# Convergence of Binarized Context-tree Weighting for Estimating Distributions of Stationary Sources

Badri N. Vellambi, Marcus Hutter
Australian National University
{badri.vellambi, marcus.hutter}@anu.edu.au

***Abstract*—This work investigates the convergence rate of learning the stationary distribution of finite-alphabet stationary ergodic sources using a binarized context-tree weighting approach. The binarized context-tree weighting ($\overline{\text{CTW}}$) algorithm estimates the stationary distribution of a symbol as a product of conditional distributions of each component bit, which are determined in a sequential manner using the well known binary context-tree weighting method. We establish that $\overline{\text{CTW}}$ algorithm is a consistent estimator of the stationary distribution, and that the worst-case $L_1$-prediction error between the $\overline{\text{CTW}}$ and frequency estimates using $n$ source symbols each of which when binarized consists of $k > 1$ bits decays as $\Theta\left(\sqrt{2^k \frac{\log n}{n}}\right)$.**

## I. Introduction

The binary context-tree weighting (CTW) algorithm proposed in [1], [2] is a universal data compression scheme that uses a Bayesian mixture model over all context trees of up to a certain design depth to assign probabilities to sequences. The probability estimate and the weights corresponding to each tree assigned by the CTW algorithm is derived efficiently using the well known Krichevsky-Trofimov (KT) estimator that models source bits as having a Bernoulli distribution whose parameter has a Dirichlet prior [3]. The CTW algorithm offers two attractive features:

- optimal compression for binary tree sources [4, Thm. 1];
- superior redundancy bounds and amenability to analysis as opposed to other universal compression schemes such as the Lempel-Ziv algorithm [5], [6], e.g., it was shown in [2, Thm. 2] that for any binary tree source $P$ with $\mathsf{C}$ context parameters, the redundancy $\rho(\cdot) := \log_2 \frac{P(\cdot)}{P_{CTW}(\cdot)}$ of the binary CTW algorithm is bounded by

$$\max_{x_{1:n}} \rho(x_{1:n}) \leq \mathsf{C}\left(\tfrac{1}{2}\log_2 \tfrac{n}{\mathsf{C}} + 1\right) + (2\mathsf{C} - 1) + 2. \quad (1)$$

The first term is the asymptotically dominant optimal redundancy achievable for the source $P$, and the second term $(2\mathsf{C} - 1)$ is the *model redundancy*, i.e., the price paid for weighting various tree models as opposed to using the actual source model if it were indeed known.

An extension of CTW to sources over non-binary alphabets was proposed in [7], [8], and a redundancy bound similar to (1) for a tree source with $\mathsf{C}$ contexts over a finite alphabet $\mathcal{A}$ was derived; the asymptotically dominant term in this bound for a general alphabet $\mathcal{A}$ was established to be $\frac{\mathsf{C}(|\mathcal{A}|-1)}{2}\log_2 \frac{n}{\mathsf{C}(|\mathcal{A}|-1)}$. In [7], a decomposition-based approach that uses binarization to map symbols from large alphabets to bit strings was also summarily proposed; this approach was later analyzed in greater detail in [9].

There are several motivating factors for taking a closer look into binarization. First, many data sources of interest such as text, image, and audio files are factored and represented into smaller units (such as bits or bytes) for digital storage. Hence, despite the fact that a source of practical interest might be distributed over a large alphabet, its symbols can effectively be viewed as a tuple of binary random variables. Second, the decomposition of larger alphabets into appropriate binary strings translates the CTW algorithm on the large alphabet to a sequence of binary CTW problems. This translation in turn has the potential of reducing the number of model parameters and can consequently offer improved redundancy guarantees [9]–[11]. This is better understood via an example. Consider an i.i.d. source over a hexadecimal alphabet whose distribution

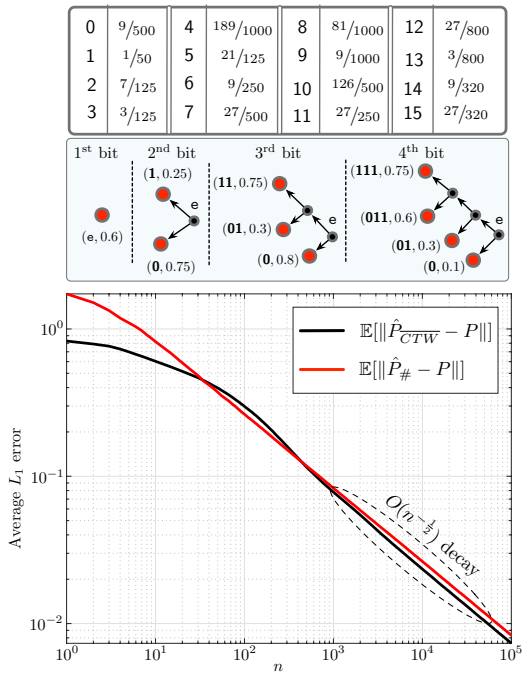


Fig. 1. Performance of the binarized CTW algorithm for a tree source.

is given in the table in Fig. 1. While no specific structure is apparent from the table, the example corresponds to a binary tree source with each source symbol comprising of

