Universal Divergence Estimation for Finite-Alphabet Sources

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

***Abstract*—This paper studies universal estimation of divergence from the realizations of two unknown finite-alphabet sources. Two algorithms that borrow techniques from data compression are presented. The first divergence estimator applies the Burrows–Wheeler block sorting transform to the concatenation of the two realizations; consistency of this estimator is shown for all finite-memory sources. The second divergence estimator is based on the Context Tree Weighting method; consistency is shown for all sources whose memory length does not exceed a known bound. Experimental results show that both algorithms perform similarly and outperform string-matching and plug-in methods.**

***Index Terms*—Block sorting, Burrows–Wheeler transform, context tree weighting method, divergence estimation, information divergence, Markov sources, universal methods.**

I. Introduction

ALTHOUGH (Kullback–Leibler) divergence (also known as relative entropy) is a fundamental information measure, special cases of which are mutual information and entropy, the problem of divergence estimation of sources whose distributions are unknown has received relatively little attention.

The sources of interest are finite-alphabet, finite-memory Markov sources, denoted by q_z and p_x. The input to the estimator consists of a realization of length n from source q_z, denoted by $\boldsymbol{z} = z^n$, and a realization of length n from source p_x, denoted by $\boldsymbol{x} = x^n$.

Ziv and Merhav [17] applied the idea of Lempel–Ziv (LZ) parsing to divergence estimation. They developed a scheme to estimate the divergence between two finite-alphabet, finite-order, stationary Markov processes. The LZ incremental parsing algorithm parses the sequence $\boldsymbol{z}$ into $c(z^n)$ distinct phrases such that each phrase is the shortest string which is not a previously parsed phrase. The entropy $H(q_z)$ can be estimated by $\frac{1}{n}c(z^n)\log_2 c(z^n)$. Analogously, $\boldsymbol{z}$ is parsed into $c(z^n||x^n)$ longest phrases, which appear in $\boldsymbol{x}$. The term $-\frac{1}{n}\log_2 p_x(z^n)$ can be estimated by $\frac{1}{n}c(z^n||x^n)\log_2 n$. Then, an estimator of the divergence rate $D(q_z||p_x)$ is given by

$$\frac{1}{n}c(z^n||x^n)\log_2 n - \frac{1}{n}c(z^n)\log_2 c(z^n).$$

Consistency of this estimator is shown in [17] under the assumption that the observations are generated by independent Markov sources, and is universal in the sense of not depending on the order or any other information about the transition probability matrices of the sources.

Another algorithm based on LZ compression was introduced in [2] and applied to problems motivated by linguistics. Unlike [17], the approach in [2] is heuristic and there is no claim that the algorithm converges to the divergence of the sources. The idea is to approximate the divergence $D(q_z||p_x)$ by calculating the additional number of bits per character required to encode the sequence emitted by source q_z with a source code that has been trained by a realization of source p_x. To that end, a realization from source q_z is partitioned into a long sequence $\boldsymbol{z}$ and a shorter sequence $\boldsymbol{z}_0$, which is appended to a long sequence $\boldsymbol{x}$ from source p_x. The new sequence $\boldsymbol{x}+\boldsymbol{z}_0$ is compressed by *gzip*[1] to $L_{\boldsymbol{x}+\boldsymbol{z}_0}$ bits, while $\boldsymbol{x}$ alone is compressed to $L_{\boldsymbol{x}}$ bits. The difference $\Delta_{\boldsymbol{x}\boldsymbol{z}_0} = L_{\boldsymbol{x}+\boldsymbol{z}_0} - L_{\boldsymbol{x}}$ is the coding length of $\boldsymbol{z}_0$ using the coding trained by $\boldsymbol{x}$. Similarly, $\Delta_{\boldsymbol{z}\boldsymbol{z}_0} = L_{z+z_0} - L_{\boldsymbol{z}}$. The divergence between q_z and p_x is approximated by

$$\frac{\Delta_{\boldsymbol{x}\boldsymbol{z}_0} - \Delta_{\boldsymbol{z}\boldsymbol{z}_0}}{|\boldsymbol{z}_0|}$$

where $|\boldsymbol{z}_0|$ is the number of characters of the short sequence $\boldsymbol{z}_0$.

A new class of "normalized information distance" loosely based on the noncomputable notion of Kolmogorov complexity is proposed in [12], and then applied to the genome phylogeny problem and the problem of building language trees considered in [2]. The method in [12] to approximate the measure therein is heuristic (see also the discussion in [11]).

In this paper, we present two divergence estimation algorithms, both of which are motivated by techniques in data compression.[2] The first estimator, originally proposed in [4], uses the Burrows–Wheeler block sorting transform (BWT) [3], while the second estimator uses the Context Tree Weighting method (CTW) [15]. We prove the convergence of our divergence estimators assuming that both sources are possibly dependent stationary ergodic Markov sources, a case for which the following almost-sure convergence result is known to hold [10]:

$$D(q_z\,||\,p_x) = \sum_{s\in\mathcal{S}} q_z(s) \sum_{\alpha\in\chi} q_z(\alpha\,|\,s)\log_2\left(\frac{q_z(\alpha\,|\,s)}{p_x(\alpha\,|\,s)}\right) \quad (1)$$

$$= \lim_{n\to\infty}\frac{1}{n}\log_2\frac{q_z(Z^n)}{p_x(Z^n)} \quad \text{a.s.} \quad (2)$$

where χ is the alphabet and $\mathcal{S}$ is the set of states.

Manuscript received November 18, 2004; revised April 3, 2006. 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 authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: hcai@ee.princeton.edu; kulkarni@ee.princeton.edu; verdu@ee.princeton.edu).

Communicated by M. Effros, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2006.878182

[1] A commercial embodiment of LZ data compression.

[2] The use of lossless data compression techniques in other related problems such as modeling and prediction has also been considered; see for example [13] and [1].

A variety of compression algorithms have been proposed using the BWT as a front end followed by modules such as move-to-front, runlength coding, and adaptive Huffman coding. An entropy estimator based on the BWT was proposed in [5] using a uniform segmentation scheme. Based on that, we can show that segments of the BWT output sequence are close to an independent and identically distributed (i.i.d.) sequence. This property is exploited in our algorithm to estimate divergence without knowing the memory length of the sources.

Recently, experimental results have been reported [8] using the CTW method [15] for classification of binary sequences. The similarity metric used in [8] for classification can be seen to be an estimate of $D(q_z\|p_x) + H(q_z)$.

Another natural method, which we refer to as the plug-in method, for the estimation of divergence between two finite-memory sources consists of assuming an upper bound on the memory length of the sources, computing the empirical conditional probabilities and stationary probabilities by counting the number of symbols following each state, and plugging the estimates in (1).

The rest of the paper is organized as follows. In Section II, we present the divergence estimator based on the BWT. Convergence results for this estimator are proved in Section III. The divergence estimator based on CTW is presented and analyzed in Section IV. Finally, experimental results on tree sources and on text files (such as novels and the Bible) are presented in Section V. These results illustrate the superiority of our new algorithms over previous methods.

II. Divergence Estimator Via the BWT

A. The Burrows–Wheeler Transform (BWT)

The BWT is a reversible block-sorting algorithm [3]. It operates on a sequence of n symbols (with a unique "end-of-file" symbol "\$" appended), produces all n cyclic shifts of the original sequence, sorts them lexicographically, and outputs the last column of the sorted table. For finite-memory sources, performing the BWT on a reversed data sequence groups together symbols in the same state. Using the BWT followed by segmentation is the basic idea behind the entropy estimation in [5]. However, how to extend this idea to divergence estimation is not immediately clear because the two sources may have different state sets. The transitions for the two sequences do not occur in the same places, and need not correspond to the same contexts. In order to overcome this hurdle, we next introduce the joint BWT of two sequences.

For our purposes, it is important to extend the BWT to operate on the concatenation of two sequences $\boldsymbol{x}$ and $\boldsymbol{z}$ defined on the same alphabet. We concatenate $\boldsymbol{x}$ and $\boldsymbol{z}$, adding "\$" to the end of each sequence, then sort the table of cyclic shifts, and output the last column. The information of the origin of each symbol (whether from $\boldsymbol{x}$ or $\boldsymbol{z}$) is kept, but the sorting does not discriminate symbols by their origin. For clarity, we will use upper case letters to represent $\boldsymbol{x}$ and lower case letters to represent $\boldsymbol{z}$ in the following. It should always be understood that a and A designate the same symbol, with upper/lower case indicating from which source the symbol originates. For example, if $\boldsymbol{z} =$ ''banana'' $\boldsymbol{x} =$ ''anbaba'', then

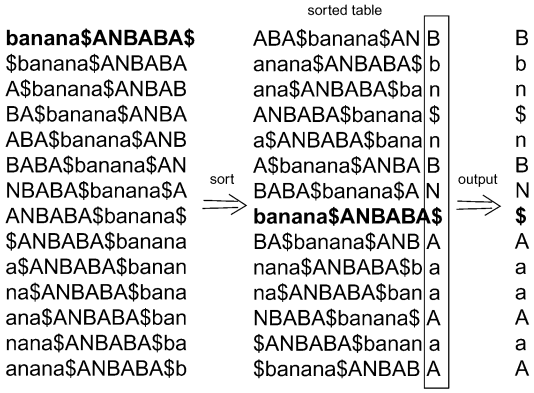

Fig. 1. An example of the joint BWT.

we feed "**banana**\$**ANBABA**\$" to the BWT, and the output is "Bbn\$nBN\$AaaAaA" as shown in Fig. 1. Unlike the standard use of the BWT in data compression, here we do not need to recover the original sequences from their block-sorted version.

B. Divergence Estimator Via the BWT

A basic task of our divergence estimator is to estimate conditional empirical distributions, which can be done through segmentation of the BWT output. As we will see, the divergence estimator runs the joint BWT on the concatenation of the two sequences.

For convenience, we will focus on the case in which the lengths of $\boldsymbol{x}$ and $\boldsymbol{z}$ are identical. However, it will be evident that this restriction is not necessary. In the discrete setting of this paper, it is convenient to decompose divergence as a difference

$$D(q_z \,\|\, p_x) = \lim_{n\to\infty} -\frac{1}{n}\log_2 p_x(Z^n) - H(q_z). \tag{3}$$

We can estimate the entropy term in (3) in a variety of ways; in particular, we can use the BWT-based algorithm in [5]. Thus, we focus on the estimation of the cross term in (3), i.e., on the sum of the entropy and divergence

$$\begin{aligned} S(q_z \,\|\, p_x) &\triangleq D(q_z \,\|\, p_x) + H(q_z) \\ &= \lim_{n\to\infty} -\frac{1}{n}\log_2 p_x(Z^n) \text{ a.s.} \end{aligned} \tag{4}$$

Thus, once $H(q_z)$ has been estimated, the problem of estimating $D(q_z \,\|\, p_x)$ boils down to estimating $\frac{1}{n}\log_2 p_x(z^n)$. To this end, we will proceed as if we were estimating the entropy of $\boldsymbol{x}$, but instead of evaluating $\log_2 p_x(\cdot)$ at $\boldsymbol{x}$, we will evaluate it at $\boldsymbol{z}$. The joint BWT allows us to do this in a natural way. We segment the joint BWT output based on statistics of the symbols from $\boldsymbol{x}$. Ideally, this would segment the joint BWT output at the transition points of $\boldsymbol{x}$, resulting in piecewise i.i.d. segments for $\boldsymbol{x}$ according to the memory structure of the source p_x. In each segment, we compute the empirical probability of each symbol according to $\boldsymbol{x}$, which gives estimates of the conditional probabilities in each state of the source p_x. These are used to estimate the probability (according to p_x) of the observed symbols from $\boldsymbol{z}$.

Note that the BWT of $\boldsymbol{z}$ can be obtained by simply retaining the symbols of $\boldsymbol{z}$ from the joint BWT. Thus, we actually need only perform the joint BWT to estimate both the entropy term

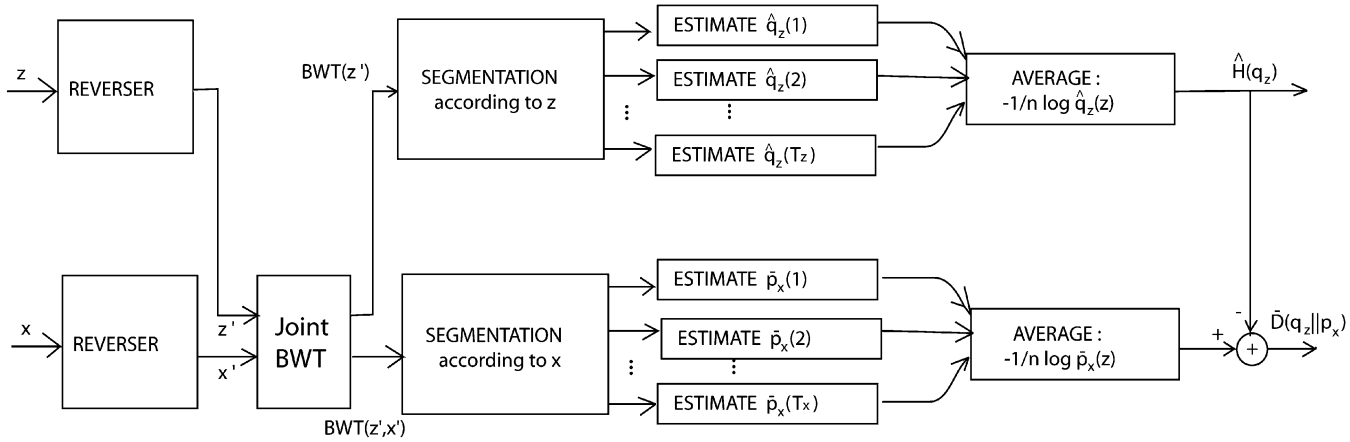

Fig. 2. Block diagram of the divergence estimator via the BWT.

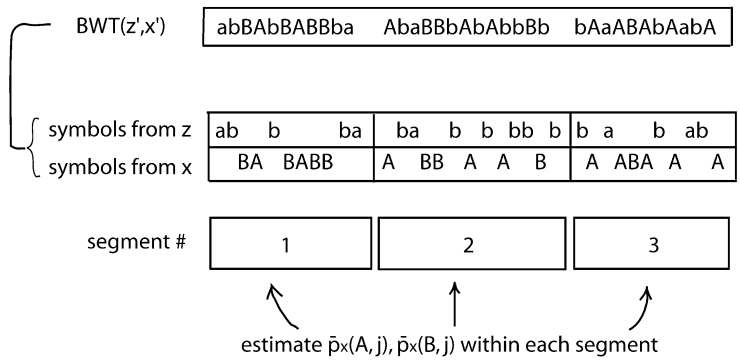

Fig. 3. The joint BWT, segmentation and estimation.

$H(q_z)$ and the divergence $D(q_z \,||\, p_x)$. However, to estimate $H(q_z)$, we segment the joint BWT output according to the symbols from $\boldsymbol{z}$, but to estimate $S(q_z \,||\, p_x)$ we segment the output according to $\boldsymbol{x}$.

