

Universal Erasure Entropy Estimation

Jiming Yu, Sergio Verdú
 Department of Electrical Engineering
 Princeton University
 Princeton, NJ 08544, USA
 Email: {jimingyu, verdu}@princeton.edu

Abstract—Erasure entropy rate (introduced recently by Verdú and Weissman) 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, universal algorithms for estimating erasure entropy rate are proposed based on the basic and extended context-tree weighting (CTW) algorithms. Consistency results are shown for those CTW based 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. An estimate of the erasure entropy of English texts based on the proposed algorithms is about 0.22 bits per letter, which can be compared to an estimate of about 1.3 bits per letter for the entropy rate of English texts by a similar CTW based algorithm.

I. INTRODUCTION

Erasure entropy rate is an information measure proposed in [1]: the erasure entropy for a finite collection of discrete random variables X_1^n is defined as

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

with the convention that X_a^b is the empty string λ if $a > b$, and the erasure entropy rate (or erasure entropy) of a process $\mathbf{X} = (X_t)_{t \in \mathbb{Z}}$ is defined as the limit

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

whenever it exists. If \mathbf{X} is stationary, then by [1] its erasure entropy rate $H^-(\mathbf{X})$ exists, and $H^-(\mathbf{X}) = H(X_0 | X_{-\infty}^-, X_1^\infty)$. A useful quantity for our purposes is the D -th order erasure entropy for stationary processes:

$$H_D^-(\mathbf{X}) \triangleq H(X_0 | X_{-D}^-, X_1^D).$$

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

Like entropy rate, erasure entropy rate also has operational significance [1], e.g.

- information rate needed to supply to the observer of an erasure channel output in order to losslessly or lossily recover the channel input, and
- denoisability of memoryless channel outputs.

Due to its significance as an information measure with operational meaning, it is desirable to (universally) estimate the erasure entropy rate $H^-(\mathbf{X})$ without any knowledge of the probability law of the process $\mathbf{X} = (X_t)_{t \in \mathbb{Z}}$, based on a finite time realization x_1^n . This paper proposes several universal

estimation algorithms and shows related consistency results for stationary ergodic processes.

The problem of estimating Shannon’s entropy rate of a stationary process has been considered in the literature via many different methods, e.g. [2]–[7]. Many of those entropy estimators cannot be modified to estimate erasure entropy rate. Nevertheless, the CTW based algorithms have shown promise dealing with noncausal/bi-directional conditioning structure in the problem of lossless compression with side information at both the encoder and decoder [8]. In fact, estimating the limit in (1) 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 the side information at both the encoder and the decoder being 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)¹. Based on this observation, we propose erasure entropy rate estimators based on the basic CTW [9] and the extended CTW [10] algorithms. Other approaches are possible by using bi-directional modelers via various kinds of context trees [11]–[14], which are all utilized in the universal discrete denoising setting [12] where a bi-directional model is needed.

II. ALGORITHMS

Let x_1^n be any given finite realization of a process $\mathbf{X} = (X_t)_{t \in \mathbb{Z}}$ taking values on a finite set A . We consider the modified CTW tree in our bi-directional setting (henceforth referred to as bi-directional basic or extended CTW trees), in which each branch is labelled by a pair of symbols (a, b) . A bi-directional context for the symbol z in the sequence $a_1^k z a_{k+1}^{k+m}$ is any pair of strings $(a_{k-l+1}^k, a_{k+1}^{k+l})$ of equal length $l \leq \min\{k, m\}$. A node in this bi-directional CTW tree obtained by travelling from the root through branches labelled by $(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)$ represents the bi-directional context $(a_m a_{m-1} \dots a_1, b_1 b_2 \dots b_m)$, thus the terms a ‘node’ and its corresponding ‘bi-directional context’ are used interchangeably. Let (λ, λ) denote the root node in both the bi-directional basic and extended CTW trees and $l(s)$ denote the depth of node s , $C(s)$ is the set of child nodes of s . Let $|\cdot|$ denote the cardinality of a set or the length of a string on the alphabet A , depending on its argument, in which no confusion will arise. We store in each node s ($s = (s_l, s_r)$), $s_l, s_r \in$

