

# Universal Estimation of Erasure Entropy

Jiming Yu and Sergio Verdú, *Fellow, IEEE*

**Abstract**—Erasure entropy rate differs from Shannon’s entropy rate in that the conditioning occurs with respect to both the past and the future, as opposed to only the past (or the future). In this paper, consistent universal algorithms for estimating erasure entropy rate are proposed based on the basic and extended context-tree weighting (CTW) algorithms. Simulation results for those algorithms applied to Markov sources, tree sources, and English texts are compared to those obtained by fixed-order plug-in estimators with different orders.

**Index Terms**—Bidirectional context tree, context-tree weighting, data compression, entropy rate, universal algorithms, universal modeling.

## I. INTRODUCTION

THE erasure entropy for a finite collection of discrete random variables  $X_1^n$  is defined as [1]

$$H^-(X_1^n) \triangleq \sum_{i=1}^n H(X_i | X_1^{i-1}, X_{i+1}^n) \quad (1)$$

and the erasure entropy rate (or erasure entropy) of a process  $\mathbf{X} = (X_t)_{t \in \mathbb{Z}}$  is defined as

$$H^-(\mathbf{X}) \triangleq \limsup_{n \rightarrow \infty} \frac{1}{n} H^-(X_1^n). \quad (2)$$

For a stationary process  $\mathbf{X}$ , we have [1]

$$H^-(\mathbf{X}) = \lim_{n \rightarrow \infty} \frac{1}{n} H^-(X_1^n) = H(X_0 | X_{-\infty}^{-1}, X_1^\infty). \quad (3)$$

A useful quantity for our purposes is the  $\ell$ th-order erasure entropy for stationary processes defined by

$$H_\ell^-(\mathbf{X}) \triangleq H(X_0 | X_{-\ell}^{-1}, X_1^\ell). \quad (4)$$

Note that  $\lim_{\ell \rightarrow \infty} H_\ell^-(\mathbf{X}) = H^-(\mathbf{X})$  for stationary  $\mathbf{X}$ .

Like entropy rate, erasure entropy rate has various operational roles [1], [2], foremost among them is

- the information rate needed to supply to the observer of an erasure channel output in order to losslessly recover the channel input in the regime of sporadic erasures.

In this paper, we consider the universal estimation of the erasure entropy rate. Namely, based on a finite time realization  $x_1^n$  of a stationary process  $\mathbf{X} = (X_t)_{t \in \mathbb{Z}}$  taking values on a finite

alphabet  $\mathcal{A}$ , estimate the erasure entropy rate  $H^-(\mathbf{X})$  without any knowledge of its probability law. This paper proposes several universal estimation algorithms whose output converges to the erasure entropy if the input is a stationary ergodic processes.

The Shannon–MacMillan–Breiman theorem has a bidirectional counterpart shown in [2]. This result suggests that the problem of universal erasure entropy estimation can be reduced to one of universal bidirectional modeling. The context-tree weighting (CTW) based algorithms [3], [4] have been adapted to deal with noncausal/bidirectional conditioning structure in the problem of lossless compression with side information at both the encoder and decoder [5], [6]. In fact, estimating the limit in (2) can be viewed as a hypothetical problem in which the objective is to estimate the asymptotic expected ideal average lossless code length for  $X_1^n$  with side information at both the encoder and the decoder equal to  $X_{i+1}^n$  at each time  $i$  (of course at each time  $i$ , the past  $X_1^{i-1}$  is also known to both the encoder and the decoder).<sup>1</sup> Building upon this observation, we design erasure entropy rate estimators based on both the basic CTW [3] and the extended CTW [4] algorithms by finding a counterpart of the CTW trees to the bidirectional setting. Since the logarithm of bidirectional conditional probability

$$\log \frac{1}{P_{X_i | X_1^{i-1}, X_{i+1}^n}(X_i | X_1^{i-1}, X_{i+1}^n)} \quad (5)$$

still enjoys the code length interpretation, estimating erasure entropy based on CTW is as natural as its sequential data compression counterpart. Other approaches are to bidirectionally model the source via various other context trees [7]–[11], which are used in the universal discrete denoising setting [7] where a bidirectional model is needed. Among those, the BCT of [10], [11] is combined with CTW in this paper to obtain erasure entropy estimators based on asymmetric bidirectional extensions of CTW. This class of estimators uses the general context weighting of [12] in the bidirectional setting by specifying a node-dependent splitting rule.

This paper is organized as follows. Section II summarizes the proposed erasure entropy estimation algorithms based on the straightforward symmetric bidirectional extension of CTW (see also [13]). In Section III, the asymmetric bidirectional version of CTW is applied to erasure entropy estimation. Other related estimators are discussed in Section IV. Consistency of some of the proposed estimators for stationary ergodic processes is proved in Section V. Not only the analytical convergence results are important, but so is the performance of those algorithms in practical applications with finite data lengths. Implementation considerations and experimental results are presented in Sections VI and VII, respectively. A typical experimental result for

<sup>1</sup>Note that for compression this is an artificial problem since the future side information reveals the whole sequence at the beginning.

Manuscript received November 23, 2006; revised August 03, 2008. Current version published December 24, 2008. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Seattle, WA, July 2006.

The authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544, USA (e-mail: jiminyu@princeton.edu; verdu@princeton.edu).

Communicated by Y. Steinberg, Associate Editor for Shannon Theory.

Digital Object Identifier 10.1109/TIT.2008.2008117

English text yields values of erasure entropy and Shannon entropy equal to 0.22 bit per letter and 1.3 bits per letter, respectively.

## II. SYMMETRIC BIDIRECTIONAL CTW

The CTW algorithms [3], [4] have been straightforwardly extended to the case of symmetric bidirectional contexts (by considering the extended alphabet of two directions) based on the unidirectional models for entropy estimation [5], [6], [14]–[16], and are used to estimate

$$\prod_{i=1}^n P_{X_i|X_1^{i-1}, X_{i+1}^n} (x_i|x_1^{i-1}, x_{i+1}^n) \quad (6)$$

for a realization  $x_1^n$  of a process  $\mathbf{X}$ , which is in turn used to estimate the erasure entropy of  $\mathbf{X}$  via the Shannon–MacMillan–Breiman theorem for erasure entropy. The erasure entropy estimator corresponding to this direct extension is called one-pass symmetric basic or extended CTW estimator depending on whether basic CTW is used or extended CTW is used. In addition, by adopting the two-pass approach of [15], [16], two-pass symmetric basic and extended CTW estimators are obtained.

