which may have different depths.

The following notation and definitions are used throughout the correspondence. Denote the realization of length $n$ by $\boldsymbol{z} = z^n$, and the probability measures of the source by $q$. The entropy rate can be expressed as

$$H(q) = \lim_{n\to\infty} \frac{1}{n} E\left[\log_2\left(\frac{1}{q(Z^n)}\right)\right] \tag{1}$$

$$= \lim_{n\to\infty} \frac{1}{n} \sum_{\alpha^n \in \chi^n} q(\alpha^n) \log_2\left(\frac{1}{q(\alpha^n)}\right) \tag{2}$$

$$= \lim_{n\to\infty} \left(-\frac{1}{n}\log_2 q(Z^n)\right) \quad \text{a.s.} \tag{3}$$

Equation (1) is the definition of entropy rate. Equation (3) is the Shannon–McMillan–Breiman theorem, and it holds for all stationary ergodic sources in the sense of almost-sure convergence. In the case of finite memory sources which have the Markov property, we have

$$H(q) = \sum_{s\in S} q(s) \sum_{\alpha\in\chi} q(\alpha|s) \log_2\left(\frac{1}{q(\alpha|s)}\right). \tag{4}$$

This immediately suggests an empirical distribution plug-in scheme for entropy estimation in the case where the source memory length is known.

The rest of the paper is organized as follows. In Section II, we present our algorithm for entropy estimation. Convergence results are proved in Section III. We also develop an alternative entropy estimator based on data compression algorithms that use statistical context modeling in Section IV. Finally, experimental results are presented in Section V. Experiments on randomly generated binary tree sources show that the speed of convergence of our algorithm is much faster than that of the LZ-based methods.

## II. Algorithm

### A. The Burrows–Wheeler Transform (BWT)

*1) Description of the BWT:* The BWT is a reversible block-sorting algorithm [5]. It operates on a sequence of $n$ symbols, produces all $n$ cyclic shifts of the original sequence, sorts them lexicographically, and outputs the last column of the sorted table as well as the location of the original sequence in the sorted table. Alternatively, a unique "end-of-file" symbol— "$" is appended, and the sorting assumes that "$" is the last symbol in the alphabet set. Then the BWT of the original sequence is the last column of the sorted table. This transform is uniquely invertible. The location of the "$" indicates the location of the original sequence in the sorted table. For example, if the original data sequence is "**banana**," then the cyclic shifts and the alphabetically sorted table are shown in Fig. 1.

By sorting the rows alphabetically, cyclic shifts that begin with the same sequence are grouped together, and the BWT outputs the symbol that precedes the sequence in each row. Thus, performing the BWT on the reversed data sequence groups together symbols with a common past context. For finite-memory sources, this process groups together symbols in the same state, and with the same conditional distribution.

*2) Output Distribution of the BWT:* The output distribution of the BWT was studied in [8], [31]. It was shown that if the input is stationary ergodic, then the normalized divergence between the output distribution and a piecewise stationary memoryless distribution vanishes as the length of the input sequence goes to infinity.

The properties of the BWT output distribution from [8], [31] suggest an efficient way to estimate entropy. Note that if the sources are known to be i.i.d., we can simply estimate the distribution empirically and plug the estimates into the formulas for entropy. What if the sources are known to be piecewise i.i.d.? We could locate the transitions and apply the method for i.i.d. sources to each segment as shown in Fig. 2.

### B. New Entropy Estimator

A basic task of our entropy estimator is to estimate conditional empirical distributions. The "context sorting" properties of the BWT discussed in Section II-A suggest a method to estimate these conditional distributions based on segmentation of the BWT output.

As shown in Fig. 3, our algorithm has four steps.

a) Run the BWT on the reversed sequence. Note that in contrast with applications of the BWT in data compression, we do not need to perform the inverse BWT. Hence, the use of the end-of-file character "$" (or the pointer to the original sequence in the sorted table) is not necessary.

b) Partition the BWT output sequence into $T_z$ segments. The segments need not have equal length. Segmentation strategies are discussed in Section II-C. As we will see, the uniform segmentation strategy where the segments have equal length is often quite effective.

c) Estimate the first-order distribution within each segment. We denote the number of occurrences of symbol $a$ in the $j$th segment by $N_j(a)$, and the probability estimate of symbol $a$ in the $j$th segment by $\hat{q}(a,j)$

$$\hat{q}(a,j) = \frac{N_j(a)}{\sum_{b\in\chi} N_j(b)}. \tag{5}$$

The contribution to the entropy estimate of the empirical distribution in the $j$th segment (5) is summarized by

$$\log_2 \hat{q}(j) = \sum_{a\in\chi} N_j(a) \log_2 \hat{q}(a,j). \tag{6}$$

d) Average the individual estimates. The estimate for entropy is

$$\hat{H}(z^n) = -\frac{1}{n}\sum_{j=1}^{T_z} \log_2 \hat{q}(j). \tag{7}$$

### C. Approaches to Segmentation

Segmentation of the BWT output is required in the entropy estimator. In this section we describe two approaches. The intuition behind our estimator suggests that we want the segmentation to align with the piecewise i.i.d. structure of the BWT output. However, these transition points depend on the unknown memory structure of the underlying source. A natural approach is to base the segmentation on the empirical distributions of the BWT output to try to detect the transitions. The adaptive segmentation method described in the following is along these lines. However, from experimental results, a very simple uniform segmentation method performs almost as well as the more complex adaptive method in most cases.

*1) Uniform Segmentation:* In uniform segmentation, we simply partition the BWT output so that each segment contains an equal number of symbols from the sequence according to which we are segmenting. Thus, uniform segmentation according to $\boldsymbol{z}$ means that each segment has the same number of symbols from $\boldsymbol{z}$, which we will denote by $w(n)$.

There is a fundamental tradeoff in the choice of the number of symbols $w(n)$. The segments should be chosen such that those segments containing transitions are negligible, so we cannot choose the segment length $w(n)$ too large. On the other hand, if $w(n)$ is too small, then the probability estimates in each segment will be less accurate. Although averaging over the segments will mitigate this disadvantage to some extent, there is a bias in the estimation in each segment which also plays a role and prohibits too small a segment length. As $n \to \infty$, we need $w(n) \to \infty$, but not too rapidly so that the number of segments $k(n) = n/w(n) \to \infty$. Taking $w(n)$ to grow as $\sqrt{n}$ is shown in Section III to be a balanced choice.

*2) Adaptive Segmentation:* For a given tree source, the number of states is fixed. But in the uniform segmentation scheme above, the number of equal-length segments $k(n) \to \infty$ as $n$ grows. Thus, we end up with many more segments than the actual number of states, even though ideally each segment should represent a different state. To improve the performance of our estimator, an adaptive segmentation scheme is described here that attempts to avoid unnecessary segmentation of the BWT output sequence.

The adaptive algorithm estimates the location of the transitions of the BWT output sequence based on the empirical distribution of the symbols. The specific algorithm we use for adaptive segmentation is inspired by the one introduced in [25]. The algorithm uses a two-level hierarchical scheme to first obtain rough estimates for transition locations, followed by a second pass that refines the locations of the estimates. In both passes, decisions are made based on local statistics of the sequence. In this way, the number and lengths of the segments are adapted to the realization from the source, and this generally results in the BWT output sequence being divided into segments of different lengths.

In the first pass, the sequence is partitioned into level-1 blocks of length $k_1$. The choice of $k_1$ will be given below. For each level-1 block of length $k_1$ (except the two ends), we compute a metric (defined below) to decide whether a transition occurs in the neighborhood of this block. The metric compares statistics in the two neighboring blocks on either side of the block currently under consideration. All those level-1 blocks for which the metric exceeds a threshold are marked for further consideration.