We run the BWT on the concatenation of the reversed sequence of $\boldsymbol{x}$ and $\boldsymbol{z}$ as explained in Section II-A. Our estimator for $S(q_z \,||\, p_x)$ has the following steps (see Fig. 2).

a) *Run the joint BWT on the reversed concatenation of $\boldsymbol{z}$ and $\boldsymbol{x}$.*

b) *Segmentation according to symbols from $\boldsymbol{x}$.* This will be described further in Section II-C. If a uniform segmentation strategy is adopted, the segments contain an equal number of symbols from $\boldsymbol{x}$ but, in general, different numbers of symbols from $\boldsymbol{z}$. Therefore, this segmentation induces a segmentation on the symbols of the sequence $\boldsymbol{z}$ which is different from the segmentation performed for the estimation of the entropy of $\boldsymbol{z}$.

c) *Estimate the first-order distribution (according to $\boldsymbol{x}$) within each segment as shown in Fig. 3.* Denote the number of occurrences of symbol a (from $\boldsymbol{z}$) in the jth segment (according to the segmentation with respect to $\boldsymbol{x}$) by $\bar{N}_j(a)$. Similarly, $\bar{N}_j(A)$ denotes the number of occurrences of symbol A (from $\boldsymbol{x}$) in the jth segment.
An estimate of the probability of the symbols from $\boldsymbol{x}$ in the jth segment is

$$\bar{p}_{\boldsymbol{x}}(A, j) = \frac{\bar{N}_j(A) + \Delta}{\sum_{B \in \chi} \bar{N}_j(B) + |\chi|\Delta}. \tag{5}$$

If a probability estimate in a segment is zero, the logarithm of the probability is $-\infty$ and the divergence estimate is $+\infty$. Therefore, a small bias $\Delta > 0$ is introduced in the probability estimates in order to deal with this issue. In Section III, we will analyze the impact of this bias on our estimator.
The contribution of the jth segment to the estimate of $\log_2 \bar{p}_{\boldsymbol{x}}(z^n)$ is

$$\log_2 \bar{p}_{\boldsymbol{x}}(j) = \sum_{a \in \chi} \bar{N}_j(a) \log_2 \bar{p}_{\boldsymbol{x}}(A, j). \tag{6}$$

Recall that a and A index the same symbol in the alphabet with the upper case notation indicating that the symbol comes from $\boldsymbol{x}$ and the lower case indicating that the symbol comes from $\boldsymbol{z}$. Hence, the term $\bar{N}_j(a)$ in (6) has "a" since this is a count of the number of occurrences in $\boldsymbol{z}$, while the term $\log_2 \bar{p}_{\boldsymbol{x}}(j, A)$ has "A" since this is an estimate of the log probability of the symbol based on $\boldsymbol{x}$.

d) *Average across segments*. The estimate of the sought-after divergence-plus-entropy functional is obtained by averaging the contributions from the T_x segments

$$\bar{S}(q_z \,||\, p_x) = -\frac{1}{n} \sum_{j=1}^{T_x} \log_2 \bar{p}_{\boldsymbol{x}}(j). \tag{7}$$

C. Segmentation

Segmentation is an important step in the divergence estimator. As discussed in [5], there are two natural approaches.

One approach, called adaptive segmentation, is to detect the transitions based on the empirical distributions of the BWT output. The other, called uniform segmentation, simply divides the BWT output into segments so that each segment contains an equal number of symbols, denoted by $w(n)$, from the sequence according to which we are segmenting. For example, in Fig. 3 each segment has the same number of symbols from $\boldsymbol{x}$. However, the segments of the joint BWT output (containing symbols from both $\boldsymbol{x}$ and $\boldsymbol{z}$) may be different in length due to the different numbers of symbols from $\boldsymbol{z}$.

Recall that to estimate $H(q_z)$, the segmentation is performed according to symbols from $\boldsymbol{z}$ (indicated by the lower case symbols), while the divergence-plus-entropy functional used to estimate $D(q_z \,||\, p_x)$ requires a segmentation according to symbols from $\boldsymbol{x}$ (indicated by the upper case symbols).

III. Performance Analysis of the BWT-Based Divergence Estimator

In this section, we prove the convergence of our estimator with uniform segmentation and obtain conditions on the growth of segment length for our algorithm. We assume that the sources are stationary Markov with a finite unknown order. The state sequence forms a Markov chain, which is assumed to be irreducible and aperiodic with a unique stationary distribution. The set of states for both sources is $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$, which is in lexicographical order, as sorted by the BWT. Denote by $S_i \in \mathcal{S}$ the state from which symbol z_i emanates, where $1 \le i \le n$. We further assume that the conditional probabilities $p_x(a|s)$ and $q_z(a|s)$ satisfy $p_x(a|s) < 1$ and $q_z(a|s) < 1$, for all $a \in \chi$ and $s \in \mathcal{S}$.

A. Empirical Probabilities in Each Segment

In order to prove convergence of the divergence estimator, we must show that the empirical probabilities in almost every $w(n)$-segment are close to the true conditional probabilities for a corresponding state of the Markov source (Lemma 1), and there are not too many symbols from $\boldsymbol{z}$ falling into any single segment (Lemma 3). The variability of the number of symbols from $\boldsymbol{z}$ falling in each segment presents a major challenge.

Let $N_{\boldsymbol{x}}(s)$ be the number of occurrences of state s in $\boldsymbol{x}$. For $1 \le k \le n/w(n)$, let $S_n(k)$ be the state of segment k if all symbols in segment k emanate from the same state. We have

$$S_n(k) = s_i \tag{8}$$

if both

$$kw(n) \le \sum_{1 \le j \le i} N_{\boldsymbol{x}}(s_j) \tag{9}$$

and

$$(k-1)w(n) \ge \sum_{j<i} N_{\boldsymbol{x}}(s_j) \tag{10}$$

are satisfied for some $1 \le i \le |\mathcal{S}|$; if there is no such i, then $S_n(k) = \phi$, which means it is a "bad" segment containing a state transition. Let $B = \{j : 1 \le j \le n/w(n), S_n(j) = \phi\}$ be the set of bad segments containing a state transition. Denote by $\hat{p}_{\boldsymbol{x}}(a,k)$ the empirical probability of symbol a in the kth segment. Let $K_s = \big|\{j : 1 \le j \le n/w(n), S_n(j) = s\}\big|$ be the number of segments whose state is $s \in S$. Let us define the following quantities:

$$\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s) = \frac{1}{K_s} \sum_{j: S_n(j) = s} \hat{p}_{\boldsymbol{x}}(a, j), \tag{11}$$

and

$$\bar{\bar{p}}_{\boldsymbol{x}}(s) = \frac{w(n) K_s}{n}. \tag{12}$$

$\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s)$ is an estimate of the conditional probability $p_x(a \,|\, s)$, and $\bar{\bar{p}}_{\boldsymbol{x}}(s)$ is an estimate of the stationary probability $\mu_x(s)$. The proof of Theorems 1 and 2 in [5] introduces a fictitious estimator, that discards bad segments and fuses segments corresponding to the same states. In fact, $\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s)$ and $\bar{\bar{p}}_{\boldsymbol{x}}(s)$ are estimates in the fused segment corresponding to state s as in the fictitious entropy estimator.

Lemma 1: If we choose the segment length $w(n) = n^{\frac{1}{2}+\eta}$, where $0 < \eta < 1/2$, then for any $a \in \chi$ and $s \in \mathcal{S}$, we have

$$E\left[\frac{1}{K_s} \sum_{j: S_n(j) = s} \left(\hat{p}_{\boldsymbol{x}}(a,j) - p_x(a \,|\, s)\right)^2\right] = O\left(\frac{\log n}{n^{\frac{1}{2}-\eta}}\right) \tag{13}$$

and therefore,

$$E\left[\frac{w(n)}{n} \sum_{k \notin B} \sum_{a \in \chi} \left(\hat{p}_{\boldsymbol{x}}(a,k) - p_x(a \,|\, S_n(k))\right)^2\right] = O\left(\frac{\log n}{n^{\frac{1}{2}-\eta}}\right) \tag{14}$$

where the expectation is with respect to the source p_x.

Proof: By the convergence rate of the entropy estimator for finite state Markov sources [5, Theorem 2], we have

$$\begin{aligned} &E\left[\left(\frac{w(n)}{n} \sum_{k \notin B} \sum_{a \in \chi} \hat{p}_{\boldsymbol{x}}(a,k) \log \frac{1}{\hat{p}_{\boldsymbol{x}}(a,k)} - H(p_x)\right)^2\right] \\ &\quad = O\left(\frac{w^2(n) \log^2 n}{n^2}\right) + O\left(\frac{\log^2 n}{w^2(n)}\right) \\ &\quad = O\left(\frac{\log^2 n}{n^{1-2\eta}}\right) \end{aligned} \tag{15}$$

and

$$\begin{aligned} &E\left[\left(\sum_{s \in S} \bar{\bar{p}}_{\boldsymbol{x}}(s) \sum_{a \in \chi} \bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s) \log \frac{1}{\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s)} - H(p_x)\right)^2\right] \\ &\quad = O\left(\frac{\log^2 n}{n^{1-2\eta}}\right). \end{aligned} \tag{16}$$

It follows that, for any $a \in \chi$ and any $s \in \mathcal{S}$

$$\begin{aligned} &E\left[\left(\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s) \log \frac{1}{\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s)} - \frac{1}{K_s} \right.\right. \\ &\qquad \left.\left. \times \sum_{j: S_n(j) = s} \hat{p}_{\boldsymbol{x}}(a,j) \log \frac{1}{\hat{p}_{\boldsymbol{x}}(a,j)}\right)^2\right] \\ &\qquad = O\left(\frac{\log^2 n}{n^{1-2\eta}}\right) \end{aligned} \tag{17}$$

when $w(n) = n^{\frac{1}{2}+\eta}$, and $0 < \eta < 1/2$. Then (13) follows from (17), using Lemma 2 and the fact that

$$E[(\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s) - p_x(a \,|\, s))^2] = O\left(\frac{1}{n^{1-2\eta}}\right). \tag{18}$$

□

In Lemma 2, we will drop a, s, and the subscript $\boldsymbol{x}$ in $\bar{\bar{p}}_{\boldsymbol{x}}(a \,|\, s)$ and $\hat{p}_{\boldsymbol{x}}(a, j)$ and simply use notation $\hat{p}(j)$ for probabilities, where $1 \le j \le K$.

Lemma 2: Let

$$E = \bar{\bar{p}} \log \frac{1}{\bar{\bar{p}}} - \frac{1}{K} \sum_{1 \le j \le K} \hat{p}(j) \log \frac{1}{\hat{p}(j)} \tag{19}$$

where

$$\bar{\bar{p}} = \frac{1}{K} \sum_{1 \le j \le K} \hat{p}(j). \tag{20}$$

There exists $\delta > 0$, such that if $\{\hat{p}(j)\}$ satisfies $E < \delta$, then

$$\frac{1}{K} \sum_{1 \le j \le K} (\hat{p}(j) - \bar{\bar{p}})^2 \le 2E. \tag{21}$$

Proof: We want to find the maximum

$$F(\bar{\bar{p}}, E, K) \triangleq \max_{\hat{p}(\cdot)} \left(\frac{1}{K} \sum_j (\hat{p}(j) - \bar{\bar{p}})^2 \right) \tag{22}$$

under the constraints that

$$\bar{\bar{p}} \log \frac{1}{\bar{\bar{p}}} - \left(\frac{1}{K} \sum_j \hat{p}(j) \log \frac{1}{\hat{p}(j)} \right) = E \tag{23}$$

$$\frac{1}{K} \sum_j \hat{p}(j) = \bar{\bar{p}} \tag{24}$$

and for $1 \le j \le K$

$$0 \le \hat{p}(j) \le 1. \tag{25}$$

We first consider maximizing over only three of the variables $\hat{p}(\cdot)$ holding the rest fixed and arbitrary. It is equivalent to maximize $p_1^2 + p_2^2 + p_3^2$ under the constraints that $0 \le p_1 \le 1, 0 \le p_2 \le 1, 0 \le p_3 \le 1$

$$\frac{p_1 + p_2 + p_3}{3} = p \tag{26}$$

and

$$\frac{p_1 \log \frac{1}{p_1} + p_2 \log \frac{1}{p_2} + p_3 \log \frac{1}{p_3}}{3} = p \log \frac{1}{p} - \varsigma \tag{27}$$

where $0 < p < 1$ and $\varsigma > 0$ are constants. Without loss of generality, we assume $1 \ge p_1 \ge p_2 \ge p_3 \ge 0$. It is shown in Appendix A that the maximum is achieved when either $p_1 = 1$ or $p_1 > p_2 = p_3$.

Going back to the original constrained maximization problem (22), we see that the maximizing $\hat{p}(j)$'s should take at most three distinct values, one of them is 1. We have $F(\bar{\bar{p}}, E, K) \le G_3(\bar{\bar{p}}, E)$, where

$$G_3(\bar{\bar{p}}, E) \triangleq \max_{p_2, p_3, \alpha_2, \alpha_3} \left\{ (1 - \alpha_2 - \alpha_3) + \alpha_2 p_2^2 \alpha_3 p_3^2 - \bar{\bar{p}}^2 \right\} \tag{28}$$

under the constraints

$$(1 - \alpha_2 - \alpha_3) + \alpha_2 p_2 + \alpha_3 p_3 = \bar{\bar{p}} \tag{29}$$

$$-\alpha_2 p_2 \log p_2 - \alpha_3 p_3 \log p_3 = -\bar{\bar{p}} \log \bar{\bar{p}} - E \tag{30}$$

and $\alpha_2 \ge 0, \alpha_3 \ge 0, 1 - \alpha_2 - \alpha_3 \ge 0$, and $1 \ge p_2 \ge 0, 1 \ge p_3 \ge 0$. It is shown in Appendix B that the maximum of $G_3(\bar{\bar{p}}, E)$ is achieved when $p_2 = 1$ (or equivalently $\alpha_2 = 0$). Therefore,

$$G_3(\bar{\bar{p}}, E) = (1 - \alpha_3) + \alpha_3 p_3^2 - \bar{\bar{p}}^2 \tag{31}$$

where

$$(1 - \alpha_3) + \alpha_3 p_3 = \bar{\bar{p}}, \tag{32}$$

and

$$-\alpha_3 p_3 \log p_3 = -\bar{\bar{p}} \log \bar{\bar{p}} - E. \tag{33}$$

As E goes to 0, p_3 goes to $\bar{\bar{p}}$, and $G_3(\bar{\bar{p}}, E) = (\bar{\bar{p}} - p_3)(1 - \bar{\bar{p}})$ goes to zero. We have

$$\lim_{E \to 0} \frac{G_3(\bar{\bar{p}}, E)}{E} = \frac{(1 - \bar{\bar{p}})^2}{-\log \bar{\bar{p}} + \bar{\bar{p}} - 1} \tag{34}$$

which belongs to $(0, 2)$. This completes the proof. □

B. Divergence Estimator With Uniform Segmentation for Markov Sources

Theorem 1: Let $\boldsymbol{z}$ and $\boldsymbol{x}$ be sequences of length n generated from finite-alphabet finite-state Markov sources q_z and p_x, respectively. Assume that $D(q_z \,\|\, p_x) < +\infty$, and $q_z(a \,|\, s) < 1, p_x(a \,|\, s) < 1$ for any $a \in \chi$ and $s \in \mathcal{S}$. Let $\bar{D}_n(q_z \,\|\, p_x)$ denote the divergence estimate using uniform segmentation with segment length $w(n) = n^{\frac{1}{2}+\eta}$, where $0 < \eta < 1/2$. Then $\bar{D}_n(q_z \,\|\, p_x)$ converges to $D(q_z \,\|\, p_x)$ in probability.