Because of the similarity between those estimators and their one-pass and two-pass unidirectional counterparts in [5], [6], [14]–[16], we refer the reader to those references and to [13], [17] for a full description of this class of algorithms: one-pass/two-pass symmetric basic CTW estimator, one-pass/two-pass symmetric extended CTW estimator.

## III. CONTEXT-TREE WEIGHTING ON BIDIRECTIONAL CONTEXT TREES: ASYMMETRIC BIDIRECTIONAL CTW

The natural extension of CTW to the case of *symmetric* bidirectional contexts is considered in Section II. When we consider bidirectional contexts we can exploit the capabilities of *asymmetric* bidirectional contexts, namely, contexts with different lengths on both sides. Previous work includes [8], [9], and [10], [11], where [8] and [9] investigate bidirectional (or multidirectional) modeling by minimizing a given loss function via a tree structure, and [10] and [11] consider probabilistic bidirectional modeling of a stationary source via another tree structure. Based on the tree structure of the bidirectional context tree (BCT) [10], [11] (cf. Fig. 2 of [11] for an example of the BCT tree), we develop our asymmetric version of CTW, though the estimators based on this class of algorithms are heuristic in nature (unless some technical issue can be resolved, cf. Section V-C). This gives an algorithm BCT-CTW that uses [12] allowing node-dependent splits in the bidirectional setting. By specifying a splitting rule that varies according to whether the bidirectional context is symmetric or not, the bidirectional context models in BCT-CTW are generated by a sequence of context splits and are thus mixed via a direct application of the algorithm of [12], where the mixing formula depends on the specification of the splits. In Section III-A, we include the (one-pass version of the) algorithm description for completeness.

### A. One-Pass Basic BCT-CTW Estimation Algorithm

We use notations and conventions similar to those in [3]. Specifically,  $N_{(u,v)}^i(a)$  represents the number of occurrences of symbol  $a$  in the bidirectional context  $(u, v)$  after updating the tree according to  $x_{1-\ell}^{i+\ell}$ ,  $P_e^{(u,v)}(i)$  represents estimated probability in node  $(u, v)$  (namely, the Krichevsky–Trofimov (KT) estimate [18] for the bidirectional context  $(u, v)$ ) after updating the tree according to  $x_{1-\ell}^{i+\ell}$ ,  $P_w^{(u,v)}(i)$  represents estimated probability in the bidirectional context  $(u, v)$  after updating the tree according to  $x_{1-\ell}^{i+\ell}$ . We use  $\mathcal{A}$  to denote the finite alphabet of the symbols,  $\lambda$  to represent the empty string, and we identify a node with its corresponding bidirectional context.

1) *Step 1*: Set  $i = 1$ .

2) *Step 2*: For all nodes  $(u, v)$  belonging to the set  $\{(x_{i-l}^{i-1}, x_{i+r}^{i+r}) : 0 \leq l \leq \ell, 0 \leq r \leq \ell\}$ , update the counts, estimated and weighted probabilities as follows.

- If  $|u| = |v|$ :

$$P_e^{(u,v)}(i) = \frac{N_{(u,v)}^{i-1}(x_i) + \frac{1}{2}}{\sum_{c \in \mathcal{A}} N_{(u,v)}^{i-1}(c) + \frac{1}{2}|\mathcal{A}|} P_e^{(u,v)}(i-1) \quad (7)$$

$$N_{(u,v)}^i(x_i) = N_{(u,v)}^{i-1}(x_i) + 1 \quad (8)$$

$$N_{(u,v)}^i(a) = N_{(u,v)}^{i-1}(a), \quad \forall a \neq x_i \quad (9)$$

$$P_w^{(u,v)}(i) = \begin{cases} \frac{1}{4} \left[ P_e^{(u,v)}(i) + \prod_{a \in \mathcal{A}} P_w^{(au,v)}(i) \right. \\ \quad \left. + \prod_{a,b \in \mathcal{A}} P_w^{(au,vb)}(i) + \prod_{b \in \mathcal{A}} P_w^{(u,vb)}(i) \right] \\ \quad \text{if } 0 \leq l(s) < \ell, \\ P_e^{(u,v)}(i) \quad \text{if } l(s) = \ell. \end{cases} \quad (10)$$

- If  $|u| < |v|$ :

$$P_e^{(u,v)}(i) = \frac{N_{(u,v)}^{i-1}(x_i) + \frac{1}{2}}{\sum_{c \in \mathcal{A}} N_{(u,v)}^{i-1}(c) + \frac{1}{2}|\mathcal{A}|} P_e^{(u,v)}(i-1) \quad (11)$$

$$N_{(u,v)}^i(x_i) = N_{(u,v)}^{i-1}(x_i) + 1 \quad (12)$$

$$N_{(u,v)}^i(a) = N_{(u,v)}^{i-1}(a), \quad \forall a \neq x_i \quad (13)$$

$$P_w^{(u,v)}(i) = \begin{cases} \frac{1}{2} \left[ P_e^{(u,v)}(i) + \prod_{b \in \mathcal{A}} P_w^{(u,vb)}(i) \right], \\ \quad \text{if } 0 \leq l(s) < \ell \\ P_e^{(u,v)}(i), \quad \text{if } l(s) = \ell. \end{cases} \quad (14)$$

- If  $|u| > |v|$ :

$$P_e^{(u,v)}(i) = \frac{N_{(u,v)}^{i-1}(x_i) + \frac{1}{2}}{\sum_{c \in \mathcal{A}} N_{(u,v)}^{i-1}(c) + \frac{1}{2}|\mathcal{A}|} P_e^{(u,v)}(i-1) \quad (15)$$

$$N_{(u,v)}^i(x_i) = N_{(u,v)}^{i-1}(x_i) + 1 \quad (16)$$

$$N_{(u,v)}^i(a) = N_{(u,v)}^{i-1}(a), \quad \forall a \neq x_i \quad (17)$$

$$P_w^{(u,v)}(i) = \begin{cases} \frac{1}{2} \left[ P_e^{(u,v)}(i) + \prod_{a \in \mathcal{A}} P_w^{(au,v)}(i) \right], \\ \quad \text{if } 0 \leq l(s) < \ell \\ P_e^{(u,v)}(i), \quad \text{if } l(s) = \ell. \end{cases} \quad (18)$$

All the paths starting from any node in the set  $\{(x_{i-l}^{i-1}, x_{i+r}^{i+r}) : 0 \leq l \leq \ell, 0 \leq r \leq \ell\}$  back to the root node  $(\lambda, \lambda)$  are called *updating paths* for the symbol  $x_i$  at time  $i$ .