After all level-1 blocks have been processed for marking, we successively retain a subset of the marked blocks as follows. We keep the marked block that gives the largest value of the metric. Then we eliminate any marked blocks that are within two blocks of this one. Among the remaining marked blocks, we then select the one that gives the next largest value of the metric. We keep this block and again eliminate any marked blocks within two blocks of this one. We repeat this procedure until all the marked blocks have either been selected or eliminated. The marked blocks that have been selected will each give rise to one transition in either the selected block itself or in one of its two neighbors. The location of the transition will be determined more precisely in the second pass. This completes the first pass.

In the second pass, for each marked and selected block from the first pass, we subdivide the block and its two neighboring blocks into subblocks of lengths $k_0$, giving $3k_1/k_0$ subblocks for each selected block. The choice for $k_0$ will be given below. A transition will be placed in the center of one of these $3k_1/k_0$ subblocks, determined as follows. For each of these subblocks, we compute the metric using the statistics in the two neighboring subblocks of length $k_0$. The transition is placed in whichever of the $3k_1/k_0$ subblocks gives the largest value of the metric. The segments are defined as the regions of the sequence between successive transitions.

Let $M_l(j)$ denote the value of the metric computed for the $j$th block/subblock in level-$l$. The metric used in both passes is given by (see Fig. 4)

$$M_l(j) \triangleq H_l(j-1, j+1) - \frac{1}{2}H_l(j-1) - \frac{1}{2}H_l(j+1) \tag{8}$$

where $H_l$ is the empirical entropy for the level-$l$ block

$$H_l(j) = -\sum_{a\in\chi} \frac{N_j(a)}{k_l} \log_2 \frac{N_j(a)}{k_l} \tag{9}$$

and

$$H_l(j,i) = -\sum_{a\in\chi} \frac{N_j(a)+N_i(a)}{2k_l} \log_2 \frac{N_j(a)+N_i(a)}{2k_l}. \tag{10}$$

This metric is shown to be an asymptotically optimal statistic for testing whether or not two blocks emerged from the same source [10], [39]. Due to the concavity of entropy as a function of probability, $M_l(j)$ defined in (8) is nonnegative. A large value of $M_l(j)$ indicates that a transition occurs near the level-$l$ block $j$. The number of segments $T$ depends on the threshold, the source, and the choice of $k_1$ and $k_0$. Normally, the threshold will be chosen depending on the length of the level-1 blocks $k_1$.

The parameters $k_1$ and $k_0$ should be chosen such that $k_1 = o(n)$ and $k_0 = o(k_1)$. In our scheme, $k_1$ is chosen to be on the order of $(\log_2 n)^{1+\mu}$, and $k_0$ on the order of $(\log_2 \log_2 n)^{1+\nu}$. Typical values for the design parameters $\mu, \nu > 0$ in our implementation are between 2 and 3.

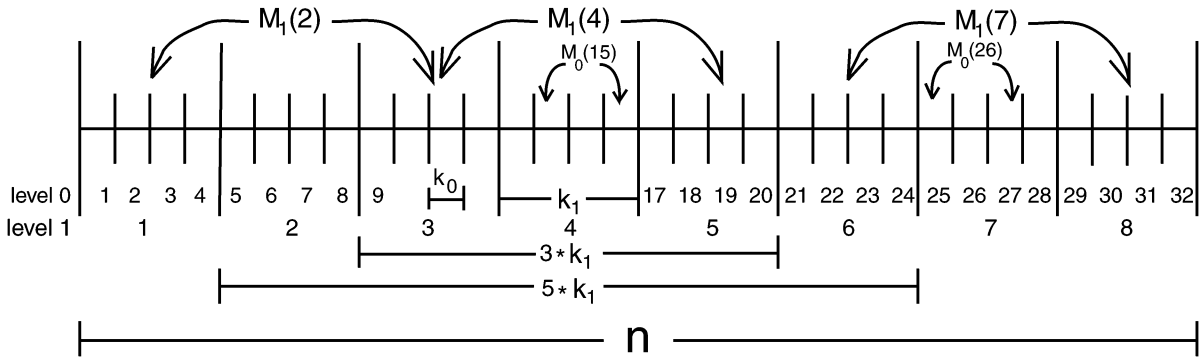


Fig. 4. Two-level block structure.

## III. Performance Analysis

In this section, we analyze the convergence of our estimator with uniform segmentation and obtain conditions on the growth of segment length for our algorithm. We prove almost-sure convergence of our entropy estimator for any stationary ergodic sources and determine the optimal segment length for Markov sources in the sense of maximizing the speed at which the mean-square error goes to zero as the sequence length goes to infinity. For finite-state Markov sources with known states, we can use the plug-in estimator. However, plug-in methods have difficulty in dealing with sources with unknown memory length. Our basic strategy is to use the BWT followed by uniform segmentation and probability estimation within each segment. We prove that our method works for general stationary ergodic sources, and also obtain an upper bound on the convergence rate for finite-state Markov sources when the order is unknown.

### A. Entropy Estimator With Uniform Segmentation for Stationary Ergodic Sources

In the main theorem, we prove almost-sure convergence of our entropy estimator for any stationary ergodic source. Since the entropy is bounded by $\log_2|\chi|$, almost-sure convergence implies convergence in mean square.

The proof consists of two parts. In the first part, since the BWT sorts the sequence by context, and the entropy is concave, the $\limsup$ of our entropy estimate is upper-bounded by the conditional entropy of any order. In the second part, we argue that the BWT is a reversible transform, so we can design a data compression algorithm via the BWT and uniform segmentation. However, the fact that it cannot compress the source below its entropy leads to a lower bound on the $\liminf$ of our entropy estimate.

*Theorem 1:* Let $\boldsymbol{z}$ be a sequence of length $n$ generated from a stationary ergodic source $q$. Our entropy estimator using uniform segmentation with segment length $w(n) = c\cdot n^{\gamma}$ (where $0 < \gamma < 1$) converges to the entropy rate with probability one

$$\lim_{n\to\infty} \hat{H}_n(q) = H(q) \quad \text{a.s.} \tag{11}$$

*Proof:* Let $H_m(q) = H(Z_m|Z_0^{m-1})$ be the $m$th-order conditional entropy. $H_m(q)$ is decreasing in $m$, and $H_m(q) \to H(q)$, as $m \to \infty$. Thus, for any $\epsilon > 0$, there exists $m$, such that

$$H(q) \le H_m(q) \le H(q) + \epsilon. \tag{12}$$

$H_m(q)$ is the entropy of the $m$th-order Markov approximation of the source, which has $|\chi|^m$ states. Bad segments are defined as those segments of the $w(n)$-uniform segmentation, which contain a transition of the $m$th-order contexts of the BWT output. At most there can be $|\chi|^m - 1$ bad segments. Let $\hat{H}'_{n,m}(q)$ be a fictitious estimator that after uniform segmentation throws away bad segments and fuses all segments in between discarded segments. The estimator then proceeds within the fused super-segments in the usual fashion. So the difference of $\hat{H}'_{n,m}(q)$ and $\hat{H}_n(q)$ is that $\hat{H}_n(q)$ is the average of entropy estimates in finer segments and without discarding bad segments. The contribution of the bad segments to $\hat{H}_n(q)$ (which is discarded by $\hat{H}'_{n,m}(q)$) is no more than $C_0 w(n)|\chi|^m/n$, which is asymptotically negligible. Here $C_0 = \log_2|\chi|$ is a constant independent of $n$ and $m$. Furthermore, due to the fact that entropy is concave, the entropy estimate from a fused segment is larger or equal to the average of entropy estimates from those single segments making up the fused segment. We have

$$\hat{H}_n(q) \le \hat{H}'_{n,m}(q) + \frac{C_0 w(n)|\chi|^m}{n}. \tag{13}$$

The fictitious estimator $\hat{H}'_{n,m}(q)$ is close to $\tilde{H}_{n,m}(q)$, which is the empirical estimator of order $m$, since the discarded segments of length $w(n)$ are asymptotically negligible. Let $N_n(b^m)$ be the number of occurrences of $b^m \in \chi^m$ in the sequence $\boldsymbol{z}$ of length $n$, and $N_n(a,b^m)$ be the number of occurrences of $b^m$ followed immediately by $a \in \chi$. $N'_n(b^m)$ and $N'_n(a,b^m)$ are the counterparts of $N_n(b^m)$ and $N_n(a,b^m)$ in the fictitious estimator $\hat{H}'_n(q)$, where a symbol is not counted if it happens to fall into a bad segment