This work was supported by the Australian Research Council Discovery Projects DP120100950 and DP150104590.

four bits, the first of which is an i.i.d. Bernoulli random variable with parameter $0.6$. For $i = 2, 3, 4$, the $i^{\text{th}}$ bit potentially depends on the realization of the first $i-1$ bits, and the conditional distribution of the $i^{\text{th}}$ bit is a Bernoulli distribution that depends on its context, e.g., the third bit is a Bernoulli random variable with parameter $0.75$ if the first two bits are both ones. The naïve CTW over the hexadecimal alphabet looks for no structure within each symbol, and hence must learn $15$ parameters, whereas the dominant trees for each of the 4 bits in the binarized factored CTW together have to learn only 10 parameters, which yields improvements in terms of redundancy. In general, for a tree source with alphabet $\mathcal{A}$ and with $\mathsf{C}$ contexts, the asymptotic dominant term for redundancy can reduce from $\frac{\mathsf{C}(|\mathcal{A}|-1)}{2}\log_2\frac{n}{\mathsf{C}(|\mathcal{A}|-1)}$ to $\sum_{i=1}^{\lceil\log_2|\mathcal{A}|\rceil}\frac{\mathsf{C}_i}{2}\log_2\frac{n}{\mathsf{C}_i}$, where $\mathsf{C}_i \leq 2^{i-1}\mathsf{C}$ is the number of contexts in the tree of the $i^{\text{th}}$ bit in the binary decomposition. The scaling factor of the $\log_2 n$ can potentially reduce from $\frac{\mathsf{C}(|\mathcal{A}|-1)}{2}$ to $\frac{\mathsf{C}\lceil\log_2|\mathcal{A}|\rceil}{2}$, which can be significant when compressing sources with large alphabets. As an illustration, again consider the i.i.d source $P$ of Fig. 1. Let $\hat{P}_{\#}$ denote the frequency estimate of $P$. Let $\hat{P}_{\overline{\text{CTW}}}$ denote the binarized factored CTW (for the complete definition, see Sec. III). Fig. 1 presents the variation of mean $L_1$-prediction error with the number of samples $n$; the following observations can be made.

- $\overline{\text{CTW}}$ is a better estimator than a naïve frequency (ML) estimator when the size $n$ of data is small, or sufficiently large ($n \lesssim 30$ or $n \gtrsim 450$ in the above example). The improved prediction error for small data size can be attributed to Bayesian averaging inherited by the $\overline{\text{CTW}}$ algorithm from the CTW algorithm whereas the frequency estimator does not have enough samples to predict accurately. When the data size is large, the contribution of the tree model that best describes the data dominates the $\overline{\text{CTW}}$ algorithm. The $\overline{\text{CTW}}$ estimator therefore offers a scaling improvement depending of the number of parameters needed to describe the '*best*' tree model, which is, in general, fewer than that required to naïvely describe the distribution completely.
- On average, the $\overline{\text{CTW}}$ and frequency estimators offer similar asymptotic convergence rates, which in this example (and for discrete memoryless sources) decay as $\frac{1}{\sqrt{n}}$.

Note that while there is a potential benefit in terms of redundancy, the computational cost of implementing the naïve CTW algorithm over the larger alphabet vs the binarized CTW are similar. In [9], [10], the focus is primarily to identify binarized decompositions that *efficiently* translate the elements of the larger alphabet as a binary string and offer good redundancy guarantees, and not on the estimation error or convergence rates offered by the binarized CTW algorithm. Recently, Veness et al. explored the idea that a good data compression scheme is a good estimator, and applied it to the problem of policy evaluation and control in reinforcement learning problems [12]. This work translates policy evaluation and on-policy control in reinforcement learning to the estimation of the stationary distribution of a hidden Markov model whose state corresponds to the concatenation of the state, action and return in the reinforcement learning problem. The following summarizes the main result of this approach.

- A consistent estimator of the stationary distribution of the state of the derived hidden Markov model, whose estimation error converges stochastically to zero suitably fast results in a consistent estimator of the state-action value function.

Simulations in [12] revealed two benefits from binarizing large state spaces, and using a binarized factored CTW:

- excellent convergence rate in estimating the stationary distribution of the underlying process;
- the ability to effectively handle considerably larger state spaces than the frequency estimator.

In this work, we assume that the source symbols take values from a vector binary alphabet[1] (i.e., $\{0,1\}^k$ for some $k \in \mathbb{N}$), and derive some fundamental properties of the binarized CTW estimate of the source distribution, thereby providing a theoretical reasoning to why the binarized CTW estimate performs well in simulations [12]. We establish the following.

- The worst-case $L_1$-prediction error between the binarized CTW and frequency (ML) estimates for the stationary distribution of a stationary ergodic source over $\{0,1\}^k$ for some $k > 1$ is $\Theta\Big(\sqrt{2^k\frac{\log n}{n}}\Big)$;
- The binarized CTW method yields a consistent estimator of the stationary distribution of stationary ergodic sources; and
- Any (Bayesian) dependency structure that exists in the binary representation of source symbols can be naturally incorporated in the binarized CTW algorithm.