*Remark 1:* The tree structure of BCT serves just as a convenient way to organize nodes and compute the mixtures in (10), (14), and (18).

Based on the tree structure of BCT, the implementation of *Step 2* is recursive: assuming the current node  $(u_0, v_0)$  at depth  $d_0 = \max\{|u_0|, |v_0|\}$  has been processed, look for its child nodes which fit into the current bidirectional data, namely, the child nodes of the form

- $(x_{i-d_0-1}u_0, v_0), (x_{i-d_0-1}u_0, v_0x_{i+d_0+1}),$  and  $(u_0, v_0x_{i+d_0+1}),$  if  $|u_0| = |v_0|;$
- $(x_{i-d_0-1}u_0, v_0),$  if  $|u_0| > |v_0|;$
- $(u_0, v_0x_{i+d_0+1}),$  if  $|u_0| < |v_0|.$

Update those fitting child nodes, and then the recursion starts at those fitting child nodes.

Notice by the above updating rules, a node should be updated *after* all its child nodes have been updated. This can be implemented by updating all nodes along the way from any visited leaf node at depth  $\ell$  back to the root node.

3) *Step 3:* Set  $i \leftarrow i + 1,$  if  $i \leq n,$  go to *Step 2,* otherwise stop.

The one-pass basic BCT-CTW estimator for the erasure entropy is

$$\frac{1}{n} \hat{H}_{\text{BCT},1,\ell,n}^- (x_{1-\ell}^{n+\ell}) \triangleq \frac{1}{n} \log \frac{1}{P_w^{(\lambda,\lambda)}(n)}. \quad (19)$$

*Remark 2:* Comparing with the general context weighting in [12], BCT-CTW simultaneously allows three types of splits ( $|\mathcal{A}|$ -ary as well as  $|\mathcal{A}| \times |\mathcal{A}|$ -ary) in symmetric nodes  $(u, v)$  with  $|u| = |v|$  and mixes those three splits together in the way specified by (10), due to the bidirectional setting. For asymmetric nodes  $(u, v)$  with  $|u| \neq |v|,$  BCT-CTW allows just one type of split as in the original CTW [3], [4] and the general context weighting [12] (cf. (14) and (18)). Overall, BCT-CTW allows node-dependent splits with three types of splits mixed for symmetric nodes, and can be viewed as a special case of the general context weighting in [12] with this modification in the bidirectional setting.

Since BCT-CTW is just imposing the weighting rules (10), (14), and (18) to the tree structure of BCT, child nodes from all three types of splits are present concurrently in the symmetric nodes  $(u, v)$  with  $|u| = |v|$  according to the definition of BCT tree structure (cf. [11]). When it comes to weighting, all the child nodes of any symmetric node are grouped into three subtrees corresponding to the three splits and are mixed together by (10). Although none of the three types of splits alone covers all child nodes of any symmetric node (i.e., not all children of a symmetric node are included in (10) as a single term out of four terms), the three types of splits together cover all the child nodes of any symmetric node and they are mixed in the way specified by (10) as different terms.

*Remark 3:* There are several differences in the way BCT-CTW and the algorithm in [8], [9] estimate bidirectional conditional distributions:

- (a) they define different splitting rules to apply the general context weighting algorithm of [12]. While [8], [9]

splits according to context increases in one direction, BCT-CTW splits according to context increases in either or both directions.

- (b) References [8], [9] generate an optimal, disjoint, and exhaustive bidirectional context set (not necessarily unique) for estimation according to a given loss function (e.g., the code length optimization criterion). In contrast, BCT-CTW mixes all bidirectional contexts  $\{(u, v) \in \mathcal{A}^* \times \mathcal{A}^* : \max\{|u|, |v|\} \leq \ell\}$  up to depth  $\ell$  (here  $\mathcal{A}^*$  is the set of all finite-length strings on  $\mathcal{A}$ ) according to its splitting rule in a specific way (cf. *Step 2* of the BCT-CTW algorithm). BCT-CTW does not solve any optimization problem explicitly and does not generate any bidirectional context set.

## B. Two-Pass Basic BCT-CTW Estimation Algorithm

The two-pass basic BCT-CTW estimator is the direct extension of the two-pass approach of [15], [16] to our basic BCT-CTW setting and thus is omitted here, cf. [17] for details.

## IV. OTHER ESTIMATORS

### A. One-Pass and Two-Pass Extended BCT-CTW Estimators

As in Sections III-A and III-B, the corresponding one-pass and two-pass algorithms combining BCT and extended CTW (extended BCT-CTW) can be developed. Generally these algorithms consume too much space to be practical, unless a maximal tree depth is imposed. This way, the extended BCT-CTW algorithms have very similar behavior to their basic versions in Section III and thus are omitted in this paper.

### B. Context-Tree Maximizing Based Estimators

Context-tree maximizing (CTM) [19] is very similar to CTW. We have also considered CTM-based erasure entropy estimators; the experimental results and complexity are very similar to those of the CTW estimators.

### C. Fixed-Order Plug-In Estimator

The empirical fixed-order plug-in estimator approximates  $H_m^-(\mathbf{X})$  substituting in its definition (4) the conditional probabilities by the empirical estimates. As in [20], this results in underestimation of the  $m$ th-order erasure entropy with a  $O(\frac{1}{n})$  bias. In general, if  $m$  underestimates the process memory length the performance is poor, while if it is sufficiently high, the empirical estimates are usually quite noisy unless  $n$  is very high.

## V. CONSISTENCY RESULTS

We refer the reader to [17] for all the proofs due to space limitations.