$$\begin{aligned}
\tilde{H}_{n,m}(q) &= -\sum_{b^m\in\chi^m}\sum_{a\in\chi}\frac{N_n(a,b^m)}{n}\log_2\frac{N_n(a,b^m)}{N_n(b^m)} \\
&= -\sum_{b^m\in\chi^m}\sum_{a\in\chi}\frac{N_n(a,b^m)}{n}\log_2\frac{N_n(a,b^m)}{n} \\
&\quad + \sum_{b^m\in\chi^m}\frac{N_n(b^m)}{n}\log_2\frac{N_n(b^m)}{n} \\
\hat{H}'_{n,m}(q) &= -\sum_{b^m\in\chi^m}\sum_{a\in\chi}\frac{N'_n(a,b^m)}{n}\log_2\frac{N'_n(a,b^m)}{N'_n(b^m)} \\
&= -\sum_{b^m\in\chi^m}\sum_{a\in\chi}\frac{N'_n(a,b^m)}{n}\log_2\frac{N'_n(a,b^m)}{n} \\
&\quad + \sum_{b^m\in\chi^m}\frac{N'_n(b^m)}{n}\log_2\frac{N'_n(b^m)}{n}
\end{aligned}$$

where the difference of the counts $N_n(\cdot)$ and $N'_n(\cdot)$ comes from the discarded bad segments on the two ends of a state, which is no more than $2w(n)$. That is,

$$\begin{aligned}
N_n(a,b^m) - 2w(n) \le N'_n(a,b^m) \le N_n(a,b^m) \\
N_n(b^m) - 2w(n) \le N'_n(b^m) \le N_n(b^m).
\end{aligned}$$

By the ergodic theorem, we have

$$\begin{aligned}
\frac{N_n(b^m)}{n} &\to q(b^m) \quad \text{a.s.} \\
\frac{N_n(a,b^m)}{n} &\to q(a,b^m) \quad \text{a.s.}
\end{aligned}$$

Since $w(n)/n \to 0$, we have

$$\frac{N'_n(b^m)}{n} \to q(b^m) \quad \text{a.s.}$$
$$\frac{N'_n(a,b^m)}{n} \to q(a,b^m) \quad \text{a.s.}$$

Therefore, we have

$$\hat{H}'_{n,m}(q) \to H_m(q) \quad \text{a.s.} \tag{14}$$

where

$$H_m(q) = -\sum_{b^m \in \chi^m} \sum_{a \in \chi} q(a,b^m) \log_2 q(a,b^m) + \sum_{b^m \in \chi^m} q(b^m) \log_2 q(b^m).$$

Now $\hat{H}'_{n,m}(q)$ converges almost surely to $H_m(q)$, which is the entropy of the $m$th-order Markov approximation of the stationary ergodic source. Combining (12)–(14), we have

$$\limsup_{n\to\infty} \hat{H}_n(q) \le H_m(q) \le H(q) + \epsilon \quad \text{a.s.} \tag{15}$$

for any $\epsilon > 0$. Hence,

$$\limsup_{n\to\infty} \hat{H}_n(q) \le H(q) \quad \text{a.s.} \tag{16}$$

Next, we prove that

$$\liminf_{n\to\infty} \hat{H}_n(q) \ge H(q) \quad \text{a.s.} \tag{17}$$

by using an argument analogous to the sample converse in source coding theory (see [3, Theorem 3.1], [13], [38]) and a connection of the entropy estimate $\hat{H}_n(q)$ with a lossless data compression algorithm.

Let $L_n(Z^n)$ denote the code length (for a prefix code). Let $A_n$ be the event that

$$\left\{ L_n(Z^n) < \log_2 \frac{1}{q(Z^n)} - n\epsilon \right\}.$$

Then, by Kraft's inequality

$$\begin{aligned} \Pr[A_n] &= \Pr\left[ \frac{1}{n} L_n(Z^n) < \frac{1}{n} \log_2 \frac{1}{q(Z^n)} - \epsilon \right] \\ &= \sum_{z^n \in A_n} q(z^n) \\ &\le \sum_{z^n \in A_n} 2^{-L_n(z^n)} 2^{-n\epsilon} \\ &\le 2^{-n\epsilon}. \end{aligned} \tag{18}$$

Next we design a data compression algorithm as follows.[2] For fixed $n$, since the BWT is reversible, we can encode $Z^n$ by encoding its BWT output and an integer between 1 and $n$ (indicating the location of the original sequence in the sorted table). We do the BWT followed by uniform segmentation and symbol counting in each segment. We encode the statistics (the empirical probability for each symbol) using $C \log_2 n$ bits per segment (where $C$ is a constant independent of $n$), $Cn \log_2 n/w(n)$ bits in total, which gives $C \log_2 n/w(n)$ additional cost per symbol; followed by the Shannon–Fano–Elias coding [7] (with a fixed block length $w(n)$) for each segment with the statistics for that segment. For each segment, the Shannon–Fano–Elias coder has a total redundancy of no more than 2 bits over the empirical entropy times segment length. So the redundancy per symbol is less than $2/w(n)$. Overall, the coding length per symbol is less than $\hat{H}_n(q) + (C \log_2 n + 2)/w(n)$. For fixed $n$, the counts encoded for the statistics take fixed amount of bits. The Shannon–Fano–Elias code (with a fixed block length) is a prefix code. Thus, there are no two codewords produced by our coding scheme (for two different sequence of length $n$) such that one is a prefix of the other. It follows from (18) that, for any $\epsilon > 0$

$$\begin{aligned} &\Pr\left[ \hat{H}_n(q) < \frac{1}{n} \log_2 \frac{1}{q(Z^n)} - \epsilon \right] \\ &\quad \le \Pr\left[ \frac{1}{n} L_n(Z^n) < \frac{1}{n} \log_2 \frac{1}{q(Z^n)} - \epsilon + \frac{C \log_2 n + 2}{w(n)} \right] \\ &\quad \le 2^{-n(\epsilon - (C\log_2 n + 2)/w(n))}. \end{aligned} \tag{19}$$

Since $w(n) = cn^\gamma$, where $0 < \gamma < 1$, we choose $\epsilon = 1/n^{\gamma/2}$ in (19) so that we can apply the Borel–Cantelli lemma. It follows that

$$\hat{H}_n(q) < \frac{1}{n} \log_2 \frac{1}{q(Z^n)} - \frac{1}{n^{\gamma/2}}$$

for finitely many $n$ almost surely. By the Shannon–McMillan–Breiman theorem, we have

$$\lim_{n\to\infty} \frac{1}{n} \log_2 \frac{1}{q(Z^n)} = H(q) \quad \text{a.s.}$$

Thus,

$$\liminf_{n\to\infty} \hat{H}_n(q) \ge H(q) \quad \text{a.s.} \tag{20}$$

Combining (16) and (20) gives the almost-sure convergence in (11). □

### *B. Entropy Estimator With Uniform Segmentation for Finite Memory Sources*

In the next theorem, we specialize to finite-state Markov sources, and establish the rate of convergence in the mean square sense. We first prove a lemma on the convergence rate of the empirical estimator when the set of states of the Markov source is known. This is equivalent to the case when we know the exact places of transitions in the BWT output sequence and segment it accordingly. In the following, we always assume stationary Markov sources with finite alphabet and finite order, and the Markov chain is irreducible and aperiodic, with a unique stationary distribution.