The rest of the paper is organized as follows. Section II presents the definitions and notation used, and Section III defines the binarized CTW algorithm. Section IV presents relevant background results regarding the contribution of various context trees in the CTW estimate, and finally, Section V presents the main results pertaining to binarized CTW. For want of space, all proofs are relegated to the full version [13].

## II. Notation and Definitions

Given a finite alphabet (set) $\mathcal{A}$ and $k \in \mathbb{N}$, let $\mathcal{A}^{<k+1} = \mathcal{A}^{\leq k} := \mathcal{A}^0 \cup \mathcal{A} \cup \mathcal{A}^2 \cup \cdots \cup \mathcal{A}^k$, where $\mathcal{A}^0 := \{\mathsf{e}\}$ and $\mathsf{e}$ is the empty string. For a string $\boldsymbol{a} \in \mathcal{A}^{\leq k}$, $\|\boldsymbol{a}\|$ denotes its length. Random variables are denoted by upper case letters ($A$, $B$, $Z$, etc), and their realizations by lower case letters ($a$, $b$, $c$, etc), and the corresponding alphabets by calligraphic font letters ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{Z}$, etc). A finite sequence of the first $n$ random variables from a random process (also referred to as a source) $\{Z_i\}_{i\in\mathbb{N}}$ is denoted by $Z_{1:n}$, and by the preceding notation, its realization is given by $z_{1:n}$. Since the estimators in this work will be based on counting the frequencies of occurrence of various substrings, we need the following notation.

[1] In this work, the source alphabet $\mathcal{Z}$ is assumed to be mapped to $\{0,1\}^k$ for some $k \in \mathbb{N}$ via a *meaningful* binarization procedure. The results hold even if we *factorize* the alphabet into a finite Cartesian product of finite sets and employ an appropriate non-binary CTW for each component.

*Definition 1:* Given sequence $z_{1:n}$ where each $z_i \in \mathcal{A}^k$, string $\mathbf{a} \in \mathcal{A}^{\leq k}$, and $m \geq \|\mathbf{a}\|$, we let

$$\#_m\mathbf{a} := |\{i : z_i = \mathbf{bac} \text{ for some } \mathbf{b} \in \mathcal{A}^{m-\|\mathbf{a}\|}, \mathbf{c} \in \mathcal{A}^{k-m}\}|.$$

Specifically, if $\mathbf{a} \in \mathcal{A}^k$, then $\#_k\mathbf{a}$ denotes the frequency of the vector symbol $\mathbf{a}$ appearing in the length-$n$ sequence $z_{1:n}$, and if $\mathbf{a} \in \mathcal{A}^{k'}$ for $k' < k$, then $\#_k\mathbf{a}$ denotes the sum of the frequencies of all vectors ending in $\mathbf{a}$ that appear in the length-$n$ sequence.

*Example 1:* Let $k = 3$, $n = 12$, $\mathcal{A} = \{0, 1\}$, and

$$z_{1:n} = \big((000), (111), (011), (100), (101), (101), (000), (111), (011), (001), (001), (111)\big).$$

Then, $\#_3000 = 2$; $\#_3101 = 2$; $\#_3110 = 0$; $\#_300 = 3$; $\#_301 = 4$; $\#_30 = 3$; $\#_31 = 9$; $\#_200 = 4$; $\#_211 = 3$; $\#_20 = 7$; $\#_21 = 5$; $\#_10 = 6$; and $\#_11 = 6$. ■

Since the CTW estimate is a weighted average of tree source estimates, we also require the following notation.

*Definition 2:* For $k \in \mathbb{N}$, let $\mathcal{T}_k$ denote the set of all context trees of depth at most $k$, i.e., a subset $T \subseteq \{0, 1\}^{\leq k}$ is an element of $\mathcal{T}_k$ iff the following two conditions hold:

C1. No element of $T$ is a proper suffix of another element of $T$, i.e., if $\boldsymbol{x}, \boldsymbol{y} \in T$ and $\boldsymbol{x} = \boldsymbol{cy}$, then $\boldsymbol{x} = \boldsymbol{y}$; and

C2. Every string of length $k$ has a suffix that lies in $T$, i.e., for every $\boldsymbol{b} \in \{0, 1\}^k$, there exists (a unique) $\boldsymbol{a} \in T$ and binary string $\boldsymbol{c}$ such that $\boldsymbol{b} = \boldsymbol{ca}$.

We term $T_k^* := \{0, 1\}^k \in \mathcal{T}_k$ the *complete* tree of depth $k$. In this work, the term complete refers solely to the membership of all strings of length $k$ in the tree, as opposed to the condition C2, which is called *completeness* in [1].

*Remark 1:* For any $k \in \mathbb{N}$ and $T \in \mathcal{T}_k$, $\sum_{\boldsymbol{a} \in T} 2^{-\|\boldsymbol{a}\|} = 1$.

Lastly, natural logarithms and logarithms to the base 2 are denoted by $\log$ and $\log_2$, respectively.