### A. Consistency Result for Extended CTW Based Algorithms

*Theorem 1:* Let  $\frac{1}{n} \hat{H}_{1,n}^- : \mathcal{A}^n \rightarrow \mathbb{R}_+$  be the one-pass symmetric extended CTW estimator and suppose  $\mathbf{X}$  is stationary ergodic, then

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \hat{H}_{1,n}^- (X_1^n) \leq H^-(\mathbf{X}) \quad \text{a.s.} \quad (20)$$

### B. Consistency Results for One-Pass and Two-Pass Symmetric Basic CTW Algorithms

*Lemma 1:* Let  $\mathbf{X}$  be stationary ergodic. Let  $s = (a_{-m}^{-1}, a_1^m)$  with  $m < \ell$  be a node in the symmetric bidirectional basic CTW tree (cf. [3] for the original unidirectional basic CTW tree, or [17] for full details) such that there exists  $s' = (a_{-\ell}^{-1}, a_1^\ell)$  with  $P_{X_{-\ell}^{-1}, X_1^\ell}(a_{-\ell}^{-1}, a_1^\ell) > 0$  and a different conditional probability distribution, i.e.,  $\exists a_0 \in \mathcal{A}, a_{-\ell}^{-m-1}, a_{m+1}^\ell \in \mathcal{A}^{\ell-m}$  such that

$$P_{X_0|X_{-m}^{-1}, X_1^m}(a_0|a_{-m}^{-1}, a_1^m) \neq P_{X_0|X_{-\ell}^{-1}, X_1^\ell}(a_0|a_{-\ell}^{-1}, a_1^\ell).$$

Let

$$\beta_i^s \triangleq \frac{P_e^s(i)}{\prod_{s' \in C(s)} P_w^{s'}(i)} \quad (21)$$

where  $C(s)$  is the set of child nodes of  $s$ , then

$$\lim_{i \rightarrow \infty} \beta_i^s = 0 \quad \text{a.s.} \quad (22)$$

*Theorem 2:* Let  $\frac{1}{n} \hat{H}_{1,\ell,n}^- : \mathcal{A}^{n+2\ell} \rightarrow \mathbb{R}_+$  be the one-pass symmetric basic CTW estimator and suppose  $\mathbf{X}$  is stationary ergodic, then

$$\lim_{n \rightarrow \infty} \frac{1}{n} \hat{H}_{1,\ell,n}^- (X_{1-\ell}^{n+\ell}) = H_\ell^-(\mathbf{X}) \quad \text{a.s.} \quad (23)$$

*Corollary 1:* If  $\mathbf{X}$  is stationary ergodic, then the one-pass symmetric basic CTW estimator  $\frac{1}{n} \hat{H}_{1,\ell,n}^- (X_{1-\ell}^{n+\ell})$  satisfies

$$\lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E} \hat{H}_{1,\ell,n}^- (X_{1-\ell}^{n+\ell}) = H_\ell^-(\mathbf{X}).$$

*Theorem 3:* Let  $\frac{1}{n} \hat{H}_{2,\ell,n}^- : \mathcal{A}^{n+2d} \rightarrow \mathbb{R}_+$  be the two-pass symmetric basic CTW estimator and suppose  $\mathbf{X}$  is stationary ergodic, then

$$\lim_{n \rightarrow \infty} \frac{1}{n} \hat{H}_{2,\ell,n}^- (X_{1-\ell}^{n+\ell}) = H_\ell^-(\mathbf{X}) \quad \text{a.s.} \quad (24)$$

*Corollary 2:* Let  $\mathbf{X}$  be stationary ergodic, then the two-pass symmetric basic CTW estimator  $\frac{1}{n} \hat{H}_{2,\ell,n}^- : \mathcal{A}^{n+2\ell} \rightarrow \mathbb{R}_+$  satisfies

$$\lim_{n \rightarrow \infty} \frac{1}{n} \mathbb{E} \hat{H}_{2,\ell,n}^- (X_{1-\ell}^{n+\ell}) = H_\ell^-(\mathbf{X}). \quad (25)$$

### C. Discussion on the Consistency of Basic BCT-CTW Estimators and Their Modifications

It is desirable to prove a counterpart of Lemma 1 for the basic BCT-CTW estimator (Section III), but we can only prove some partial results (cf. [17]), leaving two problematic cases with nodes  $(u, v)$  of lengths  $l = |u|, r = |v|$  satisfying  $0 \leq l < r = \ell$  and  $0 \leq r < l = \ell$  unproven. The problematic cases are those when we have reached a leaf node  $(a_{-l}^{-1}, a_1^r)$ , it does not guarantee a good estimate of the bidirectional conditional distribution  $P_{X_0|X_{-\ell}^{-1}, X_1^\ell}(\cdot | c_{-\ell}^{-l-1} a_{-l}^{-1}, a_1^r c_{r+1}^\ell)$  for all  $c_{-\ell}^{-l-1}, c_{r+1}^\ell$ , whereas in the (either one-pass or two-pass) symmetric basic CTW case, a leaf node always gives a good estimate of this distribution from the ergodic property. If the problematic cases were proved to satisfy the counterpart of

Lemma 1, we could show the strong consistency of our one-pass and two-pass basic BCT-CTW estimators as we showed in Theorems 2 and 3, Corollaries 1 and 2 for the one-pass and two-pass basic CTW estimators.

One way to circumvent this challenge is to modify the one-pass and two-pass basic BCT-CTW algorithms in the following way. Let  $0 < \ell_1 < \ell_2$  be integers. We first construct the symmetric bidirectional basic CTW tree (cf. Section II, or [3] for the original unidirectional basic CTW tree, or [17] for full details) with maximum depth  $\ell_1$ ; then starting from nodes of depth  $\ell_1$  we construct the basic BCT-CTW tree as discussed in Section III-A. This at least gives good estimates for bidirectional conditional distributions  $P_{X_0|X_{-\ell}^{-1}, X_1^\ell}$  with  $0 \leq l, r \leq \ell_1$  in all cases (including the above mentioned problematic cases  $0 \leq l < r = \ell$  and  $0 \leq r < l = \ell$  for the pure basic BCT-CTW approach). Nevertheless, this is a direct combination of the algorithms developed in Sections II and III, and similar consistency results are readily obtained; thus, we do not pursue this direction further.

## VI. IMPLEMENTATION CONSIDERATIONS

### A. Complexity Issues

For the one-pass and two-pass symmetric basic CTW algorithms, and the one-pass and two-pass BCT-CTW algorithms, both the time and space complexity are linear in the data length  $n$ , because  $\ell$  is a fixed constant.

For the one-pass and two-pass symmetric extended CTW algorithms, the algorithm description itself is conceptually simpler than the full-power counterpart of the extended CTW [4] though they give exactly the same estimates for a given realization. Based on the compact tree structure discussed there (but of course with slightly more complicated implementation), a symmetric bidirectional counterpart with linear space complexity in data length  $n$  can be obtained.