*Lemma 1:* Let $\boldsymbol{z}$ be a sequence of length $n$ generated from a finite-state Markov source $q$. Assume the set of states $S$ of the Markov source is known. Denote the number of occurrences of state $s \in S$ by $N_n(s)$, and the number of occurrences of symbol $a \in \chi$ in state $s$ by $N_n(a,s)$. Then the empirical estimator

$$\tilde{H}_n(q) = -\sum_{s\in S} \frac{N_n(s)}{n} \sum_{a\in\chi} \frac{N_n(a,s)}{N_n(s)} \log_2 \frac{N_n(a,s)}{N_n(s)}$$

satisfies

$$E\left[ \left( \tilde{H}_n(q) - H(q) \right)^2 \right] = O\left( \frac{1}{n} \right).$$

*Proof:* The empirical estimator $\tilde{H}_n(q)$ can be written as

$$\tilde{H}_n(q) = -\sum_{s\in S} \sum_{a\in\chi} \frac{N_n(a,s)}{n} \log_2 \frac{N_n(a,s)}{n} + \sum_{s\in S} \frac{N_n(s)}{n} \log_2 \frac{N_n(s)}{n}.$$

[2] A practical algorithm following this idea was originally proposed in [2].

The entropy $H(q)$ can be written as

$$H(q) = -\sum_{s\in S}\sum_{a\in\chi} q(a,s)\log_2 q(a,s) + \sum_{s\in S} q(s)\log_2 q(s).$$

The state sequence $S_i$, $i = 1, 2, \ldots$ is a Markov chain, and

$$N_n(s) = \sum_{i=1}^{n} \mathbf{1}_{\{S_i = s\}}.$$

Notice that $S_i' = (S_i, Z_i)$, $i = 1, 2, \ldots$ also forms a Markov chain. Since both $S$ and $\chi$ are finite, we only need to prove that for a finite-state Markov chain, we have

$$E\left[\left(\frac{N_n(s)}{n}\log_2\frac{N_n(s)}{n} - q(s)\log_2 q(s)\right)^2\right] = O\left(\frac{1}{n}\right). \quad (21)$$

Let

$$\hat{q}(s) = \frac{N_n(s)}{n}.$$

By the Taylor expansion, we have

$$\begin{aligned}\hat{q}(s)\log_2\hat{q}(s) = {} & q(s)\log_2 q(s) \\ & + \frac{1}{\ln 2}(1+\ln q(s))(\hat{q}(s)-q(s)) \\ & + \frac{1}{2\ln 2}\frac{(\hat{q}(s)-q(s))^2}{q(s)} \\ & - \frac{1}{6\ln 2}\frac{(\hat{q}(s)-q(s))^3}{(q(s)(1-\theta_s)+\hat{q}(s)\theta_s)^2} \end{aligned} \quad (22)$$

where $0 \le \theta_s \le 1$. In Appendix A, we shall prove that the moments satisfy $E[\hat{q}(s)-q(s)] = 0$, $E[(\hat{q}(s)-q(s))^2] = O(1/n)$, $E[(\hat{q}(s)-q(s))^3] = O(1/n^2)$, $E[(\hat{q}(s)-q(s))^4] = O(1/n^2)$, and use the Cauchy–Schwarz inequality to deal with the last term in (22). Then, (21) can be proved by plugging in the moments. □

*Theorem 2:* Let $\boldsymbol{z}$ be a sequence of length $n$ generated from a finite-state Markov source $q$. Then, the mean-square error of our entropy estimator using uniform segmentation with segment length $w(n) = c\sqrt{n}$, for some constant $c$, satisfies

$$E\left[\left(\hat{H}_n(q) - H(q)\right)^2\right] = O\left(\frac{\log_2^2 n}{n}\right). \quad (23)$$

*Proof:* We follow the same argument in the proof of Theorem 1. But now we can establish the rate of convergence, since the source is assumed to be a $d$th-order Markov source. Thus, $H(q) = H_d(q)$. Bad segments are segments of the $w(n)$-uniform segmentation, which contain a transition of states of the BWT output. There are at most $|\chi|^d - 1$ bad segments. Let $\hat{H}_n'(q)$ be a fictitious estimator that after uniform segmentation throws away bad segments and fuses all segments in between discarded segments and proceeds with the fused segments. Since the order $d$ is fixed, the subscript $d$ is dropped in $\hat{H}_n'(q)$. We have

$$\hat{H}_n(q) \le \hat{H}_n'(q) + \frac{C_0 w(n)|\chi|^d}{n}. \quad (24)$$

Now we show the rate of convergence of the fictitious estimator $\hat{H}_n'(q)$ for $d$th-order Markov sources. Suppose $\tilde{H}_n(q)$ is the empirical estimator assuming knowing the set of states $S$ (where $|S| = |\chi|^d$). By lemma 1, we have

$$\begin{aligned} E\left[\left(\hat{H}_n'(q) - H(q)\right)^2\right] \le {} & 2E\left[\left(\hat{H}_n'(q) - \tilde{H}_n(q)\right)^2\right] \\ & + 2E\left[\left(\tilde{H}_n(q) - H(q)\right)^2\right] \\ = {} & O\left(\frac{w^2(n)\log_2^2 n}{n^2}\right) + O\left(\frac{1}{n}\right). \end{aligned} \quad (25)$$

Combining (24) and (25), we have

$$\begin{aligned} & E\left[\left(\hat{H}_n(q) - H(q)\right)^2 \mathbf{1}_{\{\hat{H}_n(q)\ge H(q)\}}\right] \\ & \le E\left[\left(\hat{H}_n'(q) - H(q) + \frac{C_0 w(n)|\chi|^d}{n}\right)^2 \mathbf{1}_{\{\hat{H}_n(q)\ge H(q)\}}\right] \\ & = O\left(\frac{w^2(n)\log_2^2 n}{n^2}\right) + O\left(\frac{1}{n}\right). \end{aligned} \quad (26)$$

Next we use the same argument leading to (19), and choose $\epsilon = (C\log_2 n + 3)/w(n)$

$$\Pr\left[\hat{H}_n(q) < \frac{1}{n}\log_2\frac{1}{q(Z^n)} - \frac{C\log_2 n + 3}{w(n)}\right] \le 2^{-n/w(n)}. \quad (27)$$

For Markov sources, we can prove the convergence rate of $-\frac{1}{n}\log_2 q(Z^n)$ to $H(q)$

$$\begin{aligned} & E\left[\left(\frac{1}{n}\log_2\frac{1}{q(Z^n)} - H(q)\right)^2\right] \\ & = E\left[\left(\sum_{s\in S}\sum_{a\in\chi}\left(\frac{N_n(a,s)}{n} - q(s)q(a|s)\right)\log_2\frac{1}{q(a|s)}\right)^2\right] \\ & = O\left(\frac{1}{n}\right) \end{aligned} \quad (28)$$

since the sets $S$ and $\chi$ are finite sets and

$$E\left[\left(\frac{N_n(a,s)}{n} - q(s)q(a|s)\right)^2\right] = O\left(\frac{1}{n}\right)$$

which is proved in Appendix B.

