An Algorithm for Universal Lossless Compression With Side Information

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

***Abstract*—This paper proposes a new algorithm based on the Context-Tree Weighting (CTW) method for universal compression of a finite-alphabet sequence x_1^n with side information y_1^n available to both the encoder and decoder. We prove that with probability one the compression ratio converges to the conditional entropy rate for jointly stationary ergodic sources. Experimental results with Markov chains and English texts show the effectiveness of the algorithm.**

***Index Terms*—Arithmetic coding, conditional entropy, context tree weighting method, hidden Markov process, source coding, universal lossless data compression.**

I. Introduction

In this paper, we study the problem of universal lossless compression with side information. That is, we wish to encode the sequence x_1^n where both the encoder and decoder know a side information sequence y_1^n. Assuming the two sources are jointly stationary and ergodic, we would like to encode x_1^n at a compression rate equal asymptotically to the conditional entropy rate $H(X|Y)$, which is the fundamental limit that follows by straightforward extension of the Shannon–McMillan theorem. Notice that both the encoder and decoder know the side information y_1^n, but neither know anything about the joint or individual distributions of X and Y.

In several applications, side information known to both the encoder and decoder is available. For example, when two remote users $\mathbf{A}$ and $\mathbf{B}$ have identical copies of a file and $\mathbf{A}$ wants to convey an edited version of the file to $\mathbf{B}$, the side information is the original file. Universal compression with side information is also useful in data exchange protocols (see [4]). For example, in Algorithm B proposed in [4], there are three stages in data exchange between two users. In the first stage, a noisy version of y_1^n is transferred. In the second stage, further communication between the two parties is needed to ensure that an exact copy of y_1^n is decoded. Once both parties have an exact copy of y_1^n, in the third stage, x_1^n can be encoded at a rate slightly higher than $H(X|Y)$. The compression algorithm with side information can be used in the third stage to complete the data exchange process. Other applications include multiresolution image coding where one may use low-resolution images as side information for high-resolution images [11], and lossless compression of video [3] where previous frames are used as the side information.

Zero-error encoding for memoryless sources with side information at the decoder only was initially studied in [17]. In almost lossless compression, the celebrated Slepian–Wolf–Cover result [10], [5] shows that side information at the encoder does not decrease the asymptotic minimal compression rate. In contrast, when strictly lossless compression is required, the conditional entropy is not achievable if the side information is not available at the encoder [1], [8], [9].

Universal compression with side information known to both the encoder and decoder has been studied in [12], where the following "conditional" version of the Lempel–Ziv algorithm was proposed.

1) Fix a window size n_w and transmit the first n_w symbols $x_1^{n_w}$ without compression.
2) Parse the sequence of joint symbols (x, y) by the sliding-window Lempel–Ziv algorithm. Let L_i be the largest integer such that a copy of $(x,y)_{n_i+1}^{n_i+L_i-1}$ occurs in the current window $(x,y)_{n_i-n_w+1}^{n_i}$. Let the copy begin at position *start* Define the new window to be $(x,y)_{n_i+L_i-n_w+1}^{n_i+L_i}$.
3) Represent the ith phrase $x_{n_i+1}^{n_i+L_i}$ which consists of the matched portion $x_{n_i+1}^{n_i+L_i-1}$ and the last symbol $x_{n_i+L_i}$. The length of the phrase can be represented using $\lceil \log L_i \rceil + 2\lceil \log \log L_i \rceil$ bits. The starting point of the match can be specified using $\lceil \log N_i \rceil + 2\lceil \log \log N_i \rceil$ bits, where

$$N_i = \left| \left\{ k : start \le k < n_i, y_{k+1}^{k+L_i-1} = y_{n_i+1}^{n_i+L_i-1} \right\} \right|.$$

4) Repeat steps 2) and 3) as necessary until the sequence is exhausted.