### B. General Finite-Alphabet Case

If the alphabet size  $|\mathcal{A}|$  is large, CTW generally does not achieve good performance in estimation. To overcome this problem, a binary symbol decomposition method is usually used [21], [22], namely, each symbol in the alphabet  $\mathcal{A}$  is decomposed into  $\lceil \log_2 |\mathcal{A}| \rceil$  bits. Each bit in a symbol has its own symmetric CTW or BCT-CTW tree, where the symmetric CTW or BCT-CTW tree is constructed by *only* considering complete symbol contexts, not bit contexts. For example, in the sequence  $s_1 z s_2$  where  $s_1, s_2$  are strings on  $\mathcal{A}$  with equal lengths, let  $z \in \mathcal{A}$  be decomposed into  $M \triangleq \lceil \log_2 |\mathcal{A}| \rceil$  bits  $z_1^M$ , then each bit  $z_i$  has exactly the same set of bidirectional contexts, e.g.,  $(s_1, s_2)$  is one of their common bidirectional contexts.

The root node of the symmetric CTW or BCT-CTW tree for the  $j$ th bit in one symbol of  $\mathcal{A}$  has  $2^{j-1}$  child nodes,  $j = 1, 2, \dots, M$ , corresponding to each of the previous  $(j-1)$  bits. The subtrees rooted at those first-level child nodes are constructed as described in Section II, by considering only complete symbol contexts. Each internal node in those subtrees has exactly  $|\mathcal{A}|^2$  child nodes. The updating procedure for estimated

and weighted probabilities will then be done until the root node for the symmetric CTW or BCT-CTW tree of each bit is updated. Take the one-pass symmetric basic and extended CTW algorithms for example, if we let  $x_i$  be decomposed into bits  $(x_{i,j})_{j=1}^M$ , the weighted probability at the root node of the context tree for the  $j$ th bit is an estimate of either the following product:

$$\prod_{i=\ell+1}^{n-\ell} P_{X_{i,j}|X_{i-\ell}^{i-1}, X_{i+1}^{i+\ell}, (X_{i,k})_{k=1}^{j-1}} \left( x_{i,j} | x_{i-\ell}^{i-1}, x_{i+1}^{i+\ell}, (x_{i,k})_{k=1}^{j-1} \right)$$

for the one-pass symmetric basic CTW algorithm or

$$\prod_{i=1}^n P_{X_{i,j}|X_1^{i-1}, X_{i+1}^n, (X_{i,k})_{k=1}^{j-1}} \left( x_{i,j} | x_1^{i-1}, x_{i+1}^n, (x_{i,k})_{k=1}^{j-1} \right)$$

for the one-pass symmetric extended CTW algorithm. Multiplying those  $j$ th bit weighted probabilities together for  $j = 1, 2, \dots, M$ , we get either an estimate for

$$\begin{aligned} & \prod_{j=1}^M \prod_{i=\ell+1}^{n-\ell} P_{X_{i,j}|X_{i-\ell}^{i-1}, X_{i+1}^{i+\ell}, (X_{i,k})_{k=1}^{j-1}} \left( x_{i,j} | x_{i-\ell}^{i-1}, x_{i+1}^{i+\ell}, (x_{i,k})_{k=1}^{j-1} \right) \\ &= \prod_{i=\ell+1}^{n-\ell} P_{X_i|X_{i-\ell}^{i-1}, X_{i+1}^n} \left( x_i | x_{i-\ell}^{i-1}, x_{i+1}^n \right) \quad (26) \end{aligned}$$

for the one-pass symmetric basic CTW algorithm or

$$\begin{aligned} & \prod_{j=1}^M \prod_{i=1}^n P_{X_{i,j}|X_1^{i-1}, X_{i+1}^n, (X_{i,k})_{k=1}^{j-1}} \left( x_{i,j} | x_1^{i-1}, x_{i+1}^n, (x_{i,k})_{k=1}^{j-1} \right) \\ &= \prod_{i=1}^n P_{X_i|X_1^{i-1}, X_{i+1}^n} \left( x_i | x_1^{i-1}, x_{i+1}^n \right) \quad (27) \end{aligned}$$

for the one-pass symmetric extended CTW algorithm. The one-pass symmetric basic and extended CTW estimators are obtained by dividing the logarithms of the reciprocals of the above products by  $n$ , respectively. Other algorithms have similar interpretations under bit decomposition, but they are then interpreted term-by-term as in the sums.

### C. Heuristic for Choosing Memory Length $\ell$ in the Basic CTW Based Algorithms

The basic CTW based algorithms use a heuristic that chooses its memory length parameter  $\ell$ : let  $\frac{1}{n} \hat{H}_{k,\ell,n}^-$  represent the estimate by the symmetric basic CTW-based algorithms with parameter  $\ell$  for data with length  $n$ , and  $k = 1$  for the one-pass algorithm,  $k = 2$  for the two-pass algorithm (the notations are the same as before for those two estimators), then the heuristic is to find the smallest  $\ell$  such that  $\frac{1}{n} \hat{H}_{k,\ell-1,n}^- \leq \frac{1}{n} \hat{H}_{k,\ell,n}^- \leq \frac{1}{n} \hat{H}_{k,\ell+1,n}^-$  for the same data with length  $n$ ; similarly for the one-pass or two-pass basic BCT-CTW estimators  $\frac{1}{n} \hat{H}_{\text{BCT},k,\ell,n}^-$ ,  $k = 1, 2$ . From the simulations for Markov sources and tree sources in Section VII, we can see that this heuristic usually leads to the correct choice of  $\ell$  (Markov order) for the one-pass symmetric basic CTW or one-pass basic BCT-CTW estimator unless  $n$  is small. For the two-pass symmetric basic CTW or two-pass basic BCT-CTW estimator, it tends to overestimate the order in the short run, but it still converges to the right estimate.

## VII. EXPERIMENTS

We test our algorithms and fixed-order estimators with synthetic (Markov and tree) sources as well as English texts:

- one-pass symmetric basic CTW (Section II);
- two-pass symmetric basic CTW (Section II);
- one-pass symmetric extended CTW (Section II);
- two-pass symmetric extended CTW (Section II);
- one-pass basic BCT-CTW (Section III-A);
- two-pass basic BCT-CTW (Section III-B);
- fixed-order estimators (for comparison purpose, cf. Section IV-C).

Each point in the figures for synthetic sources is an average of 100 runs of the estimators for 100 independent realizations of the synthetic sources. The unit of the estimates in all figures is bits per symbol.