Proof: We first decompose the divergence into two terms, and estimate each separately.

$$D(q_z \,\|\, p_x) = S(q_z \,\|\, p_x) - H(q_z). \tag{35}$$

$$\bar{D}_n(q_z \,\|\, p_x) = \bar{S}_n(q_z \,\|\, p_x) - \hat{H}_n(q_z). \tag{36}$$

Since we have proved the convergence of the entropy estimator in [5], it remains to prove that the cross term $\bar{S}_n(q_z \,\|\, p_x)$ converges to $S(q_z \,\|\, p_x)$ in probability.

Let $N_{\boldsymbol{z}}(s)$ be the number of occurrences of state s in $\boldsymbol{z}$, and $N_{\boldsymbol{z}}(a, s)$ be the number of occurrences of symbol a emanating from state s in $\boldsymbol{z}$. Let $N_{\boldsymbol{z}}(a, k)$ be the number of occurrences of symbol a from $\boldsymbol{z}$ in the kth segment, and $N_{\boldsymbol{z}}(k)$ be the total number of symbols from $\boldsymbol{z}$ in the kth segment. Note that $w(n)$

is the number of symbols from $\boldsymbol{x}$ in each segment, so $N_{\boldsymbol{z}}(k)$ and $w(n)$ are different. Define

$$\check{S}_n(q_z \,\|\, p_x) \triangleq -\frac{1}{n}\sum_{s\in\mathcal{S}}\sum_{a\in\chi} N_{\boldsymbol{z}}(a,s)\log_2 p_x(a\,|\,s). \tag{37}$$

In Appendix D, we prove that

$$E[(\check{S}_n(q_z \,\|\, p_x) - S(q_z \,\|\, p_x))^2] = O\left(\frac{1}{n}\right). \tag{38}$$

Now consider the estimator

$$\bar{S}_n(q_z \,\|\, p_x) = -\frac{1}{n}\sum_{k}\left(\sum_{a\in\chi} N_{\boldsymbol{z}}(a,k)\log_2 \bar{p}_{\boldsymbol{x}}(a,k)\right) \tag{39}$$

where $\bar{p}_{\boldsymbol{x}}(a,k)$ is the estimate of $p_x(a|\cdot)$ in the kth segment. We only need to prove that

$$\bar{S}_n(q_z \,\|\, p_x) - \check{S}_n(q_z \,\|\, p_x) \to 0 \tag{40}$$

in probability, as $n\to\infty$. In fact

$$\check{S}_n(q_z \,\|\, p_x) = -\frac{1}{n}\log p_x(z^n)$$

where p_x is the probability of the source generating $\boldsymbol{x}$. And $\bar{S}_n(q_z \,\|\, p_x) = -\frac{1}{n}\log \bar{p}_{\boldsymbol{x}}(z^n)$, where $\bar{p}_{\boldsymbol{x}}$ is an estimate of p_x

$$\bar{p}_{\boldsymbol{x}}(z^n) = \prod_{i=1}^{n} \bar{p}_{\boldsymbol{x}}(z_i, \ell(\boldsymbol{x},\boldsymbol{z},i)), \tag{41}$$

where $\ell(\boldsymbol{x},\boldsymbol{z},i)$ is the index of the segment that z_i falls into after the joint BWT of $\boldsymbol{x}$ and $\boldsymbol{z}$.

Let

$$\tau \triangleq \min_{a\in\chi s\in\mathcal{S}}\{p_x(a\,|\,s) : p_x(a\,|\,s) > 0\}. \tag{42}$$

For any $\epsilon>0$, there exists $\delta_\epsilon>0$, such that $|\log \bar{p}_{\boldsymbol{x}} - \log p_x| \le \epsilon$, for any $|\bar{p}_{\boldsymbol{x}} - p_x| \le \delta_\epsilon$ and $p_x \ge \tau$. We proceed to bound the number of z_i that fall into segments that do not have good estimates.

We use the biased estimates $\bar{p}_{\boldsymbol{x}}(a,j)$ instead of the empirical estimates $\hat{p}_{\boldsymbol{x}}(a,j)$, so that probability estimates are lower-bounded by $\frac{\Delta}{w(n)+\Delta|\chi|}$

$$\log\frac{1}{\bar{p}_{\boldsymbol{x}}(a,j)} \le \log\left(\frac{w(n)+\Delta|\chi|}{\Delta}\right) = O(\log n). \tag{43}$$

By Lemma 1, for any $a\in\chi$ and $s\in\mathcal{S}$ we have

$$E\left[\frac{w(n)}{n}\sum_{j:S_n(j)=s}(\hat{p}_{\boldsymbol{x}}(a,j) - p_x(a\,|\,s))^2\right] = O\left(\frac{\log n}{n^{\frac{1}{2}-\eta}}\right). \tag{44}$$

Since

$$|\bar{p}_{\boldsymbol{x}}(a,j) - \hat{p}_{\boldsymbol{x}}(a,j)| \le \frac{(|\chi|-1)\Delta}{w(n)}$$

$\bar{p}_{\boldsymbol{x}}(a,j)$ has the same property as $\hat{p}_{\boldsymbol{x}}(a,j)$, namely,

$$E\left[\frac{1}{n^{\frac{1}{2}-\eta}}\sum_{j:S_n(j)=s}(\bar{p}_{\boldsymbol{x}}(a,j) - p_x(a\,|\,s))^2\right] = O\left(\frac{\log n}{n^{\frac{1}{2}-\eta}}\right) \tag{45}$$

for any $a\in\chi$ and $s\in\mathcal{S}$.

Let $A_{n,a,s}(\delta)$ and $E_{n,a,s}(\delta)$ be the sets of segment indices

$$A_{n,a,s}(\delta) = \{k : 1\le k\le n/w(n), S_n(k) = s \text{ and } |\bar{p}_{\boldsymbol{x}}(a,k) - p_x(a\,|\,s)| > \delta\}. \tag{46}$$

$$E_{n,a,s}(\delta) = \{k : 1\le k\le n/w(n), S_n(k) = s \text{ and } |\bar{p}_{\boldsymbol{x}}(a,k) - p_x(a\,|\,s)| \le \delta\}. \tag{47}$$

It follows from (45) that, for $\delta=\delta_\epsilon>0$

$$\Pr\left\{|A_{n,a,s}(\delta)| > \frac{\log^2 n}{\delta^2}\right\} \le \frac{C}{\log n}. \tag{48}$$

Now let us write

$$|\bar{S}_n(q_z \,\|\, p_x) - \check{S}_n(q_z \,\|\, p_x)|$$

$$= \sum_{i=1}^{n}\frac{1}{n}|\log p_x(z_i|S_i) - \log\bar{p}_{\boldsymbol{x}}(z_i,\ell(\boldsymbol{x},\boldsymbol{z},i))| \tag{49}$$

$$= \sum_{a\in\chi}\sum_{s\in\mathcal{S}}\sum_{i:z_i=a,S_i=s}\frac{1}{n}|\log p_x(z_i|S_i) - \log\bar{p}_{\boldsymbol{x}}(z_i,\ell(\boldsymbol{x},\boldsymbol{z},i))| \tag{50}$$

$$\triangleq \sum_{a\in\chi}\sum_{s\in\mathcal{S}} J(a,s). \tag{51}$$

Under the assumption $D(q_z \,\|\, p_x) < +\infty$, if $p_x(a\,|\,s)=0$, then $q_z(a\,|\,s)=0$, and therefore, $J(a,s)=0$. If $p_x(a\,|\,s)\ge\tau>0$, we have

$$\begin{aligned} J(a,s) \le & \sum_{i:z_i=a,\ell(\boldsymbol{x},\boldsymbol{z},i)\in E_{n,a,s}(\delta)}\frac{1}{n} \\ & \times |\log p_x(z_i|S_i) - \log\bar{p}_{\boldsymbol{x}}(z_i,\ell(\boldsymbol{x},\boldsymbol{z},i))| \\ & + \sum_{i:z_i=a,\ell(\boldsymbol{x},\boldsymbol{z},i)\in A_{n,a,s}(\delta)}\frac{1}{n} \\ & \times |\log p_x(z_i|S_i) - \log\bar{p}_{\boldsymbol{x}}(z_i,\ell(\boldsymbol{x},\boldsymbol{z},i))| \\ & + \sum_{i:z_i=a,S_i=s,\ell(\boldsymbol{x},\boldsymbol{z},i)\in B}\frac{1}{n} \\ & \times |\log p_x(z_i|S_i) - \log\bar{p}_{\boldsymbol{x}}(z_i,\ell(\boldsymbol{x},\boldsymbol{z},i))| \end{aligned} \tag{52}$$

$$\le \frac{\epsilon N_{\boldsymbol{z}}(a,s)}{n} + \frac{C\log n}{n}|\{i : \ell(\boldsymbol{x},\boldsymbol{z},i)\in A_{n,a,s}(\delta)\}| + \frac{C\log n}{n}|\{i : z_i=a, S_i=s, \ell(\boldsymbol{x},\boldsymbol{z},i)\in B\}|. \tag{53}$$

Therefore, we have

$$\begin{aligned} &|\bar{S}_n(q_z \,\|\, p_x) - \check{S}_n(q_z \,\|\, p_x)| \\ &\le \epsilon + \frac{C\log n}{n}|\{i : \ell(\boldsymbol{x},\boldsymbol{z},i)\in B\}| \\ &\quad + \sum_{a\in\chi}\sum_{s\in\mathcal{S}}\frac{C\log n}{n}|\{i : \ell(\boldsymbol{x},\boldsymbol{z},i)\in A_{n,a,s}(\delta)\}|. \end{aligned} \tag{54}$$

In order to prove convergence in probability, we only need to prove that, for any $\epsilon > 0$

$$\Pr\left\{\frac{\log n}{n}|\{i : \ell(\boldsymbol{x}, \boldsymbol{z}, i) \in A_{n,a,s}(\delta)\}| \le \epsilon\right\} \to 1 \quad (55)$$

and

$$\Pr\left\{\frac{\log n}{n}|\{i : \ell(\boldsymbol{x}, \boldsymbol{z}, i) \in B\}| \le \epsilon\right\} \to 1 \quad (56)$$

as $n \to \infty$.

It follows from Lemma 3, choosing $\varepsilon = \epsilon\delta^2$

$$\Pr\left\{\bigcup_k \left\{\frac{N_{\boldsymbol{z}}(k)}{n} \ge \frac{\varepsilon}{\log^3 n}\right\}\right\} \le \sum_k \Pr\left\{\frac{N_{\boldsymbol{z}}(k)}{n} \ge \frac{\varepsilon}{\log^3 n}\right\} \le O\left(n^{-\frac{1}{2}} \log^3 n\right). \quad (57)$$

Therefore,

$$\Pr\left\{\max_k \frac{N_{\boldsymbol{z}}(k)}{n} < \frac{\varepsilon}{\log^3 n}\right\} = \Pr\left\{\bigcap_k \left\{\frac{N_{\boldsymbol{z}}(k)}{n} < \frac{\varepsilon}{\log^3 n}\right\}\right\} \to 1 \quad (58)$$

as $n \to \infty$. Then, (55) and (56) can be proved by combining (58) and (48), and the fact that B is a finite set (namely, $|B| \le |\mathcal{S}|$). □

Lemma 3: Under the same assumptions of Theorem 1, for any $\varepsilon > 0$, and any $1 \le k \le n/w(n)$, where $w(n) = n^{\frac{1}{2}+\eta}$ $(0 < \eta < \frac{1}{2})$, we have

$$\Pr\left\{\frac{N_{\boldsymbol{z}}(k)}{n} \ge \frac{\varepsilon}{\log^3 n}\right\} = O(n^{\eta-1} \log^3 n). \quad (59)$$

Proof: In the BWT output, symbols from both $\boldsymbol{x}$ and $\boldsymbol{z}$ are sorted by context. In one segment, symbols from $\boldsymbol{x}$ and symbols from $\boldsymbol{z}$ have common context(s). Here, a context can be understood as a node in the context tree, which specifies a number of symbols occurring before the current symbol. The depth of the node equals the number of symbols the context specifies. The (infinite) sequence of past symbols is referred to as unbounded context. The uniform segmentation is determined by the realization $\boldsymbol{x}$, and results in a partition at the leaves of the context tree into $n/w(n)$ classes, as shown in Fig. 4. In this example, there are four segments. The first segment contains contexts "AA" and "AABA"; the second segment contains contexts "BABA" and "BBA"; the third segment contains contexts "AB" and "ABB"; the fourth segment contains context "BBB". The set of contexts of a segment is in fact represented by a set of nodes in the context tree. In order to determine the set of contexts of the kth segment, we must find the boundary of the kth segment and the $(k+1)$th segment (as well as the boundary of the $(k-1)$th segment and the kth segment) in the context tree.

In the above, we have described an "ideal" nonoverlapping partition of contexts based on uniform segmentation according to symbols from $\boldsymbol{x}$. However, if there are symbols (from $\boldsymbol{z}$) between the last symbol (from $\boldsymbol{x}$) of the kth segment and the first symbol (from $\boldsymbol{x}$) of the $(k+1)$th segment, we cannot immediately determine to which segment they belong according to the "ideal" partition. In our implementation of the algorithm, we resolve this by simply including those symbols from $\boldsymbol{z}$ into the previous segment. As our objective is to prove that the probability of $N_{\boldsymbol{z}}(k)$ being too large is very small, it suffices to prove it for an upper bound of $N_{\boldsymbol{z}}(k)$. So we modify the definition of the set of contexts of a segment, and allow a small overlap across two consecutive segments.

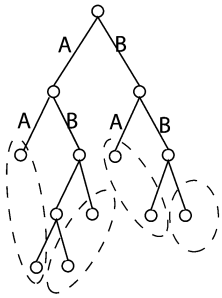

Fig. 4. "Ideal" partition of the context tree.

The set of contexts of the kth segment can be determined by finding the first different symbol in the unbounded contexts of the first and second symbols (from $\boldsymbol{x}$) of the $(k+1)$th segment and the first different symbol in the unbounded contexts of the last two symbols (from $\boldsymbol{x}$) of the $(k-1)$th segment. Note that the set of contexts of the kth segment under this definition is slightly larger than the set of contexts defined by the ideal partition.

Under the assumption that all conditional probabilities are less than 1, there exists a constant $0 < \xi < 1$ such that $p_x(a \,|\, s) \le \xi$ and $q_z(a \,|\, s) \le \xi$ for any $a \in \chi$ and $s \in \mathcal{S}$.