A conditional multilevel pattern matching (CMPM) grammar-based code was proposed in [18].[1] It was proved that the worst case redundancy per sample is upper-bounded by $O(1/\log n)$. The MPM code transforms the data sequence into a grammar, which is then compressed by the zero-order adaptive arithmetic code. The MPM grammar with parameter (r, I) is as follows. At top level I, the sequence is partitioned into blocks of length r^I. From left to right, each block is labeled either by "s" if it is the first appearance or by an integer pointing to its first appearance. Then the following steps are performed at subsequent levels. At level $i (1 < i < I)$.

1) Each block labeled by "s" in the previous level is partitioned into r subblocks of length r^i. Then concatenate all blocks of length r^i.

[1]Recently, the related problem of universal refinement source coding was studied in [7] using refinement of grammars.

Manuscript received May 10, 2005; revised May 9, 2006. This work was supported in part by ARL MURI under Grant DAAD19-00-1-0466, Draper Laboratory under IR&D 6002 Grant DL-H-546263, and the National Science Foundation under Grant CCR-0312413.

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

Communicated by S. A. Savari, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2006.880020

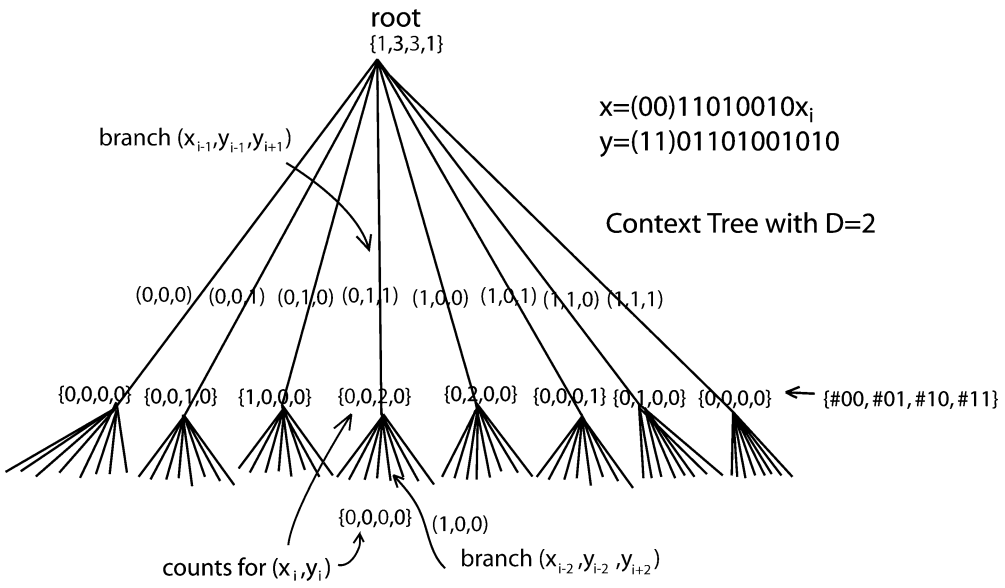

Fig. 1. Context tree with $D = 2$ of joint source (x, y). We encode x_i, where $i = 9$. A branch from depth $d - 1$ to depth d corresponds to $(x_{i-d}, y_{i-d}, y_{i+d})$, $d = 1, 2$.

2) From left to right, every first-appearing distinct block is labeled by "s." Subsequent appearances are labeled by an integer pointing to its first appearance.

The conditional MPM is performed on (x, y). At level i we get the following.

1) Each block labeled by "s" in the previous level is partitioned into r subblocks of length r^i. Then concatenate all blocks of length r^i.
2) Visit every $\mathcal{Y}$-block from left to right, and label all identical $\mathcal{Y}$-blocks with the same integer and all distinct $\mathcal{Y}$-blocks with distinct integers with increasing order, starting with 1.
3) For each distinct $\mathcal{Y}$-block γ, visit every $\mathcal{X}$-block corresponding to γ, and label the first appearance of each distinct $\mathcal{X}$-block by "s" and subsequent appearances by the integer pointing to the first appearance.

The CMPM grammar is then encoded by a conditional arithmetic coder.

In this paper, we propose a compression algorithm with side information known to both encoder and decoder based on the Context Tree Weighting (CTW) principle [14], [16]. The details of the algorithm are presented in Section II. In Section III, we show that for jointly stationary ergodic sources, the compression rate achieved by the algorithm converges to the conditional entropy rate. Implementation issues (particularly for sources with large alphabets) are discussed in Section IV. Finally, experimental results on randomly generated sources and on English text files are presented in Section V.

II. Algorithm

The CTW method updates a context tree and uses a weighting scheme to calculate a weighted probability, which is a mixture of estimated probabilities assuming different models. The weighted probability at the root of the context tree is the coding probability fed to the arithmetic coder. The context of the current symbol is a suffix of the past symbols. In a context tree, the path from any node to the root corresponds to a context, with the most recent past symbol represented by the branch closest to the root. An important notion in our algorithm is that in coding the ith symbol x_i, the concept of context is extended to include both the past observations $(x, y)_1^{i-1}$ and the future symbols y_i^n. In the following, we discuss in detail how to build the context tree and calculate the coding probability in our algorithm.

The context tree uses the joint symbol $(x, y) \in \mathcal{X} \times \mathcal{Y}$ (see Fig. 1). In the context tree with maximum order D, each node stores the counts of symbol (x, y) in the corresponding context, as well as the estimated probability P_e and the weighted probability P_w. The context here includes both past symbols of (x, y) and future symbols of y. If the current symbol is (x_i, y_i), in order to find the dth-order context, we have to take branches according to $(x_{i-k}, y_{i-k}, y_{i+k})$ for $1 \le k \le d$. Thus, the path from a node at depth d to the root corresponds to the context $x_{i-d}^{i-1} y_{i-d}^{i-1} y_{i+1}^{i+d}$. Therefore, each node stores $|\mathcal{X}||\mathcal{Y}|$ counts and has $|\mathcal{X}||\mathcal{Y}|^2$ branches.

The Basic Algorithm (conditioning on the past symbols $(x, y)_{i-D}^{i-1}$, the current symbol y_i and the future symbols y_{i+1}^{i+D}):

For the current symbol $(x_i, y_i), i = 1, 2, \ldots, n$, we perform the following steps.

1) Travel through the context tree according to $(x_{i-k}, y_{i-k}, y_{i+k}), k = 1, 2, \ldots, D$, until a leaf node is reached. Notice that both encoder and decoder have access to past symbols $(x, y)_{i-D}^{i-1}$ and future symbols y_{i+1}^{i+D}, as long as past symbols are decoded correctly. (In the basic algorithm, both encoder and decoder are assumed to know $(x, y)_{-D+1}^{0}$ and y_{n+1}^{n+D}. This assumption is removed in the extended algorithm discussed later.)
2) Travel back from the leaf to the root. In each node s in the updating path, select an $|\mathcal{X}|$-vector of counts (x, y_i) where

$x \in \mathcal{X}$, and calculate the conditional probability for x_i. The estimated probability P_e^s of node s is updated

$$P_e^s := \frac{c_s(x_i, y_i) + \frac{1}{2}}{\sum_{x \in \mathcal{X}} c_s(x, y_i) + \frac{1}{2}|\mathcal{X}|} \cdot P_e^s \tag{1}$$

The count of (x_i, y_i) in node s is updated

$$c_s(x_i, y_i) := c_s(x_i, y_i) + 1. \tag{2}$$

Then the weighted probability P_w^s of node s is updated

$$P_w^s := \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2}\prod_{v \in \mathrm{Child}(s)} P_w^v, & 0 \le l(s) < D \\ P_e^s, & l(s) = D \end{cases} \tag{3}$$

where $l(s)$ is the depth of node s, and $\mathrm{Child}(s)$ is the set of children nodes of s.

3) Once the weighted probability at the root is obtained, it is fed to the arithmetic coder to encode x_i.

Notice both encoder and decoder have access to y_i, so the decoder can follow the same steps and recover x_i. For the example shown in Fig. 1, the leaf node has counts $\{0,0,0,0\}$. Since $y_i = 0$, we should select the first and third counts $\{0,0\}$. (In Fig. 1, the selected counts in the updating path are highlighted.) The counts $(0,0)$ translate to probability $(\frac{1}{2}, \frac{1}{2})$, which are the statistics for the symbol x_i at the leaf node. The internal node has counts $\{0,0,2,0\}$. Since $y_i = 0$, we should select the counts $\{0,2\}$. The root node has counts $\{1,3,3,1\}$. Since $y_i = 0$ we should select the counts $\{1,3\}$.

The extended CTW method [16] has unbounded memory length and achieves asymptotic optimality for all stationary ergodic sources. Our conditional compression algorithm can also be extended in the same way. Note that it is unnecessary to maintain further children nodes of a *unique* node, which corresponds to a context that has occurred only once so far.

The Extended Algorithm (conditioning on the past symbols $(x,y)_1^{i-1}$, the current symbol y_i and the future symbols y_{i+1}^n):

For the current symbol $(x_i, y_i), i = 1, 2, \ldots, n$, we perform the following steps.

1) Travel through the extended context tree according to $(x_{i-k}, y_{i-k}, y_{i+k}), k = 1, 2, \ldots$ until a *null* node is encountered, which corresponds to a context that has never occurred so far. New nodes are added to the context tree during this step. The unknown past $(x,y)_{-\infty}^0$ and unknown future y_{n+1}^{∞} are padded with symbol ϵ. This null node becomes a unique node since the current context now occurs for the first time.
2) Travel back to the root. In each node s in the updating path, select an $|\mathcal{X}|$-vector of counts (x, y_i) where $x \in \mathcal{X}$, and calculate the conditional probability for x_i. The estimated probability P_e^s of node s is updated

$$P_e^s := \frac{c_s(x_i, y_i) + \frac{1}{2}}{\sum_{x \in \mathcal{X}} c_s(x, y_i) + \frac{1}{2}|\mathcal{X}|} \cdot P_e^s \tag{4}$$

The count of (x_i, y_i) in node s is updated

$$c_s(x_i, y_i) := c_s(x_i, y_i) + 1. \tag{5}$$

Then the weighted probability $\tilde{P}_w^s$ of node s is updated

$$\tilde{P}_w^s := \begin{cases} \frac{1}{|\mathcal{X}|}, & \text{if } s \text{ is unique} \\ \frac{1}{2}P_e^s + \frac{1}{2}\prod_{v \in \mathrm{Child}'(s)} \tilde{P}_w^v, & \text{otherwise} \end{cases} \tag{6}$$

where $\mathrm{Child}'(s)$ is the set of children nodes of s in the extended context tree. Note that there are two special nodes in $\mathrm{Child}'(\mathrm{s})$ symbolized by ϵ, which represent the unknown past $(x,y)_{-\infty}^0$ and unknown future y_{n+1}^{∞}, respectively. If a context occurs in the beginning or in the end of $(x,y)_1^n$, then its children nodes include the special node(s), whose estimated/weighted probabilities are simply $1/|\mathcal{X}|$.

3) The weighted probability at the root is fed to the arithmetic coder to encode x_i.

III. Analysis

In this section, we give our main results on the optimality of the compression algorithms with side information proposed in this paper. Following the technique in [14], Theorem 1 provides an upper bound on the compression ratio using the basic CTW method with maximal memory length D. Theorem 2 asserts asymptotic optimality of the algorithm using the extended CTW method.

Theorem 1: For jointly stationary and ergodic (X, Y), using the conditional CTW with a maximum memory length D, we have

$$\limsup_{n \to \infty} \frac{L\left(x_1^n | y_1^n\right)}{n} \le H\left(X_i | Y_i, X_{i-D}^{i-1}, Y_{i