### A. Markov Source

Consider a second-order Markov source on the alphabet  $\mathcal{A} = \{0, 1, 2, 3\}$  with strictly positive transition probabilities  $(P_{s,a})_{s \in \mathcal{S}, a \in \mathcal{A}}$  (cf. [17]), where  $\mathcal{S} \triangleq \mathcal{A} \times \mathcal{A}$  is the state space. The exact erasure entropy rate of this Markov source is (i.e., computed by using their *known* statistics) 1.20897 bits per symbol, whereas the entropy rate is 1.66365 bits per symbol [17].

Fig. 1 displays the estimated erasure entropy versus data length for all our estimators and several fixed-order estimators. The straight horizontal line is the true erasure entropy. In general, the one-pass algorithms perform similarly and the curves for two-pass algorithms almost coincide, respectively, with the two-pass approaches exhibiting superior performance than their one-pass counterparts. Despite the increase in complexity, the basic BCT-CTW estimators do not exhibit much better performance than the symmetric basic/extended CTW estimators, because the Markov source here is very symmetric thus considering possibly asymmetric structure does not give us much gain. It is not surprising that when we use the small order 2 for the fixed-order estimator, performance is better than any other algorithm at any length. This is because CTW mixes *all* submodels, thus it is *strictly* worse than the best model, and the fixed-order estimator works so well thanks to the *a priori* information about the exact order of the underlying process and the sufficient amount of data compared to the small order 2 of the source. Note from Fig. 1 that the fixed-order estimators suffer from severe performance degradation when the model order is either underestimated or overestimated.

### B. Tree Source

A binary tree source as used in [16] with maximal memory length  $m = 11$  with a suffix set  $\mathcal{S}$  of size  $|\mathcal{S}| = 15$  is constructed to test our algorithms, where

$$\begin{aligned} \mathcal{S} = \{ & 00000000000, 10000000000, 1000000000, \\ & 100000000, 10000000, 1000000, 100000, \\ & 10000, 1000, 100, 01, 010, 110, 011, 111 \} \quad (28) \end{aligned}$$

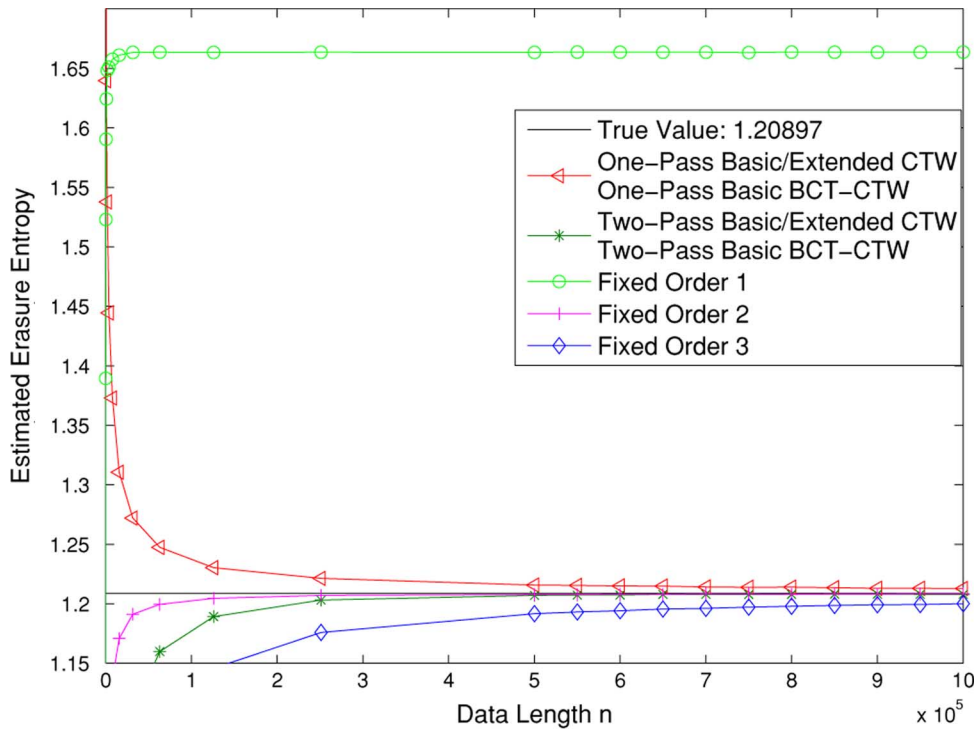


Fig. 1. Second-order Markov source with alphabet size 4.

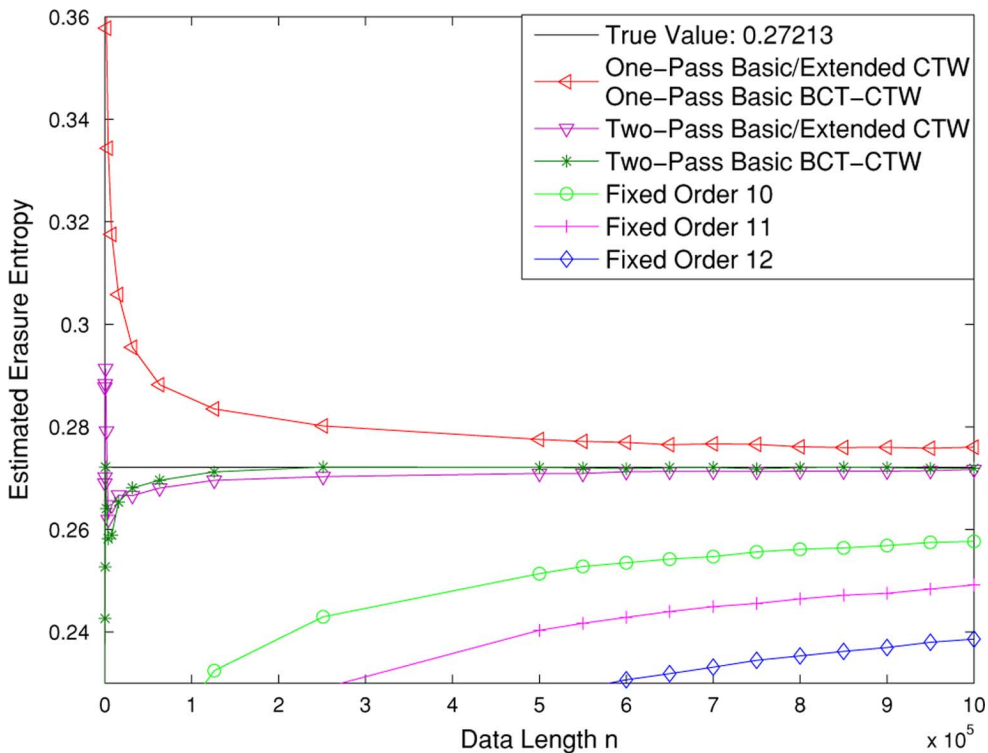


Fig. 2. Binary tree source: memory length 11, suffix set size 15.