Let us denote the stationary probability of contexts of the Markov source p_x and the Markov source q_z by μ_x and μ_z, respectively. Let us consider the set of contexts $U_n(k)$, which is a set of nodes that are to the left of a boundary in the context tree, such that all the nodes included in $U_n(k)$ have depths no more than $l \triangleq \lceil \frac{\beta \log n}{\log \frac{1}{\xi}} \rceil$, where $0 < \beta < \min\{\frac{\eta}{2}, \frac{1}{2} - \eta\}$ is a constant, and the following condition is satisfied:

if $\frac{kw(n)}{n} + \frac{1}{n^\beta} \le 1$

$$\frac{kw(n)}{n} + \frac{1}{n^\beta} \le \mu_x(U_n(k)) \le \frac{kw(n)}{n} + \frac{2}{n^\beta} \quad (60)$$

if $\frac{kw(n)}{n} + \frac{1}{n^\beta} > 1$

$$\mu_x(U_n(k)) = 1. \quad (61)$$

Since the stationary distribution of a node of depth $\lceil \frac{\beta \log n}{\log \frac{1}{\xi}} \rceil$ is upper-bounded by $\frac{1}{n^\beta}$, such $U_n(k)$ exists. Similarly, consider the set of contexts $L_n(k)$, which is a set of nodes that are to the left of a boundary in the context tree, such that all the nodes included in $L_n(k)$ have depths no more than $\lceil \frac{\beta \log n}{\log \frac{1}{\xi}} \rceil$, and the following condition is satisfied:

if $\frac{(k-1)w(n)}{n} - \frac{1}{n^\beta} \ge 0$

$$\frac{(k-1)w(n)}{n} - \frac{2}{n^\beta} \le \mu_x(L_n(k)) \le \frac{(k-1)w(n)}{n} - \frac{1}{n^\beta} \quad (62)$$

if $\frac{(k-1)w(n)}{n} - \frac{1}{n^\beta} < 0$

$$\mu_x(L_n(k)) = 0. \quad (63)$$

Let $N_{\boldsymbol{x}}(U_n(k))$ denote the number of symbols in $\boldsymbol{x}$ that have context in $U_n(k)$ and $N_{\boldsymbol{x}}(L_n(k))$ denote the number of symbols in $\boldsymbol{x}$ that have context in $L_n(k)$. Apparently, if $\mu_x(U_n(k)) = 1$, then $N_{\boldsymbol{x}}(U_n(k)) = n$; if $\mu_x(L_n(k)) = 0$, then $N_{\boldsymbol{x}}(L_n(k)) = 0$. For the nontrivial cases, we have

$$\mu_x(U_n(k)) = \frac{kw(n)}{n} + \frac{1}{n^{\delta_1}} \tag{64}$$

and

$$\mu_x(L_n(k)) = \frac{(k-1)w(n)}{n} - \frac{1}{n^{\delta_2}} \tag{65}$$

where $\frac{1}{2}\beta < \delta_1 \leq \beta$ and $\frac{1}{2}\beta < \delta_2 \leq \beta$. Therefore, we have

$$\begin{aligned}
&\Pr\{N_{\boldsymbol{x}}(U_n(k)) \leq kw(n)\} \\
&\quad = \Pr\{N_{\boldsymbol{x}}(U_n(k)) - kw(n) - n^{1-\delta_1} \leq -n^{1-\delta_1}\} && (66)\\
&\quad = \Pr\{N_{\boldsymbol{x}}(U_n(k)) - n\mu_x(U_n(k)) \leq -n^{1-\delta_1}\} && (67)\\
&\quad \leq n^{2\delta_1} E\left[\left(\frac{1}{n}N_{\boldsymbol{x}}(U_n(k)) - \mu_x(U_n(k))\right)^2\right] && (68)\\
&\quad = O(n^{2\beta-1}l^3) && (69)\\
&\quad = O(n^{2\beta-1}\log^3 n) && (70)
\end{aligned}$$

and

$$\begin{aligned}
&\Pr\{N_{\boldsymbol{x}}(L_n(k)) \geq (k-1)w(n)\} \\
&\quad = \Pr\{N_{\boldsymbol{x}}(L_n(k)) - (k-1)w(n) + n^{1-\delta_2} \geq n^{1-\delta_2}\} && (71)\\
&\quad = \Pr\{N_{\boldsymbol{x}}(L_n(k)) - n\mu_x(L_n(k)) \geq n^{1-\delta_2}\} && (72)\\
&\quad \leq n^{2\delta_2} E\left[\left(\frac{1}{n}N_{\boldsymbol{x}}(L_n(k)) - \mu_x(L_n(k))\right)^2\right] && (73)\\
&\quad = O(n^{2\beta-1}l^3) && (74)\\
&\quad = O(n^{2\beta-1}\log^3 n) && (75)
\end{aligned}$$

where (69) and (74) follow from Lemma 5 in Appendix F; (70) and (75) follow from the fact that $l = \lceil \frac{\beta \log n}{\log \frac{1}{\xi}} \rceil$. Thus, we have

$$\begin{aligned}
&\Pr\{\{N_{\boldsymbol{x}}(U_n(k)) \leq kw(n)\} \bigcup \{N_{\boldsymbol{x}}(L_n(k)) \\
&\quad \geq (k-1)w(n)\}\} \\
&\quad = O(n^{2\beta-1}\log^3 n). && (76)
\end{aligned}$$

Let $M_n(k)$ be the context set that contains contexts in $U_n(k)$ but not in $L_n(k)$

$$M_n(k) = U_n(k) \setminus L_n(k). \tag{77}$$

From (60) and (62), we have

$$\mu_x(M_n(k)) \leq \frac{1}{n^{\frac{1}{2}-\eta}} + \frac{4}{n^{\beta}} \leq \frac{5}{n^{\beta}}, \tag{78}$$

where $0 < \beta < \min\{\frac{\eta}{2}, \frac{1}{2} - \eta\}$.

Next, we establish the connection between μ_x and μ_z. Let $T_n(k)$ be any set of contexts. As shown in Fig. 5, we find the last symbol common to these contexts, which corresponds to the latest common ancestor of the set of nodes specified by $T_n(k)$. Then, the set $T_n(k)$ can be divided into two disjoint sets $T_n(k) = T_n^L(k) \bigcup T_n^R(k)$, which are in different branches of the common ancestor (the division is not unique if more than two branches of the common ancestor are involved). Denote the common ancestor of the nodes in $T_n^L(k)$ by $P^L(k)$, the node $P^L(k)$'s rightmost child in $T_n^L(k)$ by $C^L(k)$. Let $\Lambda(P^L(k))$ be the set of nodes including the node $P^L(k)$ and all its offsprings. Then we have $C^L(k) \in T_n^L(k) \subseteq \Lambda(P^L(k))$. Let

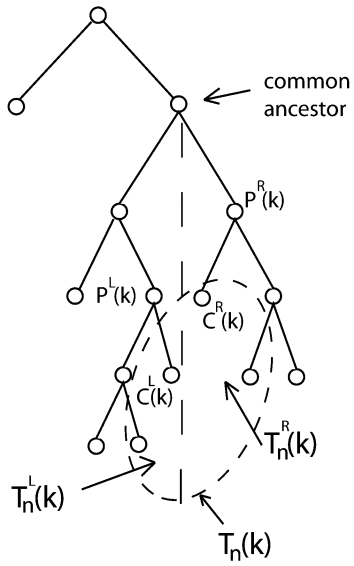

Fig. 5. The set of contexts $T_n(k)$ is divided into two disjoint sets $T_n^L(k)$ and $T_n^R(k)$.

$$\tau = \min_{a \in \chi s \in \mathcal{S}} \{p_x(a \,|\, s) : p_x(a \,|\, s) > 0\} \tag{79}$$

and

$$\max_{a \in \chi s \in \mathcal{S}} \{q_z(a \,|\, s)\} \leq \xi < 1. \tag{80}$$

If $\mu_x(C^L(k)) = 0$, we take the node $C^L(k)$ out of the set $T_n^L(k)$ and find the new common ancestor of nodes in $T_n^L(k)$. If $\mu_x(T_n^L(k)) = 0$, we exclude all the nodes in $T_n^L(k)$ from the set $T_n(k)$ and find the new common ancestor of nodes in $T_n(k)$. We repeat this until $\mu_x(C^L(k)) > 0$ is satisfied. Now suppose $P^L(k)$ is of depth l, we have $\mu_x(T_n^L(k)) \geq C_1\tau^{l+1}$ and $\mu_z(T_n^L(k)) \leq C_2\xi^l$, where $C_1 > 0$ and $C_2 > 0$ are constants. Therefore, we have

$$\mu_z\left(T_n^L(k)\right) \leq C_3 \mu_x\left(T_n^L(k)\right)^{\gamma} \tag{81}$$

where $\gamma = \frac{\log \xi}{\log \tau}$ (Note $0 < \gamma < 1$.) Similarly, denote the common ancestor of the nodes in $T_n^R(k)$ by $P^R(k)$, its leftmost child in $P^R(k)$ by $C^R(k)$. Let $\Lambda(P^R(k))$ be the set of nodes including the node $P^R(k)$ and all its offsprings. Then we have $C^R(k) \in T_n^R(k) \subseteq \Lambda(P^R(k))$ and $\mu_x(C^R(k)) > 0$. Thus,

$$\mu_z\left(T_n^R(k)\right) \leq C_4 \mu_x\left(T_n^R(k)\right)^{\gamma}. \tag{82}$$

Combining (81) and (82), we conclude that for any context set $T_n(k)$,

$$\mu_z(T_n(k)) \leq C_5 \mu_x(T_n(k))^{\gamma} \tag{83}$$

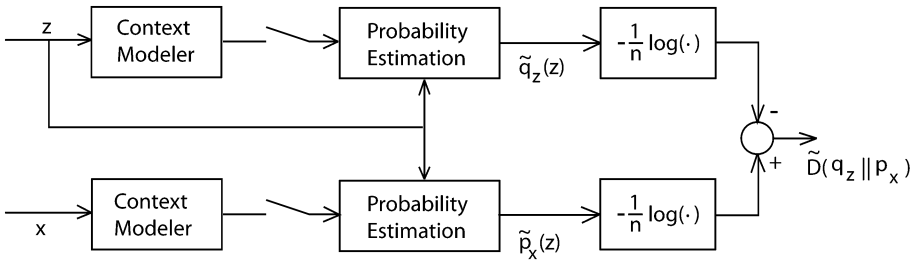

Fig. 6. Divergence estimator via context modeling.

where $C_5 = C_3 + C_4$. Combining (78) and (83), we have

$$\mu_z(M_n(k)) \le \frac{C_5 5^\gamma}{n^{\beta\gamma}}. \tag{84}$$

Finally, we have

$$\begin{aligned}
&\Pr\left\{\frac{N_{\boldsymbol{z}}(k)}{n} \ge \frac{\varepsilon}{\log^3 n}\right\} \\
&\quad\le \Pr\Big\{\{N_{\boldsymbol{x}}(U_n(k)) \le kw(n)\} \\
&\quad\bigcup\{N_{\boldsymbol{x}}(L_n(k)) \ge (k-1)w(n)\}\Big\} \\
&\qquad+ \Pr\Big\{\{N_{\boldsymbol{x}}(U_n(k)) > kw(n)\} \\
&\qquad\bigcap\{N_{\boldsymbol{x}}(L_n(k)) < (k-1)w(n)\} \\
&\qquad\bigcap\left\{\frac{N_{\boldsymbol{z}}(k)}{n} \ge \frac{\varepsilon}{\log^3 n}\right\}\Big\} \qquad (85)\\
&\quad\le \Pr\Big\{\{N_{\boldsymbol{x}}(U_n(k)) > kw(n)\} \\
&\quad\bigcap\{N_{\boldsymbol{x}}(L_n(k)) < (k-1)w(n)\} \\
&\quad\bigcap\left\{\frac{N_{\boldsymbol{z}}(M_n(k))}{n} \ge \frac{\varepsilon}{\log^3 n}\right\}\Big\} \\
&\qquad+ O(n^{2\beta-1}\log^3 n) \qquad (86)\\
&\quad\le \Pr\left\{\frac{N_{\boldsymbol{z}}(M_n(k))}{n} \ge \frac{\varepsilon}{\log^3 n}\right\} \\
&\qquad+ O(n^{2\beta-1}\log^6 n) \qquad (87)\\
&\quad\le \Pr\left\{\frac{N_{\boldsymbol{z}}(M_n(k))}{n} - \mu_z(M_n(k)) \ge \frac{\varepsilon}{\log^3 n} - \frac{C_5 5^\gamma}{n^{\beta\gamma}}\right\} \\
&\qquad+ O(n^{2\beta-1}\log^3 n) \qquad (88)\\
&\quad\le \frac{E\left[\left(\frac{1}{n}N_{\boldsymbol{z}}(M_n(k)) - \mu_z(M_n(k))\right)^2\right]}{\left(\frac{\varepsilon}{\log^3 n} - \frac{C_5 5^\gamma}{n^{\beta\gamma}}\right)^2} + O(n^{2\beta-1}\log^3 n) \\
&\qquad (89)\\
&\quad= O(n^{-1}\log^9 n) + O(n^{2\beta-1}\log^3 n) \qquad (90)\\
&\quad= O(n^{\eta-1}\log^3 n) \qquad (91)
\end{aligned}$$

where (86) follows from (76) and the fact that $N_{\boldsymbol{z}}(M_n(k)) \ge N_{\boldsymbol{z}}(k)$ holds when both $N_{\boldsymbol{x}}(U_n(k)) > kw(n)$ and $N_{\boldsymbol{x}}(L_n(k)) < (k-1)w(n)$ are satisfied; (88) follows from (84); (90) follows from Lemma 5 in Appendix F; and (91) follows from the fact that $0 < \beta < \min\{\frac{\eta}{2}, \frac{1}{2} - \eta\}$. □

IV. Divergence Estimator Via Context Modeling

A. Overview

While it is obvious that the probability estimates fed to the arithmetic coder can be used to estimate entropy, we show that the probability estimation and context model updating mechanisms can also be modified to provide an alternative way to estimate divergence. The structure of the modified algorithm is shown in Fig. 6.

We first build and update context models for both sequences $\boldsymbol{z}$ and $\boldsymbol{x}$. Once the whole sequences $\boldsymbol{z}$ and $\boldsymbol{x}$ have been observed and their modeling has been completed, the switches in Fig. 6 are closed, the estimated models are fixed, and we estimate both $\tilde{q}_{\boldsymbol{z}}(z^n)$ and $\tilde{p}_{\boldsymbol{x}}(z^n)$. We then take the difference of the normalized logarithm of $\tilde{q}_{\boldsymbol{z}}(z^n)$ and $\tilde{p}_{\boldsymbol{x}}(z^n)$ as the divergence estimate $\tilde{D}(q_z \,\|\, p_x)$. The universal context modeler can take many different embodiments leading to different algorithms. In this paper, we focus our attention on the CTW method. Note that an alternative to the divergence estimator in Fig. 6 which leads to slightly degraded estimation consists of generating $\tilde{q}_{\boldsymbol{z}}(z^n)$ "on the fly" at the same time as the context is updated, as is done in arithmetic-coding-based compression where the probability estimate must be computed causally.

B. The Context Tree Weighting Method

The CTW method uses a weighting scheme to provide a weighted probability [15], which is a mixture of estimated probabilities for different models. When the context modeler uses CTW, the lower branch in Fig. 6 is quite similar to the model-freezing method used in [8]. Note that in [8], the idea of freezing the model is important in estimating $\tilde{p}_{\boldsymbol{x}}(z^n)$, since sequence $\boldsymbol{z}$ should not affect the statistical model learned from sequence $\boldsymbol{x}$. Experimental results on text classification using CTW with and without model freezing have been shown in [8].