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

$A^* \triangleq \cup_{m=0}^{\infty} A^m$ with $A^0 \triangleq \{\lambda\}$, $|s_l| = |s_r|$ the count $N_s^i(a)$ of number of occurrences of each symbol $a \in A$ that has appeared in the bi-directional context s after updating the tree for the i -th symbol x_i , whose initial value is set to 0 at the beginning of the algorithms. We also store in each node the estimated and weighted probabilities $P_e^s(i)$ and $P_w^s(i)$ after updating the tree for the i -th symbol x_i , whose initial values are all set to be 1, as in CTW [9], [10] except for the explicit time indices. For any node p and q such that p is an offspring of q (not necessarily a direct child), define the path $\{p \rightarrow q\}$ as the set consisting of all nodes along the path in the tree from p back to q (including p and q). The a -th component of a vector $v \in \mathbb{R}^{|A|}$ is denoted by $v[a]$, $a \in A$.

A. One-Pass Basic CTW Based Estimation Algorithm

Assume the past and future D symbols $x_{-D+1}^0, x_{n+1}^{n+D}$ are known as usual [9]. Initially a complete, balanced $|A|^2$ -ary tree is constructed.

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

2) *Step 2*: Start from the root node, go through the branches labelled by $(x_{i-1}, x_{i+1}), (x_{i-2}, x_{i+2}), \dots, (x_{i-D}, x_{i+D})$ to arrive at the node $s = (x_{i-D}^{i-1}, x_{i+1}^{i+D})$.

3) *Step 3*: Update each node along the traversed path:

- Update the estimated probability for node s as:

$$P_e^s(i) = \frac{N_s^{i-1}(x_i) + \frac{1}{2}}{\sum_{a \in A} N_s^{i-1}(a) + \frac{1}{2}|A|} P_e^s(i-1).$$

- Update the counts for the context s as:

$$N_s^i(x_i) = N_s^{i-1}(x_i) + 1, N_s^i(a) = N_s^{i-1}(a), \forall a \neq x_i.$$

- Update the weighted probability for node s as:

$$P_w^s(i) = \begin{cases} \frac{1}{2} P_e^s(i) + \frac{1}{2} \prod_{s' \in C(s)} P_w^{s'}(i) & \text{if } 0 \leq l(s) < D \\ P_e^s(i) & \text{if } l(s) = D. \end{cases}$$

And then we let: $s \leftarrow$ father node of s , if the father node of s exists. We then do the same updating as above for the current node s until we have finally updated the root node. Notice that the only node without a father node is the root.

4) *Step 4*: $i \leftarrow i + 1$, if $i \leq n$, go to *Step 2*, otherwise stop.

The one-pass estimator for the erasure entropy based on the basic CTW is defined as $\frac{1}{n} \hat{H}_{1,D,n}^-(x_{1-D}^{n+D})$ where $\hat{H}_{1,D,n}^-(x_{1-D}^{n+D}) \triangleq \log \frac{1}{P_w^{(\lambda,\lambda)}(n)}$.

B. Two-Pass Basic CTW Based Estimation Algorithm

Lemma 1: For any $i \in \mathbb{N}^*$ let

$$\beta_{i-1}^s \triangleq \frac{P_e^s(i-1)}{\prod_{s' \in C(s)} P_w^{s'}(i-1)} \quad \text{for any node } s,$$

$$\alpha_{i-1}^{(x_{i-d}^{i-1}, x_{i+1}^{i+D})} \triangleq \beta_{i-1}^{(x_{i-d}^{i-1}, x_{i+1}^{i+D})} \times \prod_{q=(x_{i-k}^{i-1}, x_{i+1}^{i+k}): k=0,1,\dots,d} \frac{1}{\beta_{i-1}^q + 1},$$

$$\gamma_{i-1} \triangleq \prod_{q=(x_{i-d}^{i-1}, x_{i+1}^{i+D}): d=0,1,\dots,D-1} \frac{1}{\beta_{i-1}^q + 1},$$

$$\forall a \in A, \hat{P}_i(x_{1-D}^{i+D})[a] \triangleq \gamma_{i-1} \frac{N_v^{i-1}(a) + \frac{1}{2}}{\sum_{b \in A} N_v^{i-1}(b) + \frac{1}{2}|A|} + \sum_{s=(x_{i-d}^{i-1}, x_{i+1}^{i+D}): d=0,1,\dots,D-1} \alpha_{i-1}^s \frac{N_s^{i-1}(a) + \frac{1}{2}}{\sum_{b \in A} N_s^{i-1}(b) + \frac{1}{2}|A|} \quad (2)$$

where $v = (x_{i-D}^{i-1}, x_{i+1}^{i+D})$. Then the conditional distribution in (2) satisfies

$$\prod_{i=1}^n \hat{P}_i(x_{1-D}^{i+D})[x_i] = P_w^{(\lambda,\lambda)}(n). \quad (3)$$