## III. Binarized Context-tree Weighting Estimator

In this work, we simply assume that the random process (source) $\{Z_i\}_{i \in \mathbb{N}}$ under study is stationary, ergodic, and takes values over a vector binary alphabet $\{0, 1\}^k$ for some $k \geq 1$. For $i = 1, \ldots, k$, we let $\{B_{i\ell}\}_{\ell \in \mathbb{N}}$ to denote the $i^{\text{th}}$ component random process. The goal is to study the convergence properties of estimating the stationary distribution of the random process $Z$ using the binarized context-tree weighting ($\overline{\text{CTW}}$) estimator defined below. Given $n$ samples of the source $z_{1:n} = ((b_{11}, \ldots, b_{1k}), (b_{21}, \ldots, b_{2k}), \ldots, (b_{n1}, \ldots, b_{nk}))$, the aim is to estimate the stationary distribution. A natural and simple way to estimate the distribution is by the empirical frequency of the appearances of various symbols (binary tuples of length $k$), which is well known *frequency estimator* and is denoted by $\hat{P}_{\#}$. The frequency estimator assigns zero probability to (infrequent) symbols that haven't appeared in the data sample $z_{1:n}$. This is remedied in the Krichevsky-Trofimov (KT) or Dirichlet or the add-half estimator that models the source as an i.i.d multinomial/categorical distribution with a Dirichlet prior, and assigns to each source sequence a probability using the prior predictive distribution. In the binary case, the KT-estimate for a sequence with $\#_10$ zeros and $\#_11$ ones equals

$$P_e(\#_10, \#_11) := \frac{\Gamma(\#_10 + \frac{1}{2})\Gamma(\#_11 + \frac{1}{2})}{\pi\Gamma(n+1)}, \tag{2}$$

where $\Gamma(\cdot)$ is the well known Gamma function.

The binarized CTW ($\overline{\text{CTW}}$) estimator, illustrated in Fig. 2, estimates each of the $k$ bits sequentially by using $k$ binary CTW estimators. The $\overline{\text{CTW}}$ estimator uses the KT-estimator (equivalently, the binary CTW estimator of memory/order 1) to estimate the distribution of the first bit as

$$\hat{P}^{(1)}_{\overline{\text{CTW}}}(B_1 = \beta; z_{1:n}) := \big(\#_1\beta + \tfrac{1}{2}\big)/(n+1), \tag{3}$$

where $\#_1\beta$, by Definition 1, is the number of symbols in $z_{1:n}$ with the prefix $\beta \in \{0, 1\}$. Note that $\hat{P}^{(1)}_{\overline{\text{CTW}}}$ depends only on the bits $(b_{11}, b_{21}, \ldots, b_{n1})$. For $\ell > 1$, the (conditional) distribution of $\ell^{\text{th}}$ bit is estimated sequentially using the binary CTW estimator of memory/order $\ell$ using the first $\ell - 1$ bits as its context. Specifically, for any context $\beta_{1:\ell-1} \in \{0, 1\}^{\ell-1}$, the (conditional) distribution of the $\ell^{\text{th}}$ bit is given by the following weighted average of appropriate KT-estimates

$$\begin{aligned}\hat{P}^{(\ell)}_{\overline{\text{CTW}}}&(B_\ell = \beta \mid B_{1:\ell-1} = \beta_{1:\ell-1}; z_{1:n}) \\ &= \sum_{T \in \mathcal{T}_{\ell-1}} \Lambda_T^{(\ell)} \left(\frac{\#_\ell\xi_T(\beta_{1:\ell-1})\beta + \frac{1}{2}}{\#_{\ell-1}\xi_T(\beta_{1:\ell-1}) + 1}\right), \ \beta \in \{0, 1\},\end{aligned} \tag{4}$$

where $\xi_T(\beta_{1:\ell-1})$ is the unique element of $T$ that is also a suffix of $\beta_{1:\ell-1}$, and $\Lambda_T^{(\ell)}$ is the normalized weight corresponding to tree $T$ assigned by the binary CTW estimator. For each context tree $T \in \mathcal{T}_{\ell-1}$, the normalized weight $\Lambda_T^{(\ell)}$ is given by

$$\Lambda_T^{(\ell)} := \frac{\omega_T P_n^{(\ell)}(T, z_{1:n})}{\sum_{T' \in \mathcal{T}_{\ell-1}} \omega_{T'} P_n^{(\ell)}(T', z_{1:n})}, \tag{5}$$

where for any $T \in \mathcal{T}_\ell$, $\omega_T$ is the non-negative part of the weight that the CTW algorithm associated with a tree $T$ that is independent of the sequence $z_{1:n}$ [2]. The *sequence-dependent part* of the weight $P_n^{(\ell)}(T, z_{1:n})$ is given by

$$P_n^{(\ell)}(T, z_{1:n}) := \prod_{\boldsymbol{a} \in T} P_e(\#_\ell\boldsymbol{a}0, \#_\ell\boldsymbol{a}1), \tag{6}$$

where $P_e(\cdot, \cdot)$ is defined in (2). A closer look at (3)-(6) reveals that just as in the case of the first bit, the $\overline{\text{CTW}}$ estimate for the conditional distribution of the $\ell^{\text{th}}$ bit depends only on the context and the partial history $(b_{i1}, \ldots, b_{i\ell})$, $i = 1, \ldots, n$. Lastly, the $k$ (conditional) distributions are then pieced together to obtain the $\overline{\text{CTW}}$ estimate for the stationary distribution of the source as follows. For each $\beta_{1:k} \in \{0, 1\}^k$,

$$\hat{P}_{\overline{\text{CTW}}}(\beta_{1:k}; z_{1:n}) \triangleq \prod_{\ell=1}^{k} \hat{P}^{(\ell)}_{\overline{\text{CTW}}}(B_\ell = \beta_\ell | B_{1:\ell-1} = \beta_{1:\ell-1}; z_{1:n}). \tag{7}$$