and each of those contexts has conditional probabilities of the next symbol being 0 given by

$$\{0.97, 0.86, 0.50, 0.96, 0.78, 0.60, 0.68, 0.76, 0.84, 0.92, 0.99, 0.20, 0.36, 0.30, 0.40\}.$$

The exact erasure entropy of this tree source is (i.e., computed by using their *known* statistics) 0.27213 bits per symbol, whereas the entropy rate is 0.446894 bits per symbol [17].

Fig. 2 shows the estimated erasure entropy versus data length for our six estimators, and for fixed-order estimators of different orders. A similar trend as in the Markov source case holds here, too: one-pass approaches are close to each other and two-pass estimators have nearly the same performance, respectively, with the two-pass approaches being better than their one-pass counterparts. But here the asymmetric basic BCT-CTW estimators have slightly superior performance than the symmetric basic/

TABLE I  
ERASURE ENTROPY AND ENTROPY FOR ENGLISH TEXTS

Novels	Symmetric Basic CTW ( $\ell$ )	Symmetric Extended CTW	Basic BCT-CTW ( $\ell$ )	Entropy
Don Quixote	0.33 (7)	0.33	0.27 (4)	1.39
David Copperfield	0.33 (6)	0.33	0.26 (4)	1.40
Great Expectations	0.40 (6)	0.40	0.31 (4)	1.46
Household Tales by Brothers Grimm	0.30 (7)	0.30	0.26 (4)	1.26
Jane Eyre	0.47 (5)	0.47	0.34 (4)	1.57
Les Miserables	0.34 (7)	0.34	0.27 (4)	1.39
Little Dorrit	0.34 (6)	0.34	0.26 (4)	1.42
Micah Clarke	0.48 (5)	0.48	0.37 (4)	1.54
Notre-Dame de Paris	0.45 (5)	0.45	0.33 (4)	1.44
The Arabian Nights	0.30 (6)	0.30	0.24 (4)	1.29
The Count of Monte Cristo	0.31 (7)	0.31	0.25 (4)	1.36
The Great Boer War	0.34 (5)	0.34	0.24 (4)	1.30
The Life and Adventures of Nicholas Nickleby	0.35 (6)	0.35	0.27 (4)	1.40
The Moonstone	0.34 (6)	0.34	0.22 (5)	1.33
The White Company	0.48 (5)	0.48	0.33 (5)	1.50

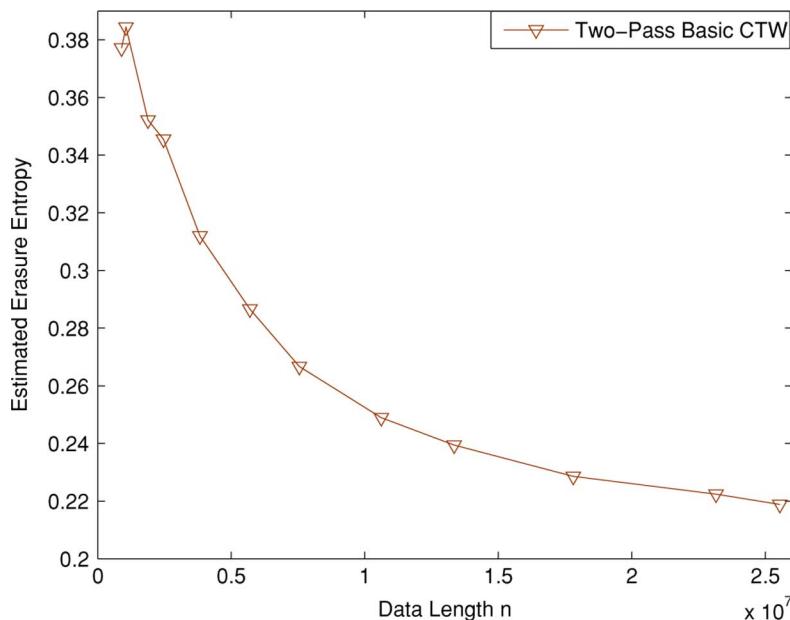


Fig. 3. Texts by Charles Dickens: erasure entropy (bits per letter).

extended CTW estimators. This is in contrast to the Markov case: for the “sparse” tree source in this subsection, exploiting asymmetric structure is obviously more economical and efficient, thus more reliable statistics can be obtained. And those fixed-order estimators suffer considerable degradation from insufficient data even with the correct *a priori* information about the order 11 of the tree source, the curves are still far away from the straight line representing the true erasure entropy. This illustrates the main usefulness of CTW based estimators over fixed-order estimators.

### C. English Texts

We experiment with 15 novels in English or their English translations (as in [10]) of various lengths, ranging from about

$9 \times 10^5$  to  $3 \times 10^6$  characters, as shown in Table I. The simulation is conducted with the two-pass symmetric extended CTW algorithm, the two-pass symmetric basic CTW algorithm with  $\ell = 1, 2, 3, 4, 5, 6, 7, 8$ , and the two-pass basic BCT-CTW algorithm with  $\ell = 1, 2, 3, 4$ .<sup>2</sup> The column labeled “Symmetric Basic CTW ( $\ell$ )” corresponds to results of the two-pass symmetric basic CTW algorithm with the heuristic explained in Section VI-C. The numbers in parentheses are the memory lengths chosen by the heuristic. The “Symmetric Extended CTW” column represents the estimates by the two-pass symmetric extended CTW algorithm. The column labeled “Basic

<sup>2</sup>Possibly with  $\ell = 5$  when some relatively short text is in use. The reason why  $\ell > 5$  is not used for the two-pass basic BCT-CTW estimator is due to limited computational capabilities, namely, too much space is needed for large  $\ell$ .