In this description, for simplicity, the sources are assumed to be binary. The basic CTW method [15] assumes that the maximum memory length D is known and also past symbols x_{1-D}^0 are known in addition to x_1^n. An example of a context tree with $D = 3$ is shown in Fig. 7. Each node in the tree corresponds to a context. Counts a_s and b_s stored in node s are the number of 0's and 1's emitted from the corresponding context. For node s, the estimated probability $P_e^s = P_e(a_s, b_s)$ is the Krichevsky–Trofimov probability estimate of a sequence

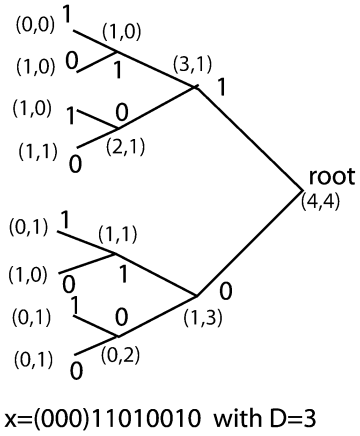

Fig. 7. The basic CTW method with $D = 3$ and $x_{-2}^{8} = 00011010010$. Counts (a_s, b_s) are stored in each node s.

containing a_s zeros and b_s ones, which is defined as follows: $P_e(0,0) = 1$ and for $a \geq 0$ and $b \geq 0$

$$P_e(a+1,b) = \frac{a+\frac{1}{2}}{a+b+1} \cdot P_e(a,b) \tag{92}$$

and

$$P_e(a,b+1) = \frac{b+\frac{1}{2}}{a+b+1} \cdot P_e(a,b). \tag{93}$$

In node s, the conditional estimated probabilities for zero and one are $\frac{a_s+\frac{1}{2}}{a_s+b_s+1}$ and $\frac{b_s+\frac{1}{2}}{a_s+b_s+1}$, respectively. The weighted probability P_w^s of node s is calculated as follows:

$$P_w^s \triangleq \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2}P_w^{0s}P_w^{1s} & : 0 \leq l(s) < D \\ P_e^s & : l(s) = D \end{cases} \tag{94}$$

where the nodes $0s$ and $1s$ are the children of node s, and $l(s)$ is the depth of node s. When we build the context tree from sequence x_1^n, we add one symbol at a time. In adding symbol x_t, we have to update the counts a_s and b_s, the estimated probability P_e^s, and the weighted probability P_w^s for each context s of x_t. The order of updating is from the context of the longest depth (a leaf node) to the root. The limitation is that with fixed maximum memory length D we can only learn statistical models of order no more than D.

The extended CTW method [16] assumes unbounded memory length, and therefore the depth of the context tree is unbounded and grows with the length n. The sources are assumed to be binary. We pre-append $\cdots\varepsilon\varepsilon\varepsilon$ before x_1^n as unknown symbols. The extended CTW method stores all relevant statistical information of different orders provided by the sequence x_1^n. When a certain context has occurred only once in the sequence x_1^n (i.e., there is only one symbol emitted from that context), the counts of the corresponding node sum up to 1, in which case it is meaningless to further store counts of its children. Eventually, we always encounter a context that has never occurred before, because the context εx_1^n has never occurred before.

Let s be a node of the context tree of x_1^n. Node s is said to be a *unique* node of the context tree of x_1^n if the corresponding subsequence occurs only once in $\varepsilon x_1 x_2 \ldots x_n$. Node s is said to be a *null* node if the corresponding subsequence has not occurred in $\varepsilon x_1 x_2 \ldots x_n$.

As we have discussed, it is unnecessary to maintain further children nodes of a unique node. But as we proceed, a unique node at time t might branch out at a later time. So we have to keep the position of the occurrence of the unique subsequence within x_1^t. In building the context tree, when we add the next symbol x_t that has the current context x_1^{t-1} to the context tree, we have to travel from the root, along nodes $x_{t-1}, x_{t-2}^{t-1}, x_{t-3}^{t-1}, \ldots$ until we encounter a null node (which means this node has never occurred before). The path from this null node to the root is called the updating path of the context x_1^{t-1}. After the addition, this null node becomes a unique node (that has occurred once). Then we should travel back to the root, updating the counts, the estimated probability and the weighted probability in every node in the updating path. The context tree and the updating process are shown in Fig. 8. The counts of 0's and 1's are shown in Fig. 9. The weighted probability is defined as follows:

$$P_w^s = \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2}P_w^{0s}P_w^{1s}P_w^{\varepsilon s}, & \text{if } s \text{ is not a unique node} \\ \frac{1}{2}, & \text{if } s \text{ is a unique node.} \end{cases} \tag{95}$$

When processing x_t, only those nodes in the updating path of x_1^{t-1} need to be updated. For a unique node u, $P_w^u = \frac{1}{2}$. For any internal node s in the updating path (including the root), we have to recalculate P_w^s recursively. Notice that $P_w^{\varepsilon s} = 1$ if s is not a prefix of x_1^n (εs is a null node); and $P_w^{\varepsilon s} = \frac{1}{2}$ if s is a prefix of x_1^n.

Define the quantity $\beta^s(x_1^n)$ for node s as follows. For the basic CTW method

$$\beta^s(x_1^n) \triangleq \frac{P_e^s(x_1^n)}{P_w^{0s}(x_1^n)\,P_w^{1s}(x_1^n)} \tag{96}$$

for the extended CTW method

$$\beta^s(x_1^n) \triangleq \frac{P_e^s(x_1^n)}{P_w^{0s}(x_1^n)\,P_w^{1s}(x_1^n)\,P_w^{\varepsilon s}(x_1^n)}. \tag{97}$$

C. Algorithms

Algorithm With Basic CTW:

1) Build the context tree based on the sequence x_1^n using the basic CTW method.
2) Once the modeling is completed, do not allow any changes of the counts, the estimated probability P_e^s, or the weighted probability P_w^s in any node s.
3) For each $i = 1, 2, \ldots, n$, estimate $\tilde{p}_{\boldsymbol{x}}(z_i | z_{i-D}^{i-1})$ based on the above context tree independently, as follows.
   i) Find the node v of depth D corresponding to context z_{i-D}^{i-1}.
   ii) Initialize $P_w^v(z_i \,|\, \cdot) = P_e^v(z_i \,|\, \cdot)$.

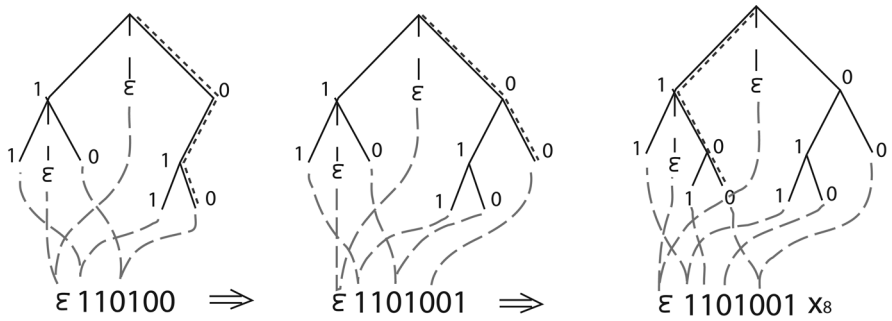

Fig. 8. Context tree updating with $x_1^6 = 110100$ and $x_7 = 1$. Dashed lines point from unique nodes to the occurrence in the sequence. Null nodes are not drawn. Dotted lines are the last updating paths.

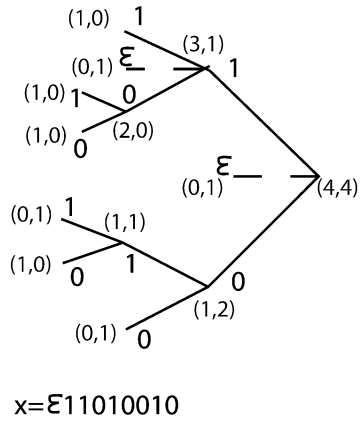

Fig. 9. The extended CTW method with $\varepsilon x_1^8 = \varepsilon 11010010$. Counts (a_s, b_s) are stored in each node s.

iii) Travel from v back to the root. For each node s in the updating path

$$P_w^s(z_i\,|\,\cdot) = \begin{cases} \frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} P_e^s(z_i\,|\,\cdot) + \frac{1}{\beta^s(x_1^n)+1} P_w^{1s}(z_i\,|\,\cdot), \\ \quad \text{if } 1s \text{ is in the updating path} \\ \frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} P_e^s(z_i\,|\,\cdot) + \frac{1}{\beta^s(x_1^n)+1} P_w^{0s}(z_i\,|\,\cdot), \\ \quad \text{if } 0s \text{ is in the updating path.} \end{cases} \tag{98}$$

iv) $\tilde{p}_{\boldsymbol{x}}(z_i|z_{i-D}^{i-1}) = P_w^{\lambda}(z_i\,|\,\cdot)$.

4) Add up the estimates and subtract the entropy estimate (cf. Fig. 6)

$$\begin{aligned} &\tilde{D}(q_z\,\|\,p_x) \\ &\quad = -\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}(z_1^n) + \frac{1}{n}\log_2 \tilde{q}_{\boldsymbol{z}}(z_1^n) \\ &\quad = -\frac{1}{n}\sum_{i=1}^{n} \log_2\left(\tilde{p}_{\boldsymbol{x}}\left(z_i|z_{i-D}^{i-1}\right)\right) + \frac{1}{n}\log_2 \tilde{q}_{\boldsymbol{z}}(z_1^n). \end{aligned} \tag{99}$$

Algorithm With Extended CTW:

1) Build the context tree based on the sequence x_1^n using the extended CTW method.
2) Once the modeling is completed, do not allow any changes of the counts or P_e^s and P_w^s in any node s.
3) For each $i = 1, 2, \ldots, n$, estimate $\tilde{p}_{\boldsymbol{x}}(z_i|z_1^{i-1})$ based on the above context tree independently, as follows.
   i) Find the longest suffix of z_1^{i-1} which occurred as a context in x_1^n, which is equivalent to find the nonnull node of the longest depth corresponding to this context. Let this node be v'.
   ii) Initialize $P_w^{v'}(z_i\,|\,\cdot) = P_e^{v'}(z_i\,|\,\cdot)$.
   iii) Travel from v' back to the root. For each node s in the updating path

$$P_w^s(z_i\,|\,\cdot) = \begin{cases} \frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} P_e^s(z_i\,|\,\cdot) + \frac{P_w^{1s}(z_i\,|\,\cdot)}{\beta^s(x_1^n)+1}, \\ \quad \text{if } 1s \text{ is in the updating path} \\ \frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} P_e^s(z_i\,|\,\cdot) + \frac{P_w^{0s}(z_i\,|\,\cdot)}{\beta^s(x_1^n)+1}, \\ \quad \text{if } 0s \text{ is in the updating path.} \end{cases} \tag{100}$$

   iv) $\tilde{p}_{\boldsymbol{x}}(z_i|z_1^{i-1}) = P_w^{\lambda}(z_i\,|\,\cdot)$.
4) Add up the estimates and subtract the entropy estimate

$$\begin{aligned} \tilde{D}(q_z\,\|\,p_x) &= -\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}(z_1^n) + \frac{1}{n}\log_2 \tilde{q}_{\boldsymbol{z}}(z_1^n) \\ &= -\frac{1}{n}\sum_{i=1}^{n} \log_2\left(\tilde{p}_{\boldsymbol{x}}\left(z_i|z_1^{i-1}\right)\right) + \frac{1}{n}\log_2 \tilde{q}_{\boldsymbol{z}}(z_1^n). \end{aligned} \tag{101}$$

It is suggested in [16] that a string of nonunique nodes without branches except the farthest one from the root are equivalent and can be replaced by a single super-node. Equivalent nodes all have the same counts and therefore the same estimated probability. Only the weighted probability corresponding to the node closest to the root is actually needed by its parent and therefore stored in the super-node. Suppose the number of nodes in the super-node s is $d(s)$ and P_w^s is the weighted probability stored in the super-node s. We have

$$\begin{aligned} P_w^s(x_1^n) &= \left(1 - \frac{1}{2^{d(s)}}\right) P_e^s(x_1^n) \\ &\quad + \frac{1}{2^{d(s)}} P_w^{0s}(x_1^n) P_w^{1s}(x_1^n) P_w^{\varepsilon s}(x_1^n) \end{aligned} \tag{102}$$

$$\beta^s(x_1^n) = \frac{(2^{d(s)} - 1) P_e^s(x_1^n)}{P_w^{0s}(x_1^n) P_w^{1s}(x_1^n) P_w^{\varepsilon s}(x_1^n)} \tag{103}$$

and

$$\frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} = \frac{\left(1 - 2^{-d(s)}\right) P_e^s(x_1^n)}{P_w^s(x_1^n)}. \tag{104}$$

Then the updating rule (100) in Step 3 iii) holds for super-nodes. The compact context tree implementation apparently has the advantage of using fewer nodes and fewer operations.

D. Analysis

It is easy to prove the convergence of the entropy estimator via CTW. The cross term estimator is more problematic and so we focus on the estimation of the cross term via the basic CTW method.

After building the context tree based on the sequence x_1^n, we proceed to estimate $-\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}(z_1^n)$ by estimating n terms

$$\tilde{p}_{\boldsymbol{x}}(z_i|z_{i-D}^{i-1}) = P_w^{\lambda}(z_i\,|\cdot), \qquad i = 1, 2, \ldots, n.$$

In order to bound the aggregate error

$$\left| -\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}\left(z_1^n\right) + \frac{1}{n}\log_2 p_x\left(z_1^n\right) \right|$$

we bound the error

$$\left|\log_2 \tilde{p}_{\boldsymbol{x}}\left(z_i|z_{i-D}^{i-1}\right) - \log_2 p_x\left(z_i|z_{i-D}^{i-1}\right)\right|$$

for any $z_i \in \chi$ and $z_{i-D}^{i-1} \in \chi^D$.

The estimate $\tilde{p}_{\boldsymbol{x}}(z_i|z_{i-D}^{i-1})$ is obtained by using the same context tree updating rule, where z_i is the next symbol and z_{i-D}^{i-1} is the current context. If we were not freezing the context tree, $P_w^{\lambda}(x_1^n)$ would be updated to include the new incoming symbol z_i and would become $P_w^{\lambda}(x_1^n, z_i)$. The conditional weighted probability at the root $P_w^{\lambda}(z_i|z_{i-D}^{i-1}) = P_w^{\lambda}(x_1^n, z_i)/P_w^{\lambda}(x_1^n)$ is the probability estimate we need. We emphasize that once the context tree modeling is completed, we do not change any counts (a_s, b_s) or P_e^s and P_w^s stored in the context tree; i.e., we just need to obtain $P_w^{\lambda}(x_1^n, z_i)$ by computing $P_w^s(x_1^n, z_i)$ along the updating path of z_{i-D}^{i-1}.

Our main consistency result for the algorithm in Section IV-C is the following.