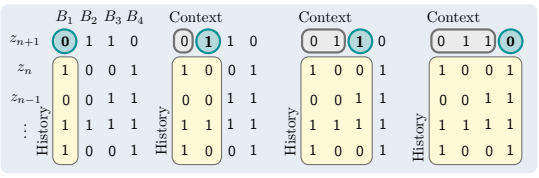


Fig. 2. Some details of the binarized CTW estimator.

Two things must be remarked at this juncture. First, the $\overline{\text{CTW}}$ estimator is a *biased* estimator of the distribution of the source. The bias is inherited directly from the KT estimator, and can be seen by simply considering a binary i.i.d. Bernoulli source with parameter $\theta \neq \frac{1}{2}$, for which

$$\mathbb{E}\,\hat{P}_{\overline{\text{CTW}}}(Z=1;z_{1:n}) = \frac{\mathbb{E}\,\#_1 1 + \frac{1}{2}}{n+1} = \frac{n\theta+\frac{1}{2}}{n+1} \neq \theta. \quad (8)$$

Second, the binarized CTW estimator defined above *learns* the stationary distribution of the source process, neglecting the statistical correlation *between* symbols of the source. Correlation between symbols can be incorporated into the binarized CTW estimator by allowing more factors in (7).

## IV. Comparing Contributions of Trees in CTW

Since the binarized CTW algorithm estimates the distribution of all but the first bit as a weighted average (i.e., mixture) of KT estimates of corresponding different tree sources, the behavior of the $\overline{\text{CTW}}$ estimate will be shaped by

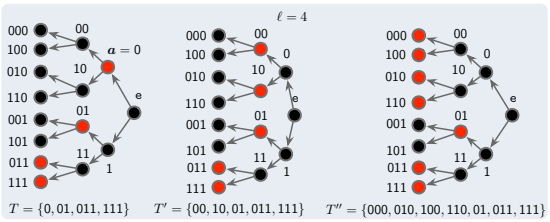


Fig. 3. An example of trees compared in the first and second stages.

the contributions of the dominant context trees, which need to be identified. One way to identify such trees is to develop tools to compare the ratio of weights of any two trees. To do just that, we proceed in three stages. In each stage, we use the bound developed in the earlier stage for a limited class of pairs of trees to generalize to a bound for a larger class of pairs of trees; after the third stage, we will have the required result to compare the weight of any tree with that of the complete tree.

Consider the $\overline{\text{CTW}}$ estimate of the $\ell^{\text{th}}$ bit for $\ell > 1$. Let $T \in \mathcal{T}_{\ell-1}$ be any context tree, and let $\boldsymbol{a} \in T$ with $\|\boldsymbol{a}\| < \ell - 1$. Let context tree $T'$ be obtained from $T$ by replacing $\boldsymbol{a}$ by its two children nodes $1\boldsymbol{a}$ and $0\boldsymbol{a}$ as in Fig. 3, i.e., $T' = (T \setminus \{\boldsymbol{a}\}) \cup \{0\boldsymbol{a}, 1\boldsymbol{a}\}$. In the first stage, by using (6), we see that the ratio of the sequence-dependent part of CTW weights $P_n(T, z_{1:n})$ and $P_n(T', z_{1:n})$ is

$$\frac{P_n^{(\ell)}(T, z_{1:n})}{P_n^{(\ell)}(T', z_{1:n})} = \frac{P_e(\#_\ell \boldsymbol{a}0, \#_\ell \boldsymbol{a}1)}{P_e(\#_\ell 0\boldsymbol{a}0, \#_\ell 0\boldsymbol{a}1) P_e(\#_\ell 1\boldsymbol{a}0, \#_\ell 1\boldsymbol{a}1)}.$$

Since $\#_\ell \boldsymbol{a}0 = \#_\ell 0\boldsymbol{a}0 + \#_\ell 1\boldsymbol{a}0$ and $\#_\ell \boldsymbol{a}1 = \#_\ell 0\boldsymbol{a}1 + \#_\ell 1\boldsymbol{a}1$, one needs to study the following quantity so as to compare the ratio of the weights of two trees. Let for $a, b, c, d \in \mathbb{N} \cup \{0\}$,

$$\eta_{a,b,c,d} := \frac{P_e(a+c, b+d)}{P_e(a,b) P_e(c,d)}. \quad (9)$$

Using Stirling's approximation, one can relate $\eta_{a,b,c,d}$ and the binary entropy function $h$ as follows. If $a+b > 0$ and $c+d > 0$,