Proofs are omitted because of space limitations. Viewing (2) as the conditional distribution of X_i given $X_{i-D}^{i-1} = x_{i-D}^{i-1}, X_{i+1}^{i+D} = x_{i+1}^{i+D}$ estimated at time i , we can replace every statistic at time i by the corresponding statistic at time n to improve accuracy. This motivates the two-pass basic CTW based algorithm which is a recursive implementation of the modified version of (2).

1) *Step 1*: Construct the bi-directional basic CTW tree for the given realization x_1^n as in Section II-A. Store β_n^q for each node q . Let $i = 1$.

2) *Step 2*: Go through branches corresponding to the sequence of pairs of symbols

$$(x_{i-1}, x_{i+1}), (x_{i-2}, x_{i+2}), \dots, (x_{i-D}, x_{i+D})$$

to reach the node $v = (x_{i-D}^{i-1}, x_{i+1}^{i+D})$.

3) *Step 3*: For each node s in the path $\{v \rightarrow (\lambda, \lambda)\} = (x_{i-k}^{i-1}, x_{i+1}^{i+k})_{k=D}^0$, let $d = D$, we do the following until $d = -1$:

- If $d = D$, define $R_{i,w}^v = R_{i,w}^v(x_{1-D}^{n+D}), R_{i,e}^v = R_{i,e}^v(x_{1-D}^{n+D}) \in [0, 1]^{|A|}$ as

$$R_{i,w}^v[a] = R_{i,e}^v[a] \triangleq \frac{N_v^n(a) + \frac{1}{2}}{\sum_{c \in A} N_v^n(c) + \frac{1}{2}|A|}.$$

If $0 \leq d < D$, let $s = (x_{i-d}^{i-1}, x_{i+1}^{i+d})$ and define $R_{i,e}^s = R_{i,e}^s(x_{1-D}^{n+D}), R_{i,w}^s = R_{i,w}^s(x_{1-D}^{n+D}) \in [0, 1]^{|A|}$ as:

$$R_{i,e}^s[a] \triangleq \frac{N_s^n(a) + \frac{1}{2}}{\sum_{c \in A} N_s^n(c) + \frac{1}{2}|A|},$$

$$R_{i,w}^s[a] \triangleq \frac{\beta_n^s}{\beta_n^s + 1} R_{i,e}^s[a] + \frac{1}{\beta_n^s + 1} R_{i,w}^{s^*}[a]$$

where s^* is the only child of s that is in $\{v \rightarrow (\lambda, \lambda)\}$.

- $d \leftarrow d - 1$.

4) *Step 4*: Set $i \leftarrow i + 1$. If $i \leq n$, go to *Step 2*, otherwise define the basic CTW based two-pass estimator as $\frac{1}{n} \hat{H}_{2,D,n}^-(x_{1-D}^{n+D})$ where $\hat{H}_{2,D,n}^-(x_{1-D}^{n+D}) \triangleq \sum_{i=1}^n \log \frac{1}{R_{i,w}^{(\lambda,\lambda)}[x_i]}$.

C. One-Pass Extended CTW Based Estimation Algorithm

To encompass the case of unbounded memory, the finite realization x_1^n is augmented to a doubly-infinite sequence $\dots \epsilon \epsilon \epsilon x_1^n \epsilon \epsilon \epsilon \dots$ by adding an additional symbol ϵ as in the extended CTW [10]. Thus whenever an index t exceeds n or become smaller than 1, $x_t = \epsilon$. *Null* and *unique* nodes are

defined in the same way as in [10]. Each branch now is labelled by a pair of integers (p, l) which is equivalent to the pair of symbols (x_{p-l}, x_{p+l}) . Here p represents the most recent time that this branch has been visited, and l represents the level of the node that this branch leads to. A node s obtained by traversing the branches $(p_1, l_1), (p_2, l_2), \dots, (p_m, l_m)$ from the root now represents the bi-directional context $(x_{p_m-l_m} x_{p_{m-1}-l_{m-1}} \dots x_{p_1-l_1}, x_{p_1+l_1} x_{p_2+l_2} \dots x_{p_m+l_m})$, and from this we identify the node s with the *ordered* sequence of pairs of integers $(p_1, l_1), (p_2, l_2), \dots, (p_m, l_m)$, which corresponds to a unique bi-directional context.