Theorem 2: Let $\boldsymbol{z}$ and $\boldsymbol{x}$ be sequences of length n generated from finite-alphabet finite-state Markov sources q_z and p_x, respectively. Let $\tilde{D}_n(q_z\,||\,p_x)$ denote our divergence estimator based on the basic CTW method whose maximum memory length is greater than or equal to the orders of both Markov sources. Then

$$\lim_{n\to\infty} \tilde{D}_n(q_z\,||\,p_x) = D(q_z\,||\,p_x) \quad \text{a.s.} \tag{105}$$

Proof: We prove the convergence of the estimate of the cross term. The same argument applies to the entropy term.

Let us examine the updating computation for z_i, where $1 \le i \le n$. For an internal node s in the updating path, if $1s$ is in the updating path we have (106) at the bottom of the page; if $0s$ is in the updating path we get (107), also at the bottom of the page. For a leaf node v in the updating path

$$P_w^v\left(z_i|z_{i-D}^{i-1}\right) = P_e^v\left(z_i|z_{i-D}^{i-1}\right). \tag{108}$$

This computation starts with a leaf and is repeated recursively along the updating path, until we reach the root and obtain $P_w^{\lambda}(z_i|z_{i-D}^{i-1})$. So $P_w^{\lambda}(z_i|z_{i-D}^{i-1})$ is a weighted sum of $P_e^s(z_i|z_{i-D}^{i-1})$, where s is any node in the updating path. Let $\{s \to \lambda\}$ denote the set of nodes in the path from s to the root. The weight associated with $P_e^s(z_i|z_{z-D}^{z-1})$ is

$$\beta^s(x_1^n) \prod_{u\in\{s\to\lambda\}} \frac{1}{\beta^u(x_1^n)+1}$$

where s is an internal node in the updating path. The weight associated with $P_w^v(z_i|z_{i-D}^{i-1})$, where v is the leaf in the updating path, is

$$\prod_{u\in\{\{v\to\lambda\}\setminus v\}} \frac{1}{\beta^u(x_1^n)+1}.$$

Suppose $z_{i-t_i}^{i-1}$ is the suffix of z_{i-D}^{i-1} that is a state of the finite memory source, where t_i is less than D (the upper bound on the memory length). Suppose $s^0, s^1, \ldots, s^{t_i-1}$ are the t_i nodes in the updating path with depth less than t_i (including the root λ). Since $s^0, \ldots, s^{t_i-1}$ have depth less than $t_i, P_e^{s^0}(z_i\,|\cdot), \ldots, P_e^{s^{t_i-1}}(z_i\,|\cdot)$ do not converge to $p_x(z_i|z_{i-D}^{i-1})$ and should not be included in the weighted conditional probability. Lemma 4 below implies that the contribution of $P_e^{s^0}(z_i\,|\cdot), \ldots, P_e^{s^{t_i-1}}(z_i\,|\cdot)$ to the weighted conditional probability $P_w^{\lambda}(z_i|z_{i-D}^{i-1})$ converges to zero. On the other hand, $P_e^{s^{t_i}}(z_i\,|\cdot), \ldots, P_e^{s^D}(z_i\,|\cdot)$ all converge to $p_x(z_i|z_{i-D}^{i-1})$ almost surely, therefore, the weighted conditional probability $P_w^{\lambda}(z_i\,|\cdot)$ converges to $p_x(z_i|z_{i-D}^{i-1})$ almost surely as $n \to \infty$. Thus, we have

$$\left| -\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}\left(z_1^n\right) + \frac{1}{n}\log_2 p_x\left(z_1^n\right) \right| \to 0 \tag{109}$$

$$\begin{aligned} P_w^s\left(z_i\,|\,z_{i-D}^{i-1}\right) &= \frac{P_w^s\left(x_1^n, z_i\right)}{P_w^s\left(x_1^n\right)} \\ &= \frac{\frac{1}{2}P_e^s\left(x_1^n\right)P_e^s\left(z_i|z_{i-D}^{i-1}\right) + \frac{1}{2}P_w^{0s}\left(x_1^n\right)P_w^{1s}\left(x_1^n\right)P_w^{1s}\left(z_i\,|\,z_{i-D}^{i-1}\right)}{P_w^s\left(x_1^n\right)} \\ &= \frac{\beta^s\left(x_1^n\right)}{\beta^s\left(x_1^n\right)+1}P_e^s\left(z_i|z_{i-D}^{i-1}\right) + \frac{1}{\beta^s\left(x_1^n\right)+1}P_w^{1s}\left(z_i|z_{i-D}^{i-1}\right) \end{aligned} \tag{106}$$

$$\begin{aligned} P_w^s\left(z_i|z_{i-D}^{i-1}\right) &= \frac{\frac{1}{2}P_e^s\left(x_1^n\right)P_e^s\left(z_i|z_{i-D}^{i-1}\right) + \frac{1}{2}P_w^{0s}\left(x_1^n\right)P_w^{1s}\left(x_1^n\right)P_w^{0s}\left(z_i|z_{i-D}^{i-1}\right)}{P_w^s\left(x_1^n\right)} \\ &= \frac{\beta^s\left(x_1^n\right)}{\beta^s\left(x_1^n\right)+1}P_e^s\left(z_i|z_{i-D}^{i-1}\right) + \frac{1}{\beta^s\left(x_1^n\right)+1}P_w^{0s}\left(z_i|z_{i-D}^{i-1}\right). \end{aligned} \tag{107}$$

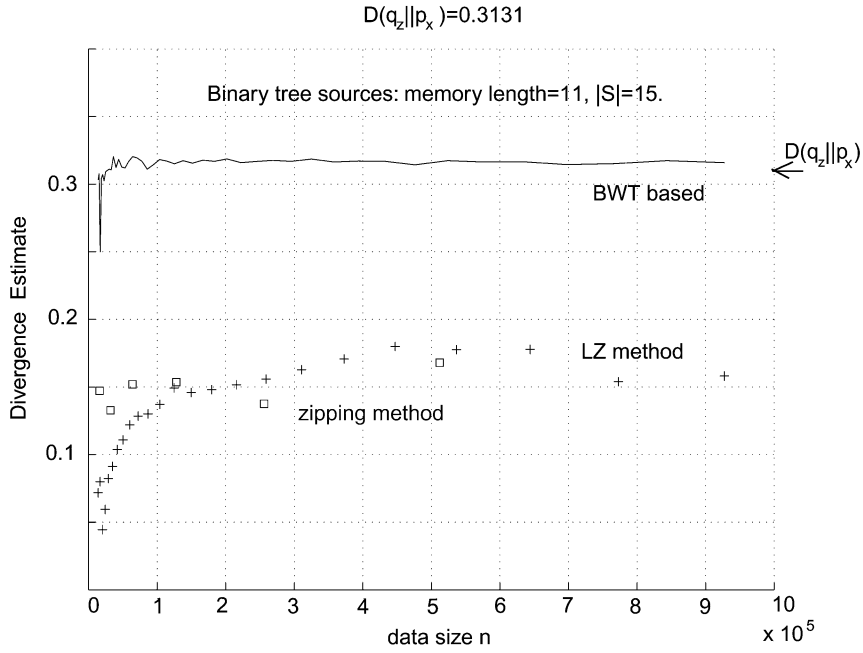

Fig. 10. Comparison of the methods in [2] (zipping) and [17] (LZ) with the new BWT-based method.

almost surely. Combined with the fact that $-\frac{1}{n}\log_2 p_x(z_1^n) \to S(q_z \,\|\, p_x)$ almost surely, we have

$$\lim_{n\to\infty} -\frac{1}{n}\log_2 \tilde{p}_{\boldsymbol{x}}(z_1^n) = S(q_z \,\|\, p_x) \quad \text{a.s.} \tag{110}$$

Applying the same argument to the estimate of the entropy term, we obtain (105). □

Lemma 4: Suppose s is an internal node in the tree representation of the source, but s itself is not a state of the source (i.e., offsprings of the node s do not all have the same conditional probabilities). Then

$$\lim_{n\to\infty} \beta^s(x_1^n) = 0 \quad \text{a.s.} \tag{111}$$

Proof:

$$\begin{aligned}\frac{\beta^s(x_1^n)}{\beta^s(x_1^n)+1} &= \frac{P_e^s(x_1^n)}{2P_w^s(x_1^n)} \\ &\le \frac{4P_e^s(x_1^n)}{P_e^{0s}(x_1^n)\,P_e^{1s}(x_1^n)} \\ &= 4\exp\left\{n\cdot\left(\frac{1}{n}\log P_e^s(x_1^n)\right.\right. \\ &\qquad \left.\left. -\frac{1}{n}\log P_e^{0s}(x_1^n)\,P_e^{1s}(x_1^n)\right)\right\} \end{aligned} \tag{112}$$

where the inequality follows from applying (94) repeatedly. Due to the log sum inequality, we have

$$\frac{1}{n}\log P_e^s(x_1^n) - \frac{1}{n}\log P_e^{0s}(x_1^n)\,P_e^{1s}(x_1^n) \to c_s < 0 \tag{113}$$

almost surely as $n \to \infty$, where c_s is a negative constant since s is not a state, and its children nodes must have different statistics. Using (113) in the bound (112) we obtain (111). □

V. Experimental Results

In Fig. 10, we compare our BWT-based algorithm with the LZ string-matching-based algorithm [17] as well as the zipping method [2]. The sources we use are randomly generated binary tree sources, with memory length 11, and 15 states. All the curves plotted are an average of 100 runs. As shown in Fig. 10, the new algorithm converges quite fast to the true divergence. In contrast, for the data sizes considered, neither the LZ-based algorithm nor the zipping method are able to offer good estimates. Although not shown in Fig. 10, the estimate of the zipping method exhibits high sensitivity to the actual source realization.

In Fig. 11, we compare our BWT-based algorithm with the empirical plug-in scheme that assumes a Markov model of a given order. Even when the memory length is known, the plug-in scheme does not perform as well as our scheme, and moreover, it suffers considerable degradation if it either underestimates or overestimates the order. Fig. 11 also shows that the performance difference for adaptive and uniform segmentation is negligible.

In the empirical plug-in scheme, unless prior knowledge is available about the tree structure of the sources, the number of transition probabilities to be estimated grows exponentiallywith the assumed order of the source. 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 Fig. 12, we consider the effect of dependence between the realizations $\boldsymbol{x}$ and $\boldsymbol{z}$ on the divergence estimate. The sources tested in this figure have the same distributions as the sources in Fig. 10. In particular, two extreme cases are tested. In the

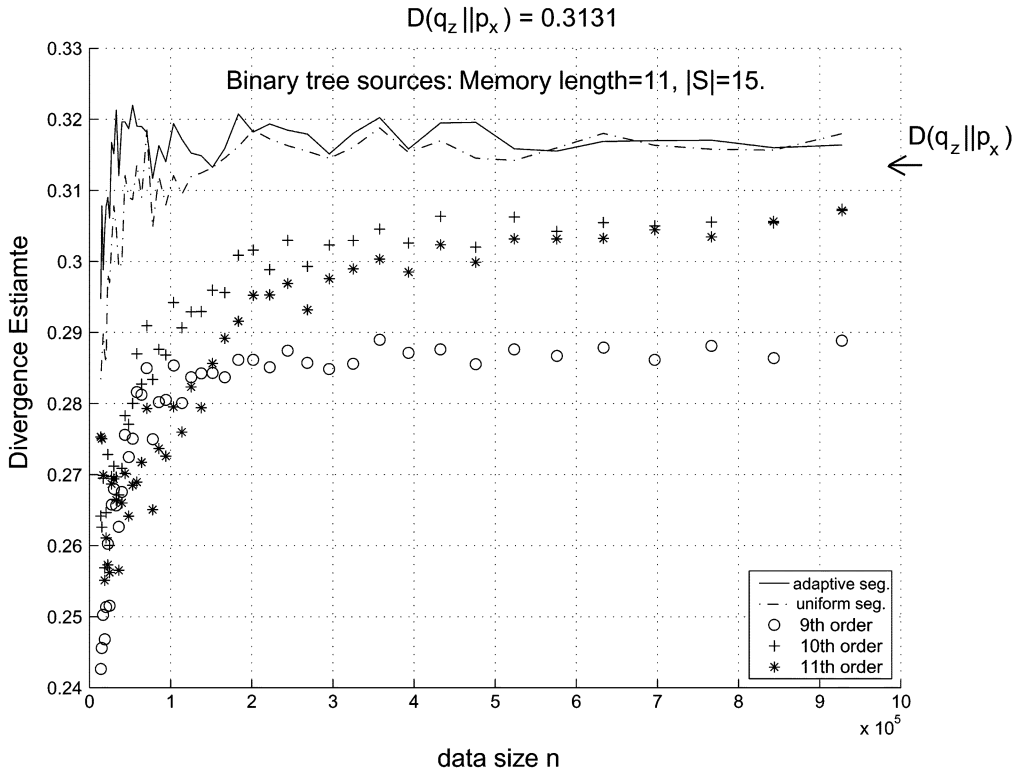

Fig. 11. Divergence estimator via the BWT and the plug-in estimator.

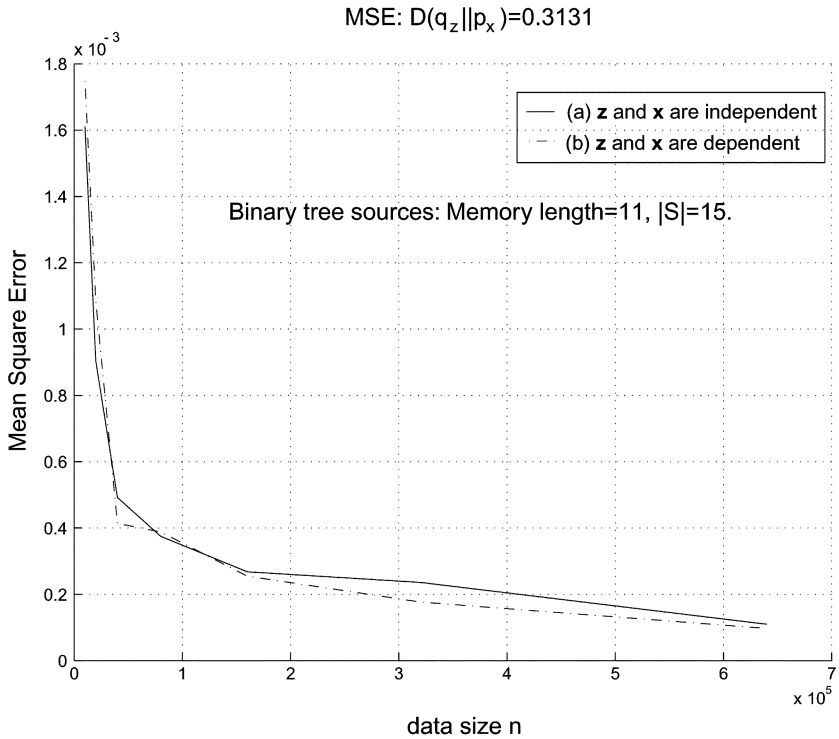

Fig. 12. Mean square error of the BWT-based divergence estimator with uniform segmentation: (a) $\boldsymbol{z}$ and $\boldsymbol{x}$ are independent realizations. (b) $\boldsymbol{z}$ is obtained by complementing $\boldsymbol{x}$. The sources generating $\boldsymbol{z}$ have the identical distributions in (a) and (b).

first case, the sequences $\boldsymbol{x}$ and $\boldsymbol{z}$ are independently generated. In the second case, $\boldsymbol{z}$ is generated by complementing $\boldsymbol{x}$. Since we chose the sources p_x and q_z to have mirror trees, the sequence $\boldsymbol{z}$ so generated does indeed have distribution q_z. The experiment shows that our divergence estimator is rather insensitive to the joint distribution, except through the marginals.