$$\eta_{a,b,c,d} = \lambda_{abcd} \frac{e^{(a+b)h\left(\frac{a}{a+b}\right)+(c+d)h\left(\frac{c}{c+d}\right)}}{\sqrt{\frac{(a+b+c+d)}{2\pi(a+b)(c+d)}}\, e^{(a+b+c+d)h\left(\frac{a+c}{a+b+c+d}\right)}}, \quad (10)$$

for some $|\lambda_{abcd}| \leq e^{\frac{1}{2}}$, and $\eta_{a,b,c,d} = 1$ otherwise. A derivation of (10) is given in [13]. Observe that when $\frac{a}{b}$ and $\frac{c}{d}$ are quite different, the sequence-dependent weight for $T'$ can by larger than that of $T$ by an exponential factor, and thus, the context tree $T'$ *better* explains the data $z_{1:n}$ than tree $T$ does. When $\frac{a}{b}$ and $\frac{c}{d}$ are close, the exponential term does not play a significant role, and the sequence-dependent weight for $T$ is larger than that of $T'$ by a polynomial factor, and hence $T$ is a marginally better model for the data $z_{1:n}$ than $T'$.

Now, for the second stage, we can, by repeated application of the above result, compare the sequence-dependent part of the weights of two trees $T$, $T''$ related as follows. Let $T, T'' \in \mathcal{T}_{\ell-1}$ be two trees such that the former contains, say, a node $\boldsymbol{a}$, and the latter contains instead of $\boldsymbol{a}$, all nodes at depth $\ell - 1$ which have $\boldsymbol{a}$ as its suffix. Consider the example in Fig. 3, where $\boldsymbol{a} = 0$, and $T''$ contains $00\boldsymbol{a}$, $01\boldsymbol{a}$, $10\boldsymbol{a}$, and $11\boldsymbol{a}$ instead of $\boldsymbol{a}$. We can compare the sequence-dependent part of weights of $T$ and $T''$ by bounding three ratios of the sequence-dependent part of weights of: (a) trees $T$ and $T_a := \{00, 10, 01, 011, 111\}$; (b) trees $T_a$ and $T_b := \{000, 100, 10, 01, 011, 111\}$, and (c) trees $T_b$ and $T''$. More generally, in the $\overline{\text{CTW}}$ estimate of the $\ell^{\text{th}}$ bit for some $\ell = 2, \ldots, k$, the ratio of the sequence-dependent part of weights of trees $T$ and $T''$ (related in the above manner) is

$$\epsilon_{\boldsymbol{a}} := \frac{P_n^{(\ell)}(T, z_{1:n})}{P_n^{(\ell)}(T'', z_{1:n})} = \frac{\prod_{\boldsymbol{b}\in T} P_e(\#_\ell \boldsymbol{b}0, \#_\ell \boldsymbol{b}1)}{\prod_{\boldsymbol{b}\in T''} P_e(\#_\ell \boldsymbol{b}0, \#_\ell \boldsymbol{b}1)}, \quad (11)$$

which can be bounded as follows.

*Lemma 1:* Let $k \geq \ell \geq 2$ and $z_{1:n}$ be a sequence over $\{0,1\}^k$ and $\boldsymbol{a} \in \{0,1\}^{<\ell}$ such that $\#_{\ell-1}\boldsymbol{a} > 0$. Let $\epsilon_{\boldsymbol{a}}$ be as in (11) and $m := \ell - 1 - \|\boldsymbol{a}\|$. Then,

$$\epsilon_{\boldsymbol{a}} \leq \frac{(2\pi e)^{\frac{2^m-1}{2}} \frac{1}{\sqrt{\#_{\ell-1}\boldsymbol{a}}} \prod_{\boldsymbol{b}\in\{0,1\}^m : \#_{\ell-1}\boldsymbol{b}\boldsymbol{a}>0} \sqrt{\#_{\ell-1}\boldsymbol{b}\boldsymbol{a}}}{\exp\left\{4 \sum_{\substack{\boldsymbol{b}\in\{0,1\}^m \\ \#_{\ell-1}\boldsymbol{b}\boldsymbol{a}>0}} \#_{\ell-1}\boldsymbol{b}\boldsymbol{a} \left(\frac{\#_\ell \boldsymbol{b}\boldsymbol{a}0}{\#_{\ell-1}\boldsymbol{b}\boldsymbol{a}} - \frac{\#_\ell \boldsymbol{a}0}{\#_{\ell-1}\boldsymbol{a}}\right)^2\right\}}.$$

*Proof:* The proof proceeds by expressing $\epsilon_{\boldsymbol{a}}$ as a product of $\eta$-terms, and then exploiting the concavity of the binary entropy function. For a complete proof, see [13]. ∎

Finally, in the third stage, we can repeatedly use Lemma 1 to compare the sequence-dependent part of weights of any tree $T \in \mathcal{T}_{\ell-1}$ with that of the complete tree $T^*_{\ell-1} = \{0,1\}^{\ell-1}$.