Let $B_n$ be the event that

$$\left\{\hat{H}_n(q) < \frac{1}{n}\log_2\frac{1}{q(Z^n)} - \frac{C\log_2 n + 3}{w(n)}\right\}$$

and $B_n^c$ be the complement of $B_n$. Combining (27) and (28), we have

$$\begin{aligned} & E\left[\left(\hat{H}_n(q) - H(q)\right)^2 \mathbf{1}_{\{\hat{H}_n(q)<H(q)\}}\right] \\ & = E\left[\left(\hat{H}_n(q) - H(q)\right)^2 \mathbf{1}_{\{\hat{H}_n(q)<H(q)\}\bigcap B_n^c}\right] \\ & \quad + E\left[\left(\hat{H}_n(q) - H(q)\right)^2 \mathbf{1}_{\{\hat{H}_n(q)<H(q)\}\bigcap B_n}\right] \\ & \le E\left[\left(H(q) - \frac{1}{n}\log_2\frac{1}{q(Z^n)} + \frac{C\log_2 n + 3}{w(n)}\right)^2 \right. \\ & \qquad \left. \cdot\, \mathbf{1}_{\{\hat{H}_n(q)<H(q)\}\bigcap B_n^c}\right] + (\log_2|\chi|)^2 2^{-\frac{n}{w(n)}} \\ & = O\left(\frac{1}{n}\right) + O\left(\frac{\log_2^2 n}{w^2(n)}\right). \end{aligned} \quad (29)$$

Finally, (23) is obtained by combining (26) and (29) and the choice of segment length $w(n) = c\sqrt{n}$. □

## IV. Connections to Data Compression

Entropy estimation is closely related to data compression. In fact, we can turn a data compression algorithm into an entropy estimator easily. Most state-of-the-art data compression algorithms, such as prediction by partial matching (PPM) [6] and CTW [34], use adaptive statistical methods. The general structure is shown in Fig. 5. We keep the statistics in a context model, which is a tree-like data structure, and we assign a probability estimate to the current symbol according to the current context based on the existing model. Then we update the context model to

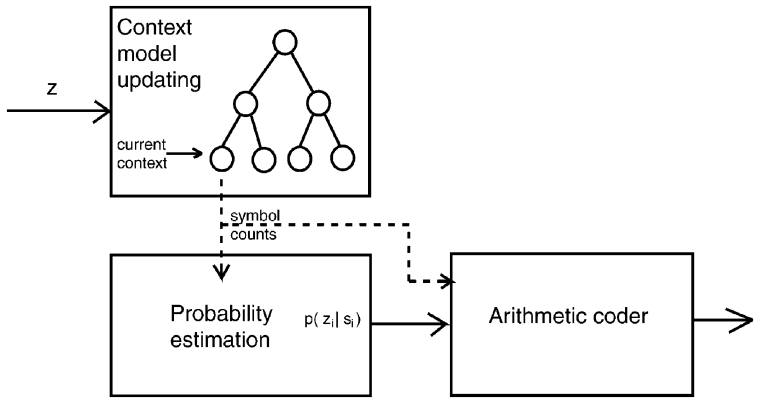


Fig. 5. Statistical methods of data compression.

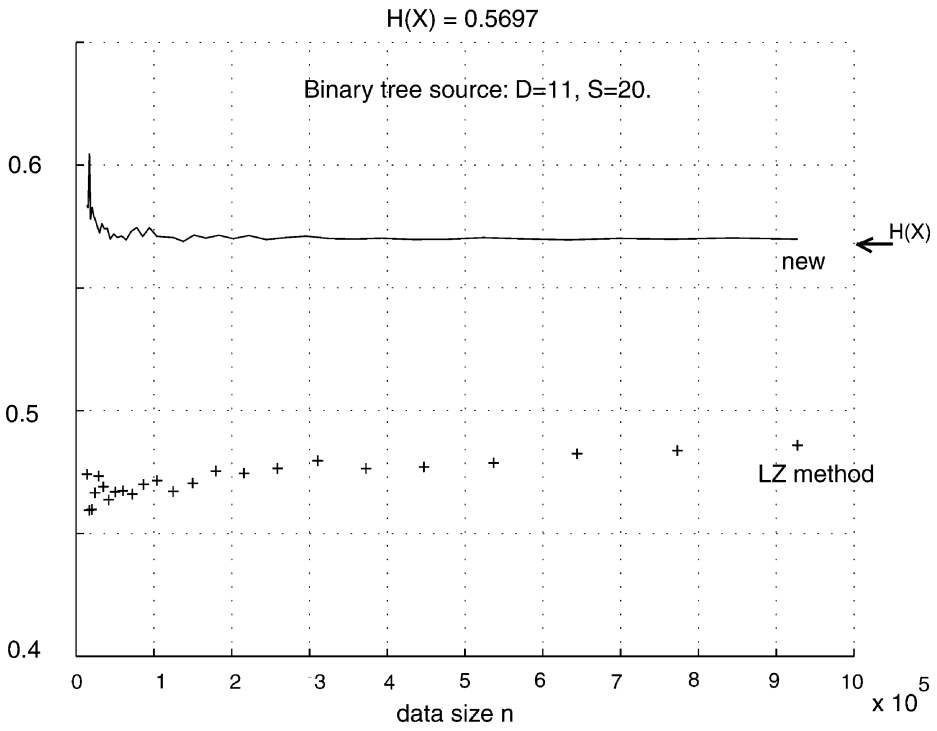


Fig. 6. Comparison of the LZ and the new entropy estimator.

account for the current symbol. For compression, the probability estimates are used in the arithmetic coder. However, instead of using the compression rate to estimate entropy, we could directly use the logarithm of the probability estimates to estimate entropy.

PPM stores statistics in contexts of different orders and uses predictions on the basis of the longest matching context. In addition, PPM can be combined with the context algorithm [32] to select the optimal-order context on which to condition each observed symbol. Recently, a computationally efficient context algorithm has been proposed in [17], which is quite attractive since it can be implemented in linear time. CTW can also be used in entropy estimation. No matter what the actual model (within a model class) is, the weighting strategy [34] guarantees that (the logarithm of) the weighted probability tends to (the logarithm of) the estimated probability according to the actual model, which is used to estimate the entropy.

## V. Experimental Results

In Fig. 6, we compare our new algorithm with the LZ string-matching based algorithm [15], where the entropy estimator is based on the average of longest string-matching lengths as mentioned in Section I. The sources we use are randomly generated binary tree sources, with memory length 11, and 20 states. All the curves plotted are an average of 100 runs. As shown in Fig. 6, the new algorithm converges quite fast to the entropy. In contrast, for the data sizes considered, the LZ based algorithm is not able to offer a good approximation.

In Fig. 7, we compare our new algorithm with the empirical plug-in scheme using Markov models of different orders. The empirical plug-in scheme computes empirical conditional probabilities and stationary probabilities by counting the numbers of symbols emitting from each state and plugs in the formula, assuming the order of the Markov sources is known. Even when the memory length is known, the plug-in scheme does not perform as well as our scheme and suffers degradation either when it underestimates or overestimates the order. Fig. 7 also shows that the performance difference for adaptive and uniform segmentation is insignificant.

In the empirical plug-in scheme, unless prior knowledge is available about the tree structure of the source, the number of transition probabilities to be estimated grows exponentially with the order of the source.