We first write down an *Updating Procedure for Symbol b at Time i* for a node s and its path back to the root node and then describe the algorithm that uses this procedure repeatedly.

Procedure 1 (Updating Procedure for Symbol b at Time i): Update the estimated probability for node s as:

$$P_e^s(i) = \frac{N_s^{i-1}(b) + \frac{1}{2}}{\sum_{a \in A} N_s^{i-1}(a) + \frac{1}{2}|A|} P_e^s(i-1).$$

Update the counts for this context s as:

$$N_s^i(b) = N_s^{i-1}(b) + 1, N_s^i(c) = N_s^{i-1}(c), \forall c \neq b.$$

Update the weighted probability for node s as:

$$P_w^s(i) = \begin{cases} \frac{1}{2} P_e^s(i) + \frac{1}{2} \prod_{s' \in C(s)} P_w^{s'}(i) & \text{if } \sum_{a \in A} N_s^i(a) > 1 \\ P_e^s(i) = \frac{1}{|A|} & \text{if } \sum_{a \in A} N_s^i(a) = 1. \end{cases}$$

For the branch (p, l) leading from s backward to its father node (if this branch exists), we let $p = i$. And then we let: $s \leftarrow$ father node of s , if the father node of s exists. And the *Updating Procedure for Symbol b at Time i* ends here.

1) *Step 1:* Let $i = 1$. Go through the branch $(i, 1)$ to the node s corresponding to the bi-directional context (x_{i-1}, x_{i+1}) , update the counts as $N_s^i(x_i) = 1, N_s^i(a) = 0, \forall a \neq x_i$. Update the estimated and weighted probabilities in this node s and the root node (λ, λ) as $P_w^{(\lambda, \lambda)}(i) = P_e^{(\lambda, \lambda)}(i) = P_w^s(i) = P_e^s(i) = \frac{1}{2}$.

2) *Step 2:* Start from the root node, go through branches labelled by $(i, 1), (i, 2), (i, 3), \dots$ until a *null* node s is encountered. Let s' be the father node of this null node s . There are two cases to be considered:

- Case 1: If s' is the root or s' is *not* a unique node. Update the whole path from root leading to s according to the Updating Procedure 1 for symbol x_i at time i until we have finally updated the root node.
- Case 2: If s' is not the root node and s' is a unique node. We need to *branch* at this node s or at one of its offsprings (not necessarily a direct child) with the same reasons as in [10]. Let $s = (p_k, k)_{k=1}^m$, let $j = m$, we do $j \leftarrow j - 1$ until $(x_{p_m-j}, x_{p_m+j}) \neq (x_{i-j}, x_{i+j})$. Then we need to branch the node $q = ((p_k, k)_{k=1}^m, (p_m, k)_{k=m+1}^{j-1})$ at level $j-1$ to the two child nodes $s_1 = ((p_k, k)_{k=1}^m, (p_m, k)_{k=m+1}^j)$, $s_2 = (i, k)_{k=1}^j$. Let $N_{s_1}^i(a) = N_{s_2}^{i-1}(a), N_{s_2}^{i-1}(a) = 0, \forall a \in A$ and

$$P_w^{s_1}(i) = P_w^s(i-1), P_e^{s_1}(i) = P_e^s(i-1), P_w^{s_2}(i-1) = P_e^{s_2}(i-1) = 1.$$

Let the branch from q to s_1 be labelled by (p_m, j) , let the branch from q to s_2 be labelled by (i, j) . Let $s = s_2$ and then update the whole path from s back to the root node according to the Updating Procedure 1 for symbol x_i at time i until we have finally updated the root.

Notice that the only node without a father node is the root.

3) *Step 4:* $i \leftarrow i + 1$, if $i \leq n$, go to *Step 2*, otherwise stop.

The one-pass estimator for the erasure entropy based on the extended CTW is defined as $\frac{1}{n} \hat{H}_{1,n}^-(x_1^n)$ where $\hat{H}_{1,n}^-(x_1^n) \triangleq \log \frac{1}{P_w^{(\lambda, \lambda)}(n)}$.

D. Two-Pass Extended CTW Based Estimation Algorithm

A result similar to Lemma 1 for the bi-directional version of extended CTW motivates the two-pass extended CTW based erasure entropy estimation algorithm. The formula corresponding to (2) can be easily deduced by using recursions in *Step 3* below. We describe this algorithm in full despite its similarity to its basic CTW based counterpart.