In Figs. 13 and 14, we compare the BWT-based algorithm and the basic CTW-based algorithm (with $D = 16$) for the same sources tested in Fig. 10. While the divergence estimator via CTW exhibits slightly faster convergence, the divergence estimator via the BWT is easier to implement.

Our divergence estimators can be used in linguistic problems such as language classification and author recognition.

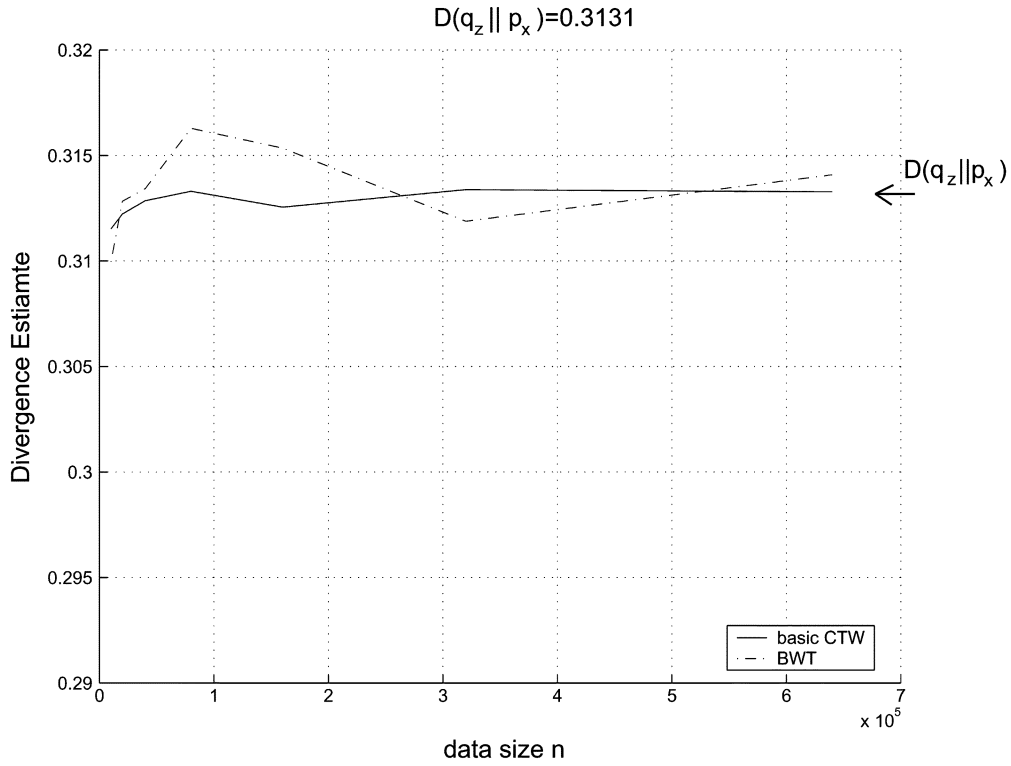

Fig. 13. Divergence estimator based on the BWT/CTW: $D(q_z \,\|\, p_x) = 0.3131$.

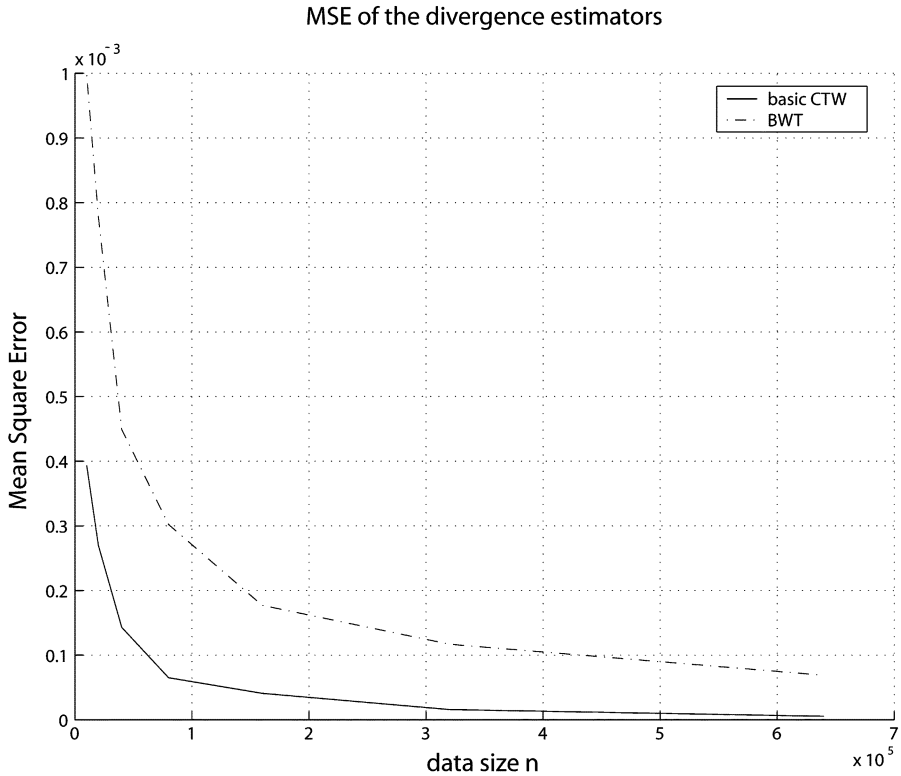

Fig. 14. Mean-square error of the divergence estimator based on the BWT/CTW: $D(q_z \,\|\, p_x) = 0.3131$.

Although having an estimator for the divergence does not necessarily lead to a classifier that minimizes the classification error, our divergence estimators yield satisfactory experimental results. First, the divergence estimator is used to classify texts written in different languages. The text files we use are the Bible (Old Testament) translated into 11 languages. The divergence table (based on the estimates via the BWT) of each pair of languages is shown in Fig. 15. (The divergence estimator via CTW gives similar estimates, generally within 10%.) Then we map the symmetrized distance matrix to a language tree using the

D(q||p) x / z	SPA	POR	FRE	ITA	ENG	GER	DUT	SWE	DAN	NOR	ICE
Spanish	/	1.58	1.98	2.14	2.73	3.12	3.01	2.77	2.70	2.71	3.08
Portuguese	1.40	/	2.11	2.02	2.79	3.11	3.02	2.80	2.68	2.70	2.92
French	2.17	2.34	/	2.34	2.34	2.77	2.76	2.73	2.50	2.45	2.93
Italian	1.85	1.94	2.16	/	2.65	3.23	3.34	2.90	2.81	2.84	3.02
English	3.01	3.49	2.75	3.24	/	2.62	2.91	2.94	2.73	2.73	3.23
German	2.95	3.46	2.66	3.09	2.50	/	2.05	2.42	2.45	2.49	2.77
Dutch	2.99	3.64	2.85	3.37	2.67	2.08	/	2.49	2.31	2.55	3.20
Swedish	2.79	3.13	2.64	2.82	2.34	2.11	2.26	/	1.48	1.20	1.97
Danish	2.77	3.01	2.36	2.73	2.29	1.95	2.03	1.10	/	0.49	1.81
Norwegian	2.85	3.16	2.41	2.77	2.36	2.06	2.14	0.99	0.45	/	1.63
Icelandic	2.95	3.19	2.81	2.85	2.90	2.50	2.80	1.97	1.99	1.79	/

Fig. 15. Divergence between languages (Bible) estimated by the BWT-based method.

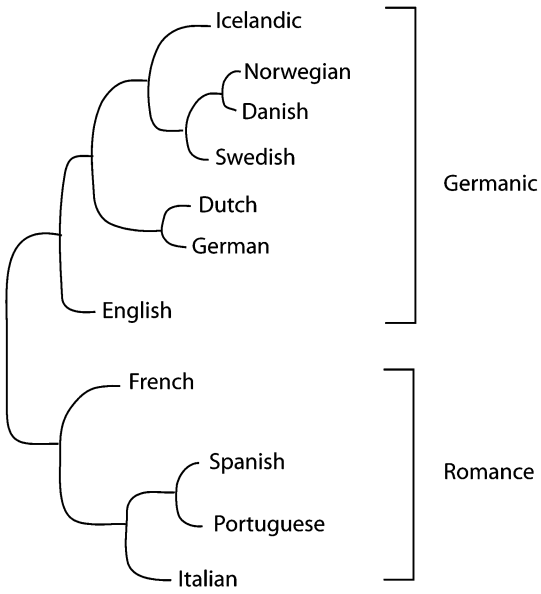

Fig. 16. Language tree derived from the divergence estimates of Fig. 15.

Fitch–Margoliash method in the package PhylIP [9] for inferring evolutionary trees. Our algorithm successfully recognizes major language groups, such as Romance and Germanic (see Fig. 16). Note that according to our algorithm, English is a Germanic language, whereas according to the algorithm in [2], English is closer to the group of Romance languages. The experimental results of [2] were based on "The Universal Declaration of Human Rights" in 50 different languages.

We have also used the divergence estimator for discriminating between authors. We run the divergence estimator on nine English novels (or English translations). One of them is *Gadsby*, a unique novel without a single letter "e" written by Ernest Vincent Wright in 1939. As shown in Fig. 17, most divergence estimates are in the range of 0.15 to 0.34, except $\bar{D}(\text{Gadsby}\|\cdot)$ which are between 0.6 and 0.8, and $\bar{D}(\cdot\|\text{Gadsby})$ which are between 1.2 and 1.4. In fact, $D(\cdot\|\text{Gadsby})$ should be infinity, because *Gadsby* does not have a single "e," but the letter "e" has a frequency of about 10% in English (and in particular has nonzero frequency in the other eight novels). However, in the case of infinite divergence, our estimator will not output infinity because we use a bias $\Delta > 0$ in the probability estimates. But the divergence estimates will grow roughly at the rate of $C \log_2 \frac{\sqrt{n}}{\Delta}$. So either increasing the sequence length n or decreasing Δ will cause the estimates to increase without bound. In contrast, note that when divergence is finite, the estimate is insensitive to shrinkage of Δ.

Reassuringly, novels by the same author have relatively small divergence, such as *Anna Karenina* and *War and Peace* by Leo Tolstoy; and the four novels by Jane Austen. Interestingly, the divergence is also comparatively small for the novels written by the Bronte sisters: (*Jane Eyre* by Charlotte Bronte and *Wuthering Heights* by Emily Bronte).

Appendix A

Let $f = p_1^2 + p_2^2 + p_3^2$. It follows from the two constraints (26), (27) that

$$dp_1 + dp_2 + dp_3 = 0 \tag{114}$$

and

$$(1 + \log p_1)dp_1 + (1 + \log p_2)dp_2 + (1 + \log p_3)dp_3 = 0. \tag{115}$$

Therefore,

$$\begin{aligned} df &= 2p_1 dp_1 + 2p_2 dp_2 + 2p_3 dp_3 \\ &= 2(p_1 \log p_2 - p_1 \log p_3 + p_2 \log p_3 - p_2 \log p_1 \\ &\quad + p_3 \log p_1 - p_3 \log p_2) dp_1 / (\log p_2 - \log p_3). \end{aligned} \tag{116}$$

Author \ D(q‖p) \ z	x	Gadsby	Anna	War	Jane	Wuth-ering	Pride	Mans-field	Sense	Emma
E. Wright	Gadsby (1939)	/	0.69	0.65	0.61	0.65	0.78	0.72	0.76	0.76
L. Tolstoy	Anna Karenina (1877)	1.31	/	0.18	0.26	0.23	0.31	0.29	0.32	0.30
	War and Peace (1869)	1.34	0.20	/	0.27	0.26	0.34	0.32	0.34	0.34
C. Bronte	Jane Eyre (1846)	1.26	0.22	0.20	/	0.13	0.24	0.20	0.23	0.23
E. Bronte	Wuthering Heights (1847)	1.32	0.23	0.22	0.16	/	0.26	0.24	0.25	0.26
J. Austen	Pride and Prejudice (1813)	1.45	0.30	0.29	0.25	0.24	/	0.18	0.19	0.19
	Mansfield Park (1814)	1.32	0.26	0.25	0.20	0.21	0.17	/	0.17	0.16
	Sense & Sensibility (1811)	1.39	0.28	0.27	0.22	0.21	0.16	0.15	/	0.17
	Emma (1816)	1.38	0.28	0.28	0.24	0.22	0.18	0.17	0.18	/

Fig. 17. Divergence between novels estimated by the BWT-based method.

When $p_1 > p_2 > p_3$, we shall prove $\frac{df}{dp_1} > 0$ in the following. We have

$$\frac{df}{dp_1} = \frac{2g(p_1,p_2,p_3)}{\log p_2 - \log p_3} \tag{117}$$

where

$$g(p_1,p_2,p_3) = p_1\log p_2 - p_1\log p_3 + p_2\log p_3 - p_2\log p_1 + p_3\log p_1 - p_3\log p_2. \tag{118}$$

We have $g(p_1,p_2,p_3) = 0$ when $p_1 = p_2$. Let

$$h(p_1,p_2,p_3) \triangleq \frac{\partial g(p_1,p_2,p_3)}{\partial p_1} = \log p_2 - \log p_3 - \frac{p_2}{p_1} + \frac{p_3}{p_1}. \tag{119}$$

We have $h(p_1,p_2,p_3) = 0$ when $p_2 = p_3$, and

$$\frac{\partial h(p_1,p_2,p_3)}{\partial p_2} = \frac{1}{p_2} - \frac{1}{p_1} > 0. \tag{120}$$

Therefore, $h(p_1,p_2,p_3) > 0$ and $g(p_1,p_2,p_3) > 0$ if $p_1 > p_2 > p_3$. Hence, f is an increasing function of p_1, and is maximized when either $p_1 = 1$ or $p_1 > p_2 = p_3$.

Appendix B

Let $\alpha = \alpha_2 + \alpha_3$ and $\beta_2 = \frac{\alpha_2}{\alpha_2+\alpha_3}$. We rewrite (28) as follows:

$$G_3(\bar{\bar{p}}, E) = \max_{p_2,p_3,\alpha,\beta_2} \{(1-\alpha) + \alpha\left(\beta_2 p_2^2 + (1-\beta_2)p_3^2\right) - \bar{\bar{p}}^2\} \tag{121}$$

under the constraints

$$(1-\alpha) + \alpha(\beta_2 p_2 + (1-\beta_2)p_3) = \bar{\bar{p}}, \tag{122}$$

$$\alpha(-\beta_2 p_2\log p_2 - (1-\beta_2)p_3\log p_3) = -\bar{\bar{p}}\log\bar{\bar{p}} - E \tag{123}$$

and $1 \geq \alpha \geq 0, 1 \geq \beta_2 \geq 0$, and $1 \geq p_2 \geq 0, 1 \geq p_3 \geq 0$.