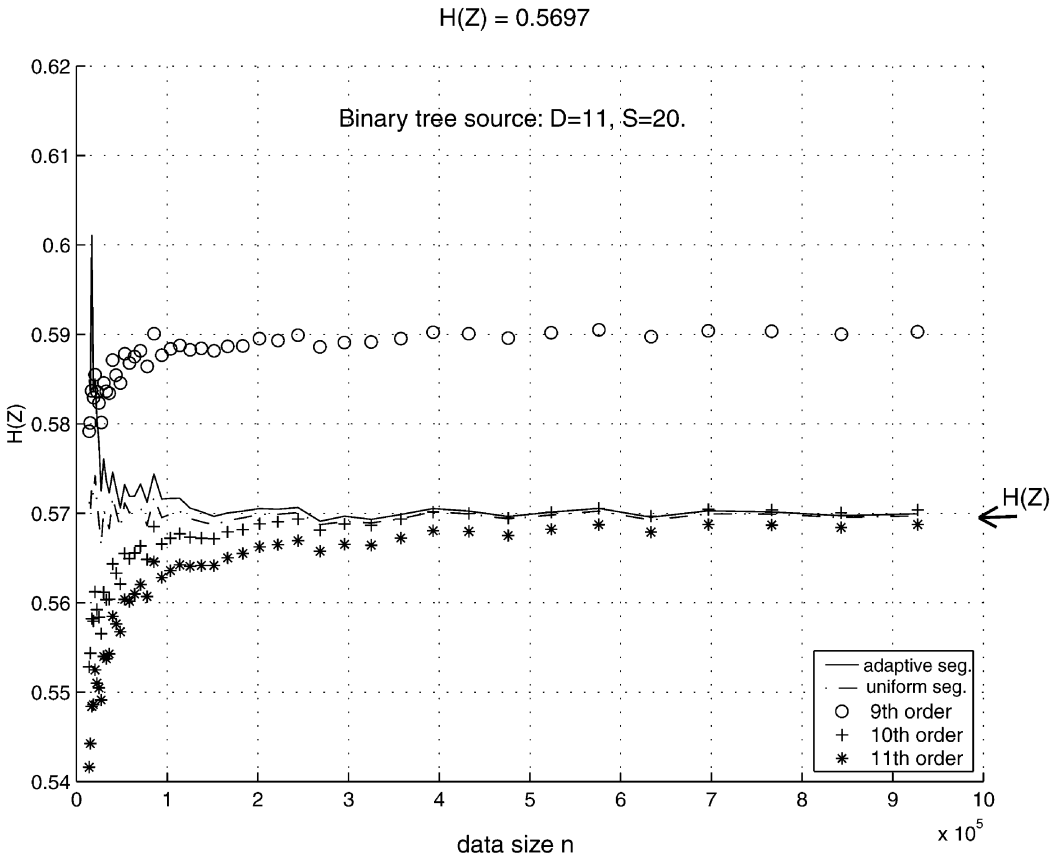


Fig. 7. Entropy estimator based on the BWT.

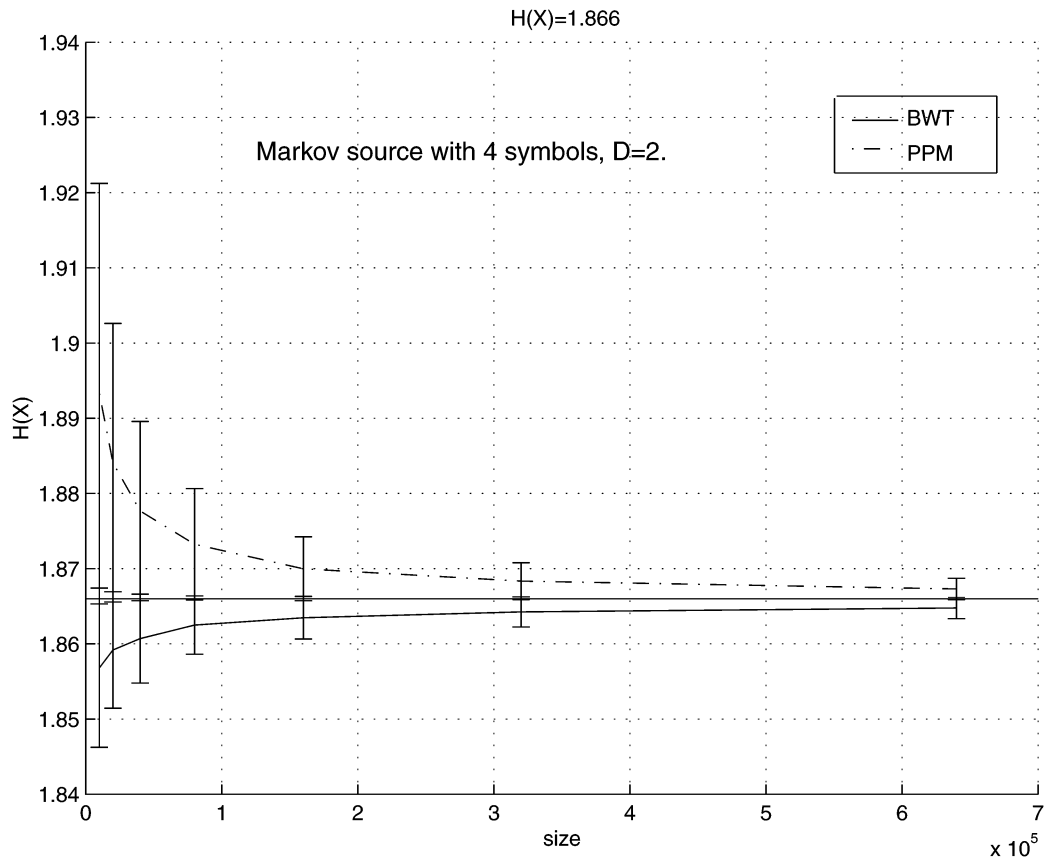


Fig. 8. Entropy estimator based on the BWT/PPM with error bars (sample variance).

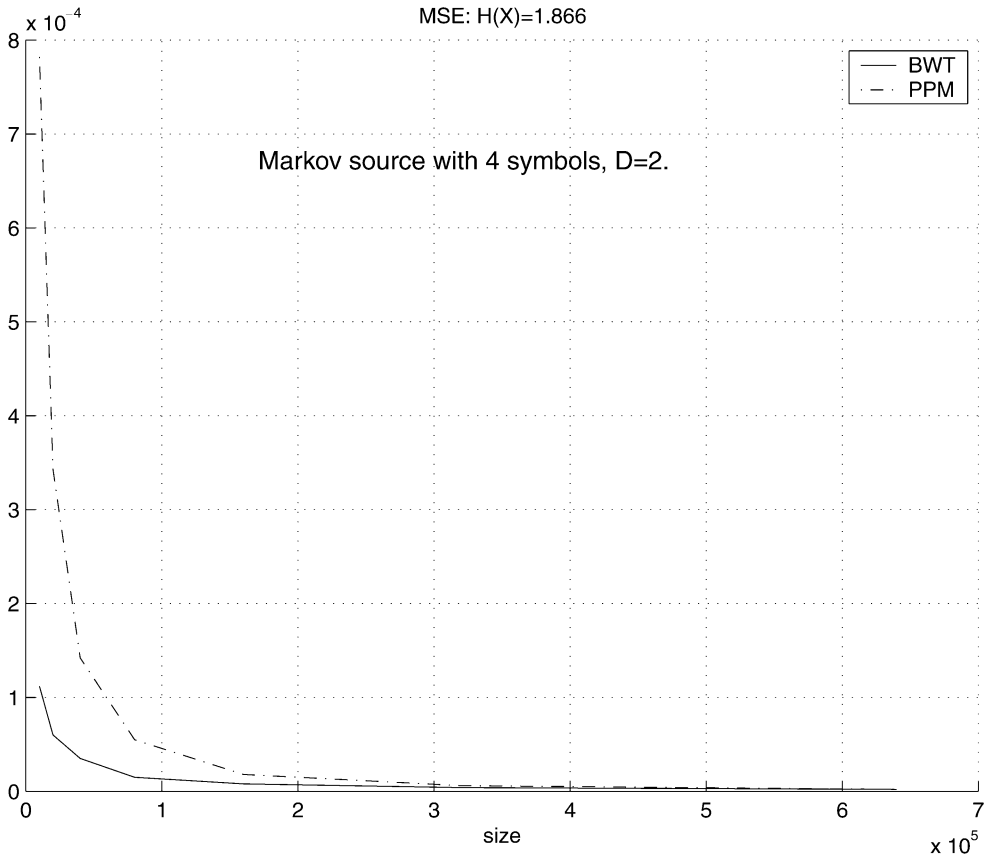


Fig. 9. Mean-square error of entropy estimator based on the BWT/PPM.

Our algorithm has the advantage that it does not require any knowledge of the memory length or the number of states. Hence, it is suitable for unknown tree sources.