1) *Step 1:* Construct the bi-directional extended CTW tree for the realization x_1^n as in Section II-C. Store β_n^q for all non-null nodes q in this tree. Let $i = 1$.

2) *Step 2:* Go through branches corresponding to the sequence of pairs of symbols

$$(x_{i-1}, x_{i+1}), (x_{i-2}, x_{i+2}), (x_{i-3}, x_{i+3}), \dots$$

until we have reached the deepest non-null node v that fits into our data in the bi-directional extended CTW tree.

3) *Step 3:* For each node s in this path $\{v \rightarrow (\lambda, \lambda)\}$, do the following until we have updated the root node (λ, λ) :

- If $s = v$, define $R_{i,w}^v(x_1^n), R_{i,e}^v(x_1^n) \in [0, 1]^{|A|}$ as:

$$R_{i,w}^v(x_1^n)[a] = R_{i,e}^v(x_1^n)[a] \triangleq \frac{N_v^n(x_i) + \frac{1}{2}}{\sum_{c \in A} N_v^n(c) + \frac{1}{2}|A|}.$$

If $s \neq v$, define $R_{i,e}^s(x_1^n), R_{i,w}^s(x_1^n) \in [0, 1]^{|A|}$ as:

$$R_{i,e}^s(x_1^n)[a] \triangleq \frac{N_s^n(a) + \frac{1}{2}}{\sum_{c \in A} N_s^n(c) + \frac{1}{2}|A|},$$

$$R_{i,w}^s(x_1^n)[a] \triangleq \frac{\beta_n^s}{\beta_n^s + 1} R_{i,e}^s(x_1^n)[a] + \frac{1}{\beta_n^s + 1} R_{i,w}^{s^*}(x_1^n)[a]$$

where s^* is the only child of s that is in $\{v \rightarrow (\lambda, \lambda)\}$.

- $s \leftarrow$ father node of s .

4) *Step 4:* Set $i \leftarrow i + 1$. If $i \leq n$, go to *Step 2*, otherwise stop and let the two-pass extended CTW based erasure entropy estimator be defined as $\frac{1}{n} \hat{H}_{2,n}^-(x_1^n)$ where $\hat{H}_{2,n}^-(x_1^n) \triangleq \sum_{i=1}^n \log \frac{1}{R_{i,w}^{(\lambda, \lambda)}(x_1^n)[x_i]}$.

III. CONSISTENCY RESULTS

Theorem 1: Let $\frac{1}{n}\hat{H}_{1,D,n}^- : A^{n+2D} \rightarrow \mathbb{R}_+$ be the one-pass basic CTW based estimator as in Section II-A. Suppose \mathbf{X} is stationary ergodic, then

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

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

Theorem 2: Let $\frac{1}{n}\hat{H}_{2,D,n}^- : A^{n+2D} \rightarrow \mathbb{R}_+$ be the two-pass basic CTW based estimator as in Section II-B. Suppose \mathbf{X} is stationary ergodic. Then

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

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

Theorem 3: Let $\frac{1}{n}\hat{H}_{1,n}^- : A^n \rightarrow \mathbb{R}_+$ be the one-pass extended CTW based estimator as in Section II-C. Suppose \mathbf{X} is stationary ergodic, then

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

IV. SIMULATIONS

We test our algorithms with synthetic (Markov and tree) sources and English texts. Each point in the figures 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.

The algorithms based on the basic CTW have $O(n)$ time and space complexity for a given memory length D . The algorithms based on the extended CTW can be turned into a bi-directional counterpart of the full-power uni-directional extended CTW [10] with $O(n)$ space complexity and this does not affect the estimates. If $|A| > 2$, it is necessary to modify CTW introducing a binary symbol decomposition method [15], [16] in order to achieve good estimation performance.

The basic CTW based algorithms use a heuristic that chooses its memory length parameter D : find the smallest D such that $\frac{1}{n}\hat{H}_{k,D-1,n}^- \leq \frac{1}{n}\hat{H}_{k,D,n}^- \leq \frac{1}{n}\hat{H}_{k,D+1,n}^-$, $k = 1, 2$ for data with length n . From the simulations for Markov sources and tree sources below, we can see that this heuristic directly leads to the correct choice of D for the one-pass basic CTW based estimator, namely in most cases it *exactly* chooses the true memory length or Markov order. For the two-pass basic CTW based estimator, it tends to overestimate the order, but as long as the data length is long enough, it still converges to the right estimate.