BCT-CTW ( $\ell$ )” displays the estimates from the two-pass basic BCT-CTW estimator, with the numbers in parentheses being the memory length chosen by the heuristic. The “Entropy” column displays the estimated entropy rates, by using a similar two-pass extended CTW based entropy estimator.

In fact, because English texts have relatively large alphabet size, and finite-order Markov approximation fully captures their dependence structure [23], the convergence of our estimators is slower than in the case of synthetic sources. This can be seen from Fig. 3, where we plot the estimates from the two-pass symmetric basic CTW estimator with memory length  $\ell = 4$  versus the data lengths for a concatenation of 12 of Charles Dickens’ writings. In the interval  $(9 \times 10^5, 3 \times 10^6)$  of lengths of English texts in Table I, the estimated erasure entropy by two-pass symmetric basic/extended CTW estimators is about 0.35 bits per letter and is comparable to the two-pass basic CTW based estimator at the same lengths in Fig. 3. As data length increases to about  $2.55 \times 10^7$  (by concatenating several pieces of Charles Dickens’ writings), the estimated erasure entropy decreases to about 0.22 bits per letter. This latter estimate is close to the values obtained in Table I by the two-pass basic BCT-CTW estimators, which characterize the dependence structure in a more efficient, parsimonious, and asymmetric way compared to two-pass symmetric basic/extended CTW estimators.

#### ACKNOWLEDGMENT

The referee’s conscientious comments and suggestions are gratefully acknowledged.

#### REFERENCES

- [1] S. Verdú and T. Weissman, “Erasure entropy,” in *Proc. 2006 IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2006, pp. 98–102.
- [2] S. Verdú and T. Weissman, “The information lost in erasures,” *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 5030–5058, Nov. 2008.
- [3] 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.
- [4] F. M. J. Willems, “The context-tree weighting method: Extensions,” *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.
- [5] H. Cai, S. R. Kulkarni, and S. Verdú, “A universal lossless compressor with side information based on context tree weighting,” in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 2340–2344.
- [6] H. Cai, S. R. Kulkarni, and S. Verdú, “An algorithm for universal lossless compression with side information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 9, pp. 4008–4016, Sep. 2006.
- [7] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, “Universal discrete denoising: Known channel,” *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 5–28, Jan. 2005.
- [8] E. Ordentlich, M. J. Weinberger, and T. Weissman, “Efficient pruning of bi-directional context trees with applications to universal denoising and compression,” in *Proc. IEEE Information Theory Workshop*, San Antonio, TX, Oct. 2004.
- [9] E. Ordentlich, M. J. Weinberger, and T. Weissman, “Multi-directional context sets with applications to universal denoising and compression,” in *Proc. IEEE Int. Symp. Information Theory*, Adelaide, Australia, Sep. 2005, pp. 1270–1274.

- [10] J. Yu and S. Verdú, “Schemes for bi-directional modeling of discrete stationary sources,” in *Proc. 39th Annual Conf. Information Science and Systems*, Baltimore, MD, Mar. 2005.
- [11] J. Yu and S. Verdú, “Schemes for bidirectional modeling of discrete stationary sources,” *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4789–4807, Nov. 2006.
- [12] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “Context weighting for general finite-context sources,” *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1514–1520, Sep. 1996.
- [13] J. Yu and S. Verdú, “Universal erasure entropy estimation,” in *Proc. IEEE Int. Symp. Information Theory*, Seattle, WA, Jul. 2006, pp. 2358–2362.
- [14] H. Cai, S. R. Kulkarni, and S. Verdú, “Universal entropy estimation via block sorting,” *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1551–1561, Jul. 2004.
- [15] H. Cai, “Universal Estimation of Information Measures for Finite Alphabet Sources,” Ph.D. dissertation, Princeton Univ., Princeton, NJ, 2005.
- [16] H. Cai, S. R. Kulkarni, and S. Verdú, “Universal divergence estimation for finite-alphabet sources,” *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3456–3475, Aug. 2006.
- [17] J. Yu, “Bidirectional Modeling of Finite-Alphabet Sources,” Ph.D. dissertation, Princeton Univ., Princeton, NJ, 2007.
- [18] R. E. Krichevsky and V. K. Trofimov, “The performance of universal encoding,” *IEEE Trans. Inf. Theory*, vol. IT-27, no. 2, pp. 199–207, Mar. 1981.
- [19] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “Context-tree maximizing,” in *Proc. 2000 Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2000, pp. TP6 7–12.
- [20] L. Paninski, “Estimation of entropy and mutual information,” *Neural Comput.*, vol. 15, pp. 1191–1253, 2003.
- [21] F. M. J. Willems and T. J. Tjalkens, “Complexity reduction of the context-tree weighting algorithm: A study for KPN research,” Tech. Univ. Eindhoven, Eindhoven, The Netherlands, EIDMA Rep. RS.97.01, Jan. 1997.
- [22] P. Volf, “Weighting Techniques in Data Compression: Theory and Algorithm,” Ph.D. dissertation, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 2002.
- [23] N. Chomsky, “Three models for the description of language,” *IRE Trans. Inf. Theory*, vol. IT-2, pp. 113–124, 1956.

**Jiming Yu** received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2002, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 2004 and 2007, respectively.

**Sergio Verdú** (S’80–M’84–SM’88–F’93) received the Telecommunications Engineering degree from the Universitat Politècnica de Barcelona, Barcelona, Spain, in 1980 and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Urbana, IL, in 1984.

Since 1984, he has been a member of the faculty at Princeton University, Princeton, NJ, where he is the Eugene Higgins Professor of Electrical Engineering.

Prof. Verdú is the recipient of the 2007 Claude E. Shannon Award and the 2008 IEEE Richard W. Hamming Medal. He is a member of the National Academy of Engineering and was awarded a Doctorate *Honoris Causa* from the Universitat Politècnica de Catalunya, Barcelona, in 2005. He is a recipient of several paper awards from the IEEE: the 1992 Donald Fink Paper Award, the 1998 Information Theory Outstanding Paper Award, an Information Theory Golden Jubilee Paper Award, the 2002 Leonard Abraham Prize Award, and the 2006 Joint Communications/Information Theory Paper Award. In 1998, Cambridge University Press published his book *Multiuser Detection*, for which he received the 2000 Frederick E. Terman Award from the American Society for Engineering Education. He served as President of the IEEE Information Theory Society in 1997 and as Associate Editor for Shannon Theory of the IEEE Transactions on Information Theory. He is currently Editor-in-Chief of *Foundations and Trends in Communications and Information Theory*.