In Figs. 8 and 9, we compare the algorithm via the BWT and the algorithm based on the PPM that we outlined in Section IV. In Fig. 8, we plot averaged estimates of 100 runs with the error bar indicating the sample variance $\sigma$ of the simulations. Since we use the probability estimates to estimate entropy in both schemes, neither PPM-based estimation nor BWT-based estimation are actual compression rates. The compression overhead is not included in the estimation and, therefore, the comparison is fair. In addition, we assume the maximum order of the Markov source is known in the PPM-based estimator (although the maximum order is not needed in the unbounded PPM), while the BWT-based estimator does not use that information. Indeed, if the maximum order used in PPM is not less than the order of the Markov source, one can show that the mean square error of the PPM-based estimator converges to zero at the speed of $O(1/n)$, whereas we guarantee the convergence rate $O(\log_2^2 n/n)$ for the BWT-based estimator. In the experiment, both algorithms show similar convergence properties. Even for large block lengths, the BWT-based algorithm performs very well compared to the PPM-based algorithm. It is plausible that a faster convergence rate can be shown for the BWT-based estimator.

## APPENDIX A

We need to calculate the moments $E[(N_n(s) - nq(s))^2]$, $E[(N_n(s) - nq(s))^3]$, and $E[(N_n(s) - nq(s))^4]$, where $N_n(s)$ is the counts of occurrences of state $s$ and $q(s)$ is the stationary distribution of state $s$ of the Markov chain. We follow the approach in [4], using the fact that for an ergodic (irreducible and aperiodic) Markov chain, there exist positive constants $\gamma$ and $\rho < 1$, such that $|q_{st}^{(n)} - q(t)| < \gamma\rho^n$ holds for all state $s$ and $t$ and all $n$, where $q_{st}^{(n)} = \Pr\{S_n = t | S_0 = s\}$ is the $n$th-order transition probabilities. Let $\mathbf{1}_s(i) = \mathbf{1}_{\{S_i = s\}}$ be the indicator function

$$
\begin{aligned}
&E\left[(N_n(s) - nq(s))^2\right] \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{n} E\left[(\mathbf{1}_s(i) - q(s))(\mathbf{1}_s(j) - q(s))\right] \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{n} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\quad = \sum_{i=1}^{n} E\left[\mathbf{1}_s(i) - q^2(s)\right] + 2\sum_{i<j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\quad = n\left(q(s) - q^2(s)\right) + 2q(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) \\
&E\left[(N_n(s) - nq(s))^3\right] \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} E\Big[(\mathbf{1}_s(i) - q(s))(\mathbf{1}_s(j) - q(s)) \\
&\qquad\qquad \cdot (\mathbf{1}_s(k) - q(s))\Big] \\
&\quad = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k) - q^3(s)\right] \\
&\qquad - 3nq(s)\sum_{i=1}^{n}\sum_{j=1}^{n} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\quad = n\left(q(s) - q^3(s)\right) + 6\sum_{i<j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^3(s)\right] \\
&\qquad + 6\sum_{i<j<k} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k) - q^3(s)\right]
\end{aligned}
$$

$$
\begin{aligned}
&- 3nq(s)\sum_{i=1}^{n}\sum_{j=1}^{n} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&= n\left(q(s) - 3q^2(s) + 2q^3(s)\right) \\
&\quad + 6q(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) \\
&\quad - 6nq^2(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) \\
&\quad + 6q(s)\sum_{d+e\le n-1}(n-d-e)\left(q_{ss}^{(d)}q_{ss}^{(e)} - q^2(s)\right) \\
&= n(q(s) - 3q^2(s) + 2q^3(s)) \\
&\quad + 6q(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) \\
&\quad - 6q^2(s)\sum_{d=1}^{n-1}(n-d)(d+1)\left(q_{ss}^{(d)} - q(s)\right) \\
&\quad + 6q(s)\sum_{d+e\le n-1}(n-d-e) \\
&\quad \cdot \left(q_{ss}^{(d)} - q(s)\right)\left(q_{ss}^{(e)} - q(s)\right).
\end{aligned}
$$

Due to the fact that $\sum_{d=1}^{\infty} d^l(q_{ss}^{(d)} - q(s))$ converges absolutely, we have

$$E\left[(N_n(s) - nq(s))^2\right] = O(n) \tag{30}$$

and

$$E\left[(N_n(s) - nq(s))^3\right] = O(n). \tag{31}$$

Next we show

$$E\left[(N_n(s) - nq(s))^4\right] = O(n^2). \tag{32}$$

$$
\begin{aligned}
&E\left[(N_n(s) - nq(s))^4\right] \\
&\quad = \sum_{i,j,k,l} E\Big[(\mathbf{1}_s(i) - q(s))(\mathbf{1}_s(j) - q(s)) \\
&\qquad \cdot (\mathbf{1}_s(k) - q(s))(\mathbf{1}_s(l) - q(s))\Big] \\
&\quad = \sum_{i,j,k,l} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k)\mathbf{1}_s(l) - q^4(s)\right] \\
&\qquad - 4nq(s)\sum_{i,j,k} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k) - q^3(s)\right] \\
&\qquad + 6n^2q^2(s)\sum_{i,j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\quad = -4nq(s) \\
&\qquad \cdot \sum_{i,j,k} E\left[(\mathbf{1}_s(i) - q(s))(\mathbf{1}_s(j) - q(s))(\mathbf{1}_s(k) - q(s))\right] \\
&\qquad - 6n^2q^2(s)\sum_{i,j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\qquad + n(q(s) - q^4(s)) + 14\sum_{i<j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^4(s)\right] \\
&\qquad + 36\sum_{i<j<k} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k) - q^4(s)\right] \\
&\qquad + 24\sum_{i<j<k<l} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k)\mathbf{1}_s(l) - q^4(s)\right] \\
&\quad = -4nq(s) \\
&\qquad \cdot \sum_{i,j,k} E\left[(\mathbf{1}_s(i) - q(s))(\mathbf{1}_s(j) - q(s))(\mathbf{1}_s(k) - q(s))\right] \\
&\qquad + 14\sum_{i<j} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j) - q^2(s)\right] \\
&\qquad + 36\sum_{i<j<k} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k) - q^3(s)\right] \\
&\qquad + n\left(q(s) - q^4(s)\right) + 7n(n-1)\left(q^2(s) - q^4(s)\right) \\
&\qquad + \left(-18n^2 + 12n\right)\left(q^3(s) - q^4(s)\right) \\
&\qquad - 12n^2q^3(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) \\
&\qquad + 24\sum_{i<j<k<l} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k)\mathbf{1}_s(l) - q^4(s)\right] \\
&\quad = 24\sum_{i<j<k<l} E\left[\mathbf{1}_s(i)\mathbf{1}_s(j)\mathbf{1}_s(k)\mathbf{1}_s(l) - q^4(s)\right] \\
&\qquad - 12n^2q^3(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) + O(n^2) \\
&\quad = 24q(s)\sum_{d+e+f\le n-1}(n-d-e-f) \\
&\qquad \cdot \left(q_{ss}^{(d)}q_{ss}^{(e)}q_{ss}^{(f)} - q^3(s)\right) \\
&\qquad - 12n^2q^3(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) + O(n^2) \\
&\quad = 24q(s)\sum_{d+e+f\le n-1}(n-d-e-f)\left(q_{ss}^{(d)} - q(s)\right) \\
&\qquad \cdot \left(q_{ss}^{(e)} - q(s)\right)\left(q_{ss}^{(f)} - q(s)\right) \\
&\qquad + 36q^2(s)\sum_{d+e\le n-2}(n-d-e)(n-d-e-1) \\
&\qquad \cdot \left(q_{ss}^{(d)} - q(s)\right)\left(q_{ss}^{(e)} - q(s)\right) \\
&\qquad + 12q^3(s)\sum_{d=1}^{n-3}((n-d-1)^3 - (n-d-1)) \\
&\qquad \cdot \left(q_{ss}^{(d)} - q(s)\right) \\
&\qquad - 12n^2q^3(s)\sum_{d=1}^{n-1}(n-d)\left(q_{ss}^{(d)} - q(s)\right) + O(n^2) \\
&\quad = O(n^2).
\end{aligned}
$$