A. Markov Source Simulations

Consider a second-order Markov source on an alphabet of size 4 with strictly positive transition probabilities (omitted here). Its exact erasure entropy rate is (computed by using their *known* statistics) 1.20897 bits per symbol, whereas the entropy rate is 1.66365 bits per symbol.

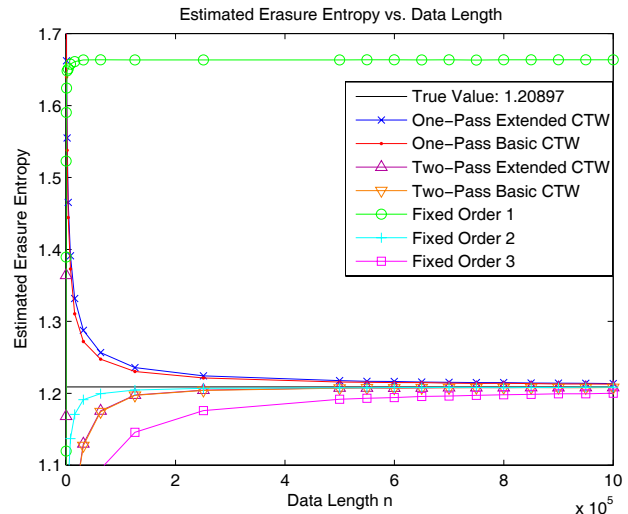


Fig. 1. 2nd Order Markov Source with Alphabet Size 4

In Fig. 1, the straight horizontal line is the true erasure entropy. For large lengths, the one-pass basic and extended CTW based algorithms have essentially the same performance, whereas for moderate lengths the one-pass basic CTW based algorithm with our heuristic works slightly better. The two-pass approaches are better than the one-pass approaches all the time, and the two-pass basic and extended CTW based algorithms have nearly coincident curves. The fixed-order estimator of order 2 is superior than any other algorithm at any length because of its *a priori* information about the exact (and small) order of the process. Note from Fig. 1 that the fixed-order estimators suffer from severe performance degradation when the model order is either underestimated or overestimated (order is 2).

B. Tree Source Simulations

A binary tree source as in [7] with maximal memory length 11 and a suffix set of size 15 is used to test our algorithms (parameter values are omitted here). Its exact erasure entropy is (computed by using their *known* statistics) 0.27213 bits per symbol, whereas the entropy rate is 0.446894 bits per symbol.

In Fig. 2, the straight horizontal line is the true erasure entropy. The basic and extended CTW based algorithms have essentially the same performance at all those lengths for both the one-pass and two-pass approaches, with the two-pass approaches being better than the one-pass approaches. Those fixed-order estimators perform poorly even with the correct *a priori* information about the order of the tree source.

C. English Text Experiments

We experiment with 15 English texts (as in [14]) of various lengths, ranging from about 9×10^5 to 3×10^6 characters, as shown in Table I. The simulation has been conducted with the two-pass extended CTW based algorithm, the two-pass basic CTW based algorithm with $D = 1, 2, 3, 4, 5, 6, 7, 8$. The 2nd column corresponds to results of the two-pass basic

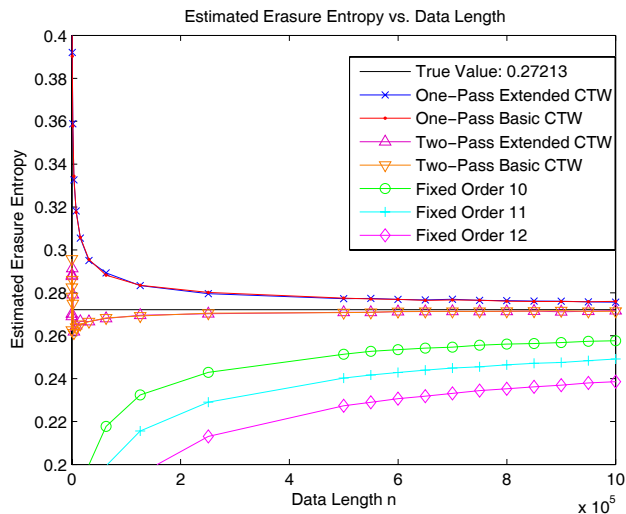


Fig. 2. Binary Tree Source with Memory Length 11 and Suffix Set Size 15

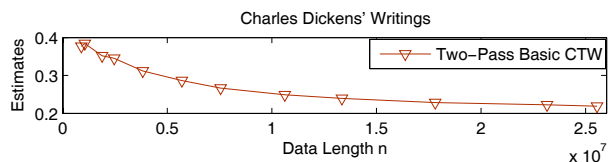


Fig. 3. English Texts by Charles Dickens: Erasure Entropy (Bits per Letter)