*Lemma 2:* Let $k \geq \ell \geq 2$ and $z_{1:n}$ be a sequence over $\{0,1\}^k$ and $T \in \mathcal{T}_{\ell-1}$. Let $m_{\boldsymbol{a}} := \ell - 1 - \|\boldsymbol{a}\|$ for $\boldsymbol{a} \in \{0,1\}^{<\ell}$. Then, for some $\lambda_{\boldsymbol{a}}$ depending on $z_{1:n}$ such that $|\log \lambda_{\boldsymbol{a}}| \leq \frac{2^{m_{\boldsymbol{a}}}-1}{2}$, the ratio $\frac{P_n^{(\ell)}(T,z_{1:n})}{P_n^{(\ell)}(T^*_{\ell-1},z_{1:n})}$ is bounded above by

$$\frac{(2\pi)^{\frac{2^{\ell-1}-|T|}{2}} \left(\prod_{\substack{\boldsymbol{a}\in T \\ \#_{\ell-1}\boldsymbol{a}>0}} \frac{\lambda_{\boldsymbol{a}}}{\sqrt{\#_{\ell-1}\boldsymbol{a}}}\right) \prod_{\substack{\boldsymbol{c}\in\{0,1\}^{\ell-1} \\ \#_{\ell-1}\boldsymbol{c}>0}} \sqrt{\#_{\ell-1}\boldsymbol{c}}}{\exp\left\{4 \sum_{\boldsymbol{a}\in T} \sum_{\substack{\boldsymbol{b}\in\{0,1\}^{m_{\boldsymbol{a}}} \\ \#_{\ell-1}\boldsymbol{b}\boldsymbol{a}>0}} \#_{\ell-1}\boldsymbol{b}\boldsymbol{a} \left(\frac{\#_\ell \boldsymbol{b}\boldsymbol{a}0}{\#_{\ell-1}\boldsymbol{b}\boldsymbol{a}} - \frac{\#_\ell \boldsymbol{a}0}{\#_{\ell-1}\boldsymbol{a}}\right)^2\right\}}. \quad (12)$$

*Proof:* For a complete proof, see [13]. ∎

Having proven the required machinery, we are now equipped to state our results pertaining to the $\overline{\text{CTW}}$ estimator.

## V. New Results

We first present an upper bound for the worst-case $L_1$-error between the $\overline{\text{CTW}}$ and frequency estimates, followed by a lower bound established by a suitable choice for $z_{1:n}$.

*Theorem 1 (Upper Bound):* Let $\mathcal{Z} = \{0,1\}^k$, $k \in \mathbb{N}$. Then,

$$\Delta_k := \max_{z_{1:n} \in \mathcal{Z}^n} \sum_{\boldsymbol{c} \in \{0,1\}^k} \left| \hat{P}_{\overline{\text{CTW}}}(Z = \boldsymbol{c} \,|\, ; z_{1:n}) - \frac{\#_k \boldsymbol{c}}{n} \right| \leq \begin{cases} \frac{1}{2n} & k = 1 \\ \Delta_{k-1} + \frac{2^{k-1}}{2n} + \sqrt{\frac{2^{k-2}}{n} \log\left(\frac{2\pi e^5 n}{2^{k-1}}\right)} & k > 1 \end{cases}.$$

*Proof:* The proof proceeds by induction, and uses (3)-(7) to relate the worst-case $L_1$-error $\Delta_k$ for sequences over $\{0,1\}^k$ to worst-case $L_1$-error $\Delta_{k-1}$ for sequences over $\{0,1\}^{k-1}$. The proof is then complete by using (12) and some standard information-theoretic bounds. For a full proof, see [13]. ∎

Since Theorem 1 quantifies a worst-case bound that holds for *all* sequences, and since the frequency estimator is a consistent estimator of the stationary distribution of any finite-alphabet stationary ergodic source, the following two inferences ensue.

*Corollary 1:*

$$\Delta_k \leq \frac{2^{k-1}-1}{2n} + \sqrt{\frac{\log n}{n}} \left( \frac{\sqrt{2^k} - \sqrt{2}}{\sqrt{2} - 1} \right) (1 + o(1)), \; k \in \mathbb{N}.$$

*Corollary 2:* The binarized context-tree weighting estimator $\overline{\text{CTW}}$ is a consistent estimator of the stationary distribution of any finite-alphabet stationary ergodic source.

At this point, one might suspect the tightness of the worst-case bound, especially since for i.i.d. and finite-order ergodic Markov sources one can show using, say, Hoeffding's inequality that the expected $L_1$-error between binarized CTW (or the frequency) estimator and the actual stationary distribution of the source decays as $n^{-\frac{1}{2}}$. Hence, for these sources,

$$\mathbb{E} \left| \hat{P}_{\overline{\text{CTW}}}(Z = \boldsymbol{c} \,|\, ; z_{1:n}) - \frac{\#_k \boldsymbol{c}}{n} \right| = \Theta(n^{-\frac{1}{2}}). \tag{13}$$

However, the following result identifies sequences $z_{1:n}$ for which the $L_1$-error between the $\overline{\text{CTW}}$ and frequency estimates is of the same order as the upper bound of Theorem 1.