We first keep α fixed and maximize over β_2, p_2 and p_3. Define

$$G_2(p',y') \triangleq \max_{p_2,p_3,\beta_2} \left\{\beta_2 p_2^2 + (1-\beta_2)p_3^2\right\} \tag{124}$$

under the constraints

$$\beta_2 p_2 + (1-\beta_2)p_3 = p' \tag{125}$$

$$-\beta_2 p_2\log p_2 - (1-\beta_2)p_3\log p_3 = y' \tag{126}$$

and $1 \geq \beta_2 \geq 0, 1 \geq p_2 \geq 0, 1 \geq p_3 \geq 0$.

We can rewrite $G_2(p',y') = \max_\theta G(p',y',\theta)$ as an optimization problem over θ. Notice that $G(p',y',\theta) = (p_2 + p_3)p' - p_2p_3$, where p_2 and p_3 are determined by p', y', and θ through the constraints (see Fig. 18). Since p_2 and p_3 are two intersections of the line $y = \tan\theta(x-p') + y'$ and the curve $y = -x\log x$, they are two solutions of the following equation:

$$x\log x + \tan\theta(x-p') + y' = 0. \tag{127}$$

We have

$$dp_2 = -\frac{p_2 - p'}{\cos^2\theta(1+\log p_2 + \tan\theta)}d\theta \tag{128}$$

and

$$dp_3 = -\frac{p_3 - p'}{\cos^2\theta(1+\log p_3 + \tan\theta)}d\theta \tag{129}$$

Hence,

$$\frac{dG}{d\theta} = (p'-p_3)\frac{dp_2}{d\theta} + (p'-p_2)\frac{dp_3}{d\theta} = -\frac{(p_2-p')(p'-p_3)}{\cos^2\theta} \times \left(\frac{1}{\tan\theta+1+\log p_2} + \frac{1}{\tan\theta+1+\log p_3}\right) \tag{130}$$

where $\tan\theta = \frac{-p_2\log p_2 + p_3\log p_3}{p_2-p_3}$. Without loss of generality, we assume $p_2 > p' > p_3$. In Appendix C, we prove that

$$\tan\theta + 1 + \log p_2 > 0 \tag{131}$$

$$\tan\theta + 1 + \log p_3 < 0 \tag{132}$$

and

$$\tan\theta + 1 + \log p_2 < -(\tan\theta + 1 + \log p_3). \tag{133}$$

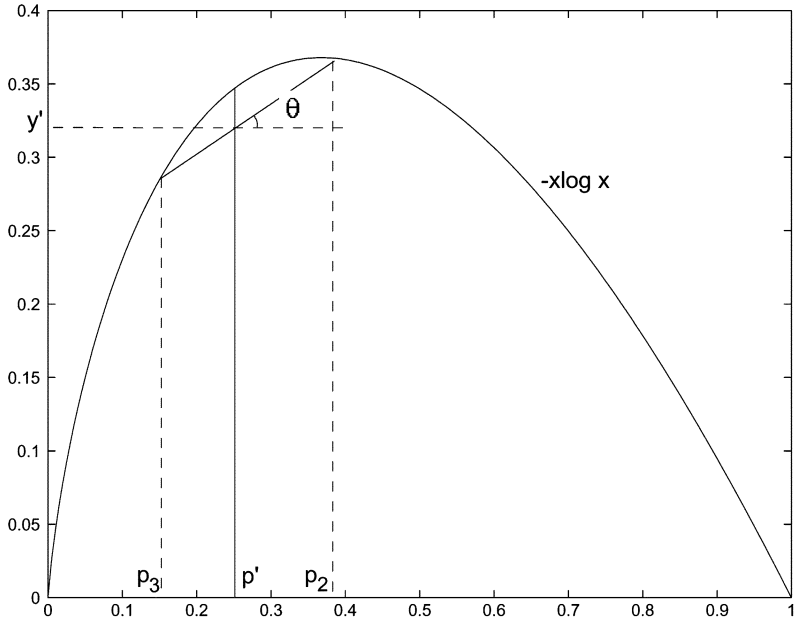

Fig. 18. Maximizing $G(p', y', \theta) = (p_2 + p_3)p' - p_2 p_3$ over θ, where p_2 and p_3 are two intersections of the line $y = \tan\theta(x - p') + y'$ and the curve $y = -x \log x$.

From (130), (131), (132), and (133), we have

$$\frac{dG}{d\theta} < 0. \tag{134}$$

Therefore, the maximum $G_2(p', y') = \max_\theta G(p', y', \theta)$ is achieved when $\tan\theta = \frac{-y'}{1-p'}$ (in which case $p_2 = 1$ and $p_3 < p'$ satisfying $p_3 \log p_3 + \frac{-y'}{1-p'}(p_3 - p') + y' = 0$). From the solution of the optimization problem (124), we see that (121) is maximized when $p_2 = 1$ (or, equivalently, $\beta_2 = 0$).

Appendix C

Under the assumption $p_2 > p_3$, we want to prove the inequalities

$$1 + \log p_2 + \frac{-p_2 \log p_2 + p_3 \log p_3}{p_2 - p_3} > 0 \tag{135}$$

$$1 + \log p_3 + \frac{-p_2 \log p_2 + p_3 \log p_3}{p_2 - p_3} < 0 \tag{136}$$

and

$$2 + \log p_2 + \log p_3 + \frac{2(-p_2 \log p_2 + p_3 \log p_3)}{p_2 - p_3} < 0. \tag{137}$$

We have

$$\frac{p_3}{p_2 - p_3}\left(\frac{p_2}{p_3} - 1 - \log\frac{p_2}{p_3}\right) > 0 \tag{138}$$

and

$$-\frac{p_2}{p_2 - p_3}\left(\frac{p_3}{p_2} - 1 - \log\frac{p_3}{p_2}\right) < 0 \tag{139}$$

since $p_2 > p_3$ and $x - 1 - \log x > 0$ holds for all $x > 0$ and $x \neq 1$. Let

$$f(x) = 2(x - p_3) - (x + p_3)(\log x - \log p_3). \tag{140}$$

We have $f(p_3) = 0$ and for all $x > p_3$

$$f'(x) = \log\frac{p_3}{x} + 1 - \frac{p_3}{x} < 0. \tag{141}$$

Therefore, $f(p_2) < 0$. So we have

$$\frac{2(p_2 - p_3) - (p_2 + p_3)(\log p_2 - \log p_3)}{p_2 - p_3} < 0. \tag{142}$$

Appendix D

In order to prove (38), we need to prove that

$$E\left[\left(\frac{N_{\boldsymbol{z}}(a, s)}{n} - q_z(s)q_z(a \mid s)\right)^2\right] = O\left(\frac{1}{n}\right). \tag{143}$$

Notice that $S'_1, S'_2, \ldots$ form a Markov chain with the stationary distribution $q_z(a, s) = q_z(s)q_z(a \mid s)$, where $S'_i = (S_i, Z_i), i = 1, 2, \ldots, n$.

$$N_{\boldsymbol{z}}(a, s) = \sum_{i=1}^{n} \mathbf{1}_{\{S_i = s, Z_i = a\}} = \sum_{i=1}^{n} \mathbf{1}_{\{S'_i = (s,a)\}}. \tag{144}$$

Then, (143) follows from the fact [5] that

$$E\left[\left(\frac{N_{\boldsymbol{z}}(s')}{n} - q_z(s')\right)^2\right] = O\left(\frac{1}{n}\right). \tag{145}$$

APPENDIX E

Under the assumption that the Markov chain is irreducible and aperiodic, there exists an integer $k_0 = d_0 + c_0$, where d_0 is the order of the Markov source and $c_0 \geq 0$ is a constant, such that

$$\max_{s_i, s_j \in \mathcal{S}, A \subset \mathcal{S}} \left| p_x^{(k_0)}(s_i, A) - p_x^{(k_0)}(s_j, A) \right| = \zeta < 1 \tag{146}$$

where $p_x^{(k_0)}(s_i, A)$ is the k_0-step transition probability. It follows that (see [6], [14]) for any $s \in \mathcal{S}$ and any $A \subset \mathcal{S}$

$$\left| p_x^{(n)}(s, A) - \mu_x(A) \right| \leq \zeta^{(n/k_0)-1} = \gamma_0 \rho^n \tag{147}$$

where $\rho = \zeta^{1/k_0}$ and $\gamma_0 = \zeta^{-1}$.

For arbitrary initial states s_i and s_j, we have

$$\begin{aligned} \max_{A \subset \chi^l} & \left| p_x^{(l+c_0)}(s_i, A) - p_x^{(l+c_0)}(s_j, A) \right| \\ &= \max_{A \subset \mathcal{S}} \left| p_x^{(d_0+c_0)}(s_i, A) - p_x^{(d_0+c_0)}(s_j, A) \right| \end{aligned} \tag{148}$$

where $l \geq d_0$. It follows that

$$\max_{s_i', s_j' \in \chi^l, A \subset \chi^l} \left| p_x^{(l+c_0)}(s_i', A) - p_x^{(l+c_0)}(s_j', A) \right| = \zeta < 1. \tag{149}$$

Therefore, for any $s' \in \chi^l$ and any $A' \subset \chi^l$, we have

$$\left| p_x^{(n)}(s', A') - \mu_x(A') \right| \leq \zeta^{\frac{n}{l+c_0}-1} = \gamma_0 \rho_l^{\,n} \tag{150}$$

where $\rho_l = \zeta^{\frac{1}{l+c_0}}$ and $\gamma_0 = \zeta^{-1}$.

APPENDIX F

Lemma 5: Suppose $\boldsymbol{x}$ is a sequence of length n generated from Markov source p_x with alphabet χ and order d_0. The Markov chain is assumed to be irreducible and aperiodic. Let W be a set of nodes in the context tree that have depths no more than $l = o(n)$. Assume W is equivalent to a set of consecutive nodes at depth l in the context tree. Let $N_{\boldsymbol{x}}(W)$ be the number of symbols in the sequence $\boldsymbol{x}$ that have a context in W and $\mu_x(W)$ be the stationary probability of W. Then we have

$$E\left[\left(\frac{1}{n} N_{\boldsymbol{x}}(W) - \mu_x(W) \right)^2 \right] = O\left(\frac{l^3}{n} \right). \tag{151}$$

Proof: We first find the upper bound of the number of nodes in W, where the children nodes in W are replaced by the parent node, if all children nodes of a node (of depth at least d_0) are included in W. In addition, W is assumed to consist of consecutive nodes. Thus, the number of nodes in W is upper-bounded by $2|\chi|l + |\chi|^{d_0}$, since for each depth larger than d_0 there are at most $2|\chi|$ nodes included in W. Therefore, we have

$$\begin{aligned} E\left[\left(\frac{1}{n} N_{\boldsymbol{x}}(W) - \mu_x(W) \right)^2 \right] &\leq (2|\chi|l + |\chi|^{d_0})^2 \\ &\quad \max_{s \in \bigcup_{d_0 \leq m \leq l} \chi^m} E\left[\left(\frac{1}{n} N_{\boldsymbol{x}}(s) - \mu_x(s) \right)^2 \right]. \end{aligned} \tag{152}$$

Now suppose s is any node of depth m, $d_0 \leq m \leq l$. We have

$$\begin{aligned} E\left[\left(\frac{1}{n} N_{\boldsymbol{x}}(s) - \mu_x(s) \right)^2 \right] &= \frac{\mu_x(s) - \mu_x^2(s)}{n} \\ &\quad + \frac{2\mu_x(s)}{n^2} \sum_{d=1}^{n-1} (n-d) \left(p_x^{(d)}(s, s) - \mu_x(s) \right). \end{aligned} \tag{153}$$

It follows from (150) in Appendix E that

$$\sum_{d=1}^{\infty} |p_x^{(d)}(s,s) - \mu_x(s)| \leq \sum_{d=1}^{\infty} \zeta^{\frac{d}{m+c_0}-1} \tag{154}$$

$$= \frac{1}{\zeta\left(1 - \zeta^{\frac{1}{m+c_0}}\right)} - \frac{1}{\zeta} \tag{155}$$

$$\leq \frac{1}{\zeta\left(1 - \zeta^{\frac{1}{l+c_0}}\right)} \tag{156}$$

$$= O\left(\frac{l + c_0}{\zeta \log \frac{1}{\zeta}} \right), \tag{157}$$

and

$$\sum_{d=1}^{\infty} d \left| p_x^{(d)}(s,s) - \mu_x(s) \right| \leq \sum_{d=1}^{\infty} d\zeta^{\frac{d}{m+c_0}-1} \tag{158}$$

$$= \frac{\zeta^{\frac{1}{m+c_0}}}{\zeta\left(1 - \zeta^{\frac{1}{m+c_0}}\right)^2} \tag{159}$$

$$\leq \frac{1}{\zeta\left(1 - \zeta^{\frac{1}{l+c_0}}\right)^2} \tag{160}$$

$$= O\left(\frac{(l + c_0)^2}{\zeta \log^2 \frac{1}{\zeta}} \right). \tag{161}$$

Combining (153), (157), and (161), we have

$$E\left[\left(\frac{1}{n} N_{\boldsymbol{x}}(s) - \mu_x(s) \right)^2 \right] = O\left(\frac{l}{n} + \frac{l^2}{n^2} \right). \tag{162}$$

From (152) and (162), we obtain (151) where $l = o(n)$. □

REFERENCES

[1] R. Begleiter, R. El-Yaniv, and G. Yona, "On prediction using variable order markov models," *J. Artificial Intell. Res.*, vol. 22, pp. 385–421, 2004.

[2] D. Benedetto, E. Caglioti, and V. Loreto, "Language trees and zipping," *Phys. Rev. Lett.*, vol. 88, no. 4, Jan. 28, 2002.

[3] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Digital Systems Research Center, Tech. Rep. 124, 1994.

[4] H. Cai, S. Kulkarni, and S. Verdú, "Universal estimation of entropy and divergence via block sorting," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun./Jul. 2002, p. 433.

[5] H. Cai, S. Kulkarni, and S. Verdú, "Universal entropy estimation via block sorting," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1551–1561, Jul. 2004.

[6] J. Doob, *Stochastic Processes*. New York: Wiley, 1953.

[7] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun. Theory*, vol. COM-32, no. 4, pp. 396–402, Apr. 1984.

[8] Z. Dawy, J. Hagenauer, and A. Hoffmann, "Implementing the context tree weighting method for context recognition," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2004, p. 536.

[9] J. Felsenstein, "Phylogeny inference package," *Cladistics*, vol. 5, pp. 164–166, 1989.
[10] R. M. Gray, *Entropy and Information Theory*. New York: Springer-Verlag, 1990.
[11] A. Kaltchenko, "Algorithms for estimation of information distance with application to bioinformatics and linguistics," in *Proc. 2004 Canadian Conf. Electrical and Computer Engineering*, Niagara Fall, ON, Canada, May 2004.
[12] M. Li, X. Chen, X. Li, B. Ma, and P. Vitanyi, "The similarity metric," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3250–3264, Dec. 2004.
[13] A. Martin, G. Seroussi, and M. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1442–1468, Jul. 2004.
[14] S. V. Nagaev, "More exact statements of limit theorems for homogeneous Markov chains," *Theory Probab. Appl.*, vol. 6, pp. 62–81, 1961.
[15] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.
[16] F. M. J. Willems, "The context tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.
[17] J. Ziv and N. Merhav, "A measure of relative entropy between individual sequences with application to universal classification," *IEEE Trans. Inf. Theory*, vol. 39, no. 4, pp. 1270–1279, Jul. 1993.