CTW based algorithm with the heuristic explained above. The numbers in parentheses are the memory lengths chosen by the heuristic. The ‘Extended CTW’ column represents the estimates by the two-pass extended CTW based algorithm. The last 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 any finite-order Markov approximation does not fully characterize their dependence structure, the convergence of our estimators is slower than the case of synthetic sources. This can be seen from Fig. 3, where we plot the estimates from the two-pass basic CTW based estimator with memory length $D = 4$ versus the data lengths for 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 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 2.55×10^7 (by concatenating many pieces of Charles Dickens’ writings), the estimated erasure entropy decreases to about 0.22 bits per letter.

REFERENCES

- [1] S. Verdú and T. Weissman, “Erasure entropy,” in *Proc. IEEE Int. Symp. Information Theory*, Seattle, Washington, July 2006.
- [2] C. E. Shannon, “Prediction and entropy of printed English,” *Bell Syst. Techn. J.*, vol. 30, pp. 50–64, 1951.
- [3] T. M. Cover and R. C. King, “A convergent gambling estimate of the entropy of English,” *IEEE Trans. Inform. Theory*, vol. 24, pp. 413–421, July 1978.

No.	Basic CTW (D)	Extended CTW	Entropy
1	0.33 (7)	0.33	1.39
2	0.33 (6)	0.33	1.40
3	0.40 (6)	0.40	1.46
4	0.30 (7)	0.30	1.26
5	0.47 (5)	0.47	1.57
6	0.34 (7)	0.34	1.39
7	0.34 (6)	0.34	1.42
8	0.48 (5)	0.48	1.54
9	0.45 (5)	0.45	1.44
10	0.30 (6)	0.30	1.29
11	0.31 (7)	0.31	1.36
12	0.34 (5)	0.34	1.30
13	0.35 (6)	0.35	1.40
14	0.34 (6)	0.34	1.33
15	0.48 (5)	0.48	1.50

TABLE I

ERASURE ENTROPY AND ENTROPY FOR ENGLISH TEXTS

- [4] P. Grassberger, “Estimating the information content of symbol sequences and efficient codes,” *IEEE Trans. Inform. Theory*, vol. 35, pp. 669–675, May 1989.
- [5] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner, “Nonparametric entropy estimation for stationary processes and random fields, with applications to English text,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1319–1327, May 1998.
- [6] H. Cai, S. R. Kulkarni, and S. Verdú, “Universal entropy estimation via block sorting,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1551–1561, July 2004.
- [7] H. Cai, S. R. Kulkarni, and S. Verdú, “Universal divergence estimation for finite-alphabet sources,” *Submitted to IEEE Trans. on Inform. Theory*, 2005.
- [8] 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, Sept. 2005.
- [9] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, “The context-tree weighting method: Basic properties,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 653–664, May 1995.
- [10] F. M. J. Willems, “The context-tree weighting method: Extensions,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 792–798, Mar. 1998.
- [11] M. J. Weinberger, J. J. Rissanen, and M. Feder, “A universal finite memory source,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 643–652, May 1995.
- [12] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, “Universal discrete denoising: Known channel,” *IEEE Trans. Inform. Theory*, vol. 51, pp. 5–28, Jan. 2005.
- [13] 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, Sept. 2005.
- [14] J. Yu and S. Verdú, “Schemes for bi-directional modeling of discrete stationary sources,” in *Proc. 39th Annual Conference on Information Science and Systems*, Baltimore, MD, Mar. 2005.
- [15] F. M. J. Willems and T. J. Tjalkens, “Complexity reduction of the context-tree weighting algorithm: A study for KPN research,” *EIDMA Report Series: EIDMA-RS. 97. 01*, Euler Institute of Discrete Mathematics and its Applications, Jan. 1997.
- [16] P. Volf, *Weighting Techniques in Data Compression: Theory and Algorithm*. PhD thesis, Technische Universiteit Eindhoven, 2002.