*Theorem 2 (Lower Bound):* Let $\mathcal{Z} = \{0,1\}^k$ for $k \geq 2$. Then, for $\epsilon > 0$, there exist $n \in \mathbb{N}$ and $z_{1:n} \in \mathcal{Z}^n$ such that

$$\sum_{\boldsymbol{c} \in \{0,1\}^k} \left| \hat{P}_{\overline{\text{CTW}}}(Z = \boldsymbol{c} \,|\, ; z_{1:n}) - \frac{\#_k \boldsymbol{c}}{n} \right| \geq \sqrt{\frac{2^{k-2}(1-\epsilon)\log n}{n}}.$$

*Proof:* The proof establishes that the above bound using a sequence of length $2^k n$ over $\{0,1\}^k$ with $n - \lfloor \sigma \sqrt{n \log n} \rfloor$ occurrences of each $\boldsymbol{b} \in \{0,1\}^k$ of even weight, and $n + \lfloor \sigma \sqrt{n \log n} \rfloor$ occurrences of each $\boldsymbol{b} \in \{0,1\}^k$ of odd weight, where $\sigma = \frac{1}{2}\sqrt{1-\alpha}$ and $\alpha \in (0,1)$. For details, see [13]. ∎

### A. Extensions of $\overline{CTW}$.

The formulation of the binarized CTW and the bounds in Theorems 1 and 2 assume no prior dependency structure between the bits $B_1, \ldots, B_k$ in the binary representation of each symbol of the random process under study. Any known dependency structure between the bits representable by a Bayesian network $\mathcal{B}$ can be incorporated to define an appropriate binarized CTW estimate $\hat{P}^{\mathcal{B}}_{\overline{\text{CTW}}}(\cdot \,|\, ; z_{1:n})$ by modifying (7). In general, the binarized CTW variant $\hat{P}^{\mathcal{B}}_{\overline{\text{CTW}}}$ will be a Bayesian mixture over all tree sources that satisfy the dependencies imposed by $\mathcal{B}$, and will yield some computational benefits over $\hat{P}_{\overline{\text{CTW}}}$. Similar to Theorems 1 and 2, we can establish the following result quantifying the $L_1$-prediction error between $\hat{P}^{\mathcal{B}}_{\overline{\text{CTW}}}$ and the maximum-likelihood estimate

$$\hat{P}_{\mathsf{ML},\mathcal{B}}(\cdot \,; z_{1:n}) := \text{argmax}_{P \in \mathcal{P}(\mathcal{B})} P(z_{1:n}), \tag{14}$$

where $\mathcal{P}(\mathcal{B})$ is the set of all probability distributions that satisfy the Markov/dependency conditions of the Bayesian network $\mathcal{B}$. When $\mathcal{B}$ imposes no Markov conditions among $B_1, \ldots, B_k$, $\hat{P}_{\mathsf{ML},\mathcal{B}}(\cdot \,; z_{1:n})$ is simply the frequency estimate.

*Theorem 3:* Given Bayesian network $\mathcal{B}$ consisting of $k$ binary random variables, and $\mathcal{Z} = \{0,1\}^k$,

$$\max_{z_{1:n} \in \mathcal{Z}^n} \left\| \hat{P}^{\mathcal{B}}_{\overline{\text{CTW}}}(\boldsymbol{c} \,|\, ; z_{1:n}) - \hat{P}_{\mathsf{ML},\mathcal{B}}(\boldsymbol{c} \,; z_{1:n}) \right\| = \Theta\left( \sqrt{\frac{\log n}{n}} \right).$$

## References


[1] F. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "Context tree weighting: a sequential universal source coding procedure for FSMX sources," *Proc. IEEE International Symposium on Information Theory*, p. 59, Jan 1993.

[2] ——, "The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[3] R. Krichevsky and V. Trofimov, "The performance of universal encoding," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199–207, Mar 1981.

[4] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Transactions on Information Theory*, vol. 30, no. 4, pp. 629–636, Jul 1984.

[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[6] ——, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, no. 5, pp. 530–536, Sept 1978.

[7] T. J. Tjalkens, Y. M. Shtarkov, and F. Willems, "Context tree weighting: Multi-alphabet sources," *Proc. 14th Symposium on Information Theory,* Benelux, pp. 128–135, 1993.

[8] T. J. Tjalkens, F. Willems, and Y. M. Shtarkov, "Sequential weighting algorithms for multi-alphabet sources," *Proc. 6th Swedish-Russian Workshop on Information Theory,* Mölle, Sweden, pp. 22–27, Aug 1993.

[9] ——, "Multi-alphabet universal coding using a binary decomposition context tree weighting algorithm," *Proc. 15th Symposium on Information Theory,* Benelux, pp. 259–265, 1994.

[10] T. J. Tjalkens, P. A. J. Volf, and F. Willems, "A context-tree weighting method for text generating sources," *Proc. Data Compression Conference (DCC '97)*, p. 472, March 1997.

[11] R. Begleiter and R. El-Yaniv, "Superior guarantees for sequential prediction and lossless compression via alphabet decomposition," *J. Mach. Learn. Res.*, vol. 7, pp. 379–411, Dec. 2006.

[12] J. Veness, M. G. Bellemare, M. Hutter, A. Chua, and G. Desjardins, "Compress and control," *Proc. Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15)*, pp. 3016–3023, 2015.

[13] B. N. Vellambi and M. Hutter, "Convergence of binarized context-tree weighting for estimating distributions of stationary sources." Online: http://www.hutter1.net/publ/convbinctwx.pdf.