Finally, we show that

$$R_{i,j}(s) = E\left[\frac{(\hat{q}(s) - q(s))^i}{(q(s)(1-\theta_s) + \hat{q}(s)\theta_s)^j}\right] \tag{33}$$

is at most $O(1/n)$, where $i \ge 3$, $j \ge 2$ and $\hat{q}(s) = N_n(s)/n$. Notice that in the Taylor expansion (22), if $\hat{q}(s) = 0$, then $\theta_s = 1 - 1/\sqrt{3}$. Define event $A = \{\hat{q}(s) > 0\}$ and $A^c = \{\hat{q}(s) = 0\}$. By Cauchy–Schwarz inequality, we have

$$
\begin{aligned}
&\left(E\left[\frac{(\hat{q}(s) - q(s))^i}{(q(s)(1-\theta_s) + \hat{q}(s)\theta_s)^j}\right]\right)^2 \\
&\quad \le E\left[(\hat{q}(s) - q(s))^{2i}\right] E\left[\frac{1}{(q(s)(1-\theta_s) + \hat{q}(s)\theta_s)^{2j}}\right] \\
&\quad \le E\left[(\hat{q}(s) - q(s))^{2i}\right] \\
&\qquad \cdot \left(\frac{3^j \Pr[A^c]}{q^{2j}(s)} + E\left[\mathbf{1}_A \frac{1}{(q(s)(1-\theta_s) + \hat{q}(s)\theta_s)^{2j}}\right]\right) \\
&\quad \le E\left[(\hat{q}(s) - q(s))^{2i}\right] \\
&\qquad \cdot \left(\frac{3^j \Pr[A^c]}{q^{2j}(s)} + E\left[\mathbf{1}_A\left(\frac{1}{q^{2j}(s)} + \frac{1}{\hat{q}^{2j}(s)}\right)\right]\right) \\
&\quad = O\left(\frac{1}{n^2}\right).
\end{aligned} \tag{34}
$$

## APPENDIX B

We need to prove that

$$E\left[\left(\frac{N_n(a,s)}{n} - q(s)q(a|s)\right)^2\right] = O\left(\frac{1}{n}\right). \tag{35}$$

Notice that $S_1', S_2', \ldots$ form a Markov chain with the stationary distribution $q(a,s) = q(s)q(a|s)$, where $S_i' = (S_i, Z_{i+1})$, $i = 1, 2, \ldots$

$$N_n(a,s) = \sum_{i=1}^{n-1} \mathbf{1}_{\{S_i = s, Z_{i+1} = a\}} = \sum_{i=1}^{n-1} \mathbf{1}_{\{S_i' = (s,a)\}}.$$

Then, (35) follows from Appendix A.

## ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments. The authors are also grateful to the Guest Editor En-hui Yang for his helpful suggestions.

## REFERENCES

[1] A. Antos and I. Kontoiannis, "Convergence properties of functional estimates of discrete distributions," *Random Structures and Algorithms*, vol. 19, pp. 163–193, Oct. 2002.

[2] D. Baron and Y. Bresler, "Linear complexity MDL universal coding with the BWT," presented at the IEEE International Symposium on Information Theory, Washington, DC, June 2001.

[3] A. R. Barron, "Logically smooth density estimation," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1985.

[4] P. Billingsley, "Statistical methods in Markov chains," *Ann. Math. Statist.*, vol. 32, no. 1, pp. 12–40, Mar. 1961.

[5] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Digital Systems Res. Ctr., Palo Alto, CA, Tech. Rep. SRC 124, May 1994.

[6] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun. Technol.*, vol. COM-32, pp. 396–402, Apr. 1984.

[7] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[8] M. Effros, K. Visweswariah, S. R. Kulkarni, and S. Verdú, "Universal lossless source coding with the Burrows Wheeler transform," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1061–1081, May 2002.

[9] Y. Gao, I. Kontoiannis, and E. Bienenstock, "Estimating the entropy rate of spike trains," preprint, 2004.

[10] M. Gutman, "Asymptotically optimal classification for multiple tests with empirically observed statistics," *IEEE Trans. Inform. Theory*, vol. 35, pp. 401–408, Mar. 1989.

[11] A. O. Hero and O. Michel, "Asymptotic theory of greedy approximations to minimal k-point random graphs," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1921–1938, Sept. 1999.

[12] A. O. Hero, B. Ma, O. Michel, and J. Gorman, "Applications of entropic spanning graphs," *IEEE Signal Processing Mag.*, vol. 19, pp. 85–95, Sept. 2002.

[13] J. C. Kieffer, "Sample converses in source coding theory," *IEEE Trans. Inform. Theory*, vol. 37, pp. 263–268, Mar. 1991.

[14] J. C. Kieffer and E.-h. Yang, "Grammar based codes: A new class of universal lossless source codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 737–754, May 2000.

[15] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner, "Nonparametric entropy estimation for stationary processes and random fields, with applications to english text," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1319–1327, May 1998.

[16] J. Lanctot, M. Li, and E.-H. Yang, "Estimating DNA sequence entropy," in *Proc. Symp. Discrete Algorithms*, San Francisco, CA, Jan. 2000, pp. 409–418.

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

[18] E. M. Mccreight, "A space-economical suffix tree construction algorithm," *J. Assoc. Comput. Mach.*, vol. 23, pp. 262–272, 1976.

[19] I. Nemenman, W. Bialek, and R. Steveninck, "Entropy and information in neural spike trains: Progress on the sampling problem," *Phys. Rev. E*, to be published.

[20] D. S. Ornstein and B. Weiss, "Entropy and data compression schemes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 78–83, Jan. 1993.

[21] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, pp. 1191–1253, 2003.

[22] A. N. Quas, "An entropy estimation for a class of infinite alphabet processes," *Theory Probab. Appl.*, vol. 43, pp. 496–507, 1995.

[23] A. O. Schmitt and H. Herzel, "Estimating the entropy of DNA sequences," *J. Theor. Biol.*, vol. 188, no. 3, pp. 369–377, 1997.

[24] J. Seward, "On the performace of BWT sorting algorithms," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2000, p. 173.

[25] G. I. Shamir and D. J. Costello, Jr, "Asymptotically optimal low-complexity sequential lossless coding for piecewise-stationary memoryless sources—Part I: The regular case," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2444–2467, Nov. 2000.

[26] C. E. Shannon, "Prediction and entropy of printed english," *Bell Syst. Tech. J.*, pp. 50–64, 1951.

[27] P. C. Shields, "Entropy and prefixes," *Ann. Probab.*, vol. 20, no. 1, pp. 403–409, Jan. 1992.

[28] ——, "The ergodic theory of discrete sample paths," in *Graduate Studies in Mathematics* Providence, RI, 1996, vol. 13.

[29] S. P. Strong, R. Koberle, R. Steveninck, and W. Bialek, "Entropy and information in neural spike trains," *Phys. Rev. Lett.*, vol. 80, pp. 197–200, 1998.

[30] W. Szpankowski, "Asymptotic properties of data compression and suffix trees," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1647–1659, Sept. 1993.

[31] K. Visweswariah, S. R. Kulkarni, and S. Verdú, "Output distribution of the Burrows-Wheeler transform," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, June 2000, p. 53.

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

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

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

[35] A. D. Wyner and J. Ziv, "Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1250–1258, Nov. 1989.

[36] A. J. Wyner and D. Foster, "On the lower limits of entropy estimation," *IEEE Trans. Inform. Theory*, submitted for publication.

[37] E.-h. Yang and J. C. Kieffer, "Efficient universal lossless data compression algorithms based on a greedy sequential grammar transform—Part one: Without context models," *IEEE Trans. Inform. Theory*, vol. 46, pp. 755–788, May 2000.

[38] E.-H. Yang and S. Shen, "Chaitin complexity, Shannon information content of a single event and infinite random sequences (i)," *Science in China*, ser. A, vol. 34, pp. 1183–1193, 1991.

[39] J. Ziv, "On classification with empirically observed statistics and universal data compression," *IEEE Trans. Inform. Theory*, vol. 34, pp. 278–286, Mar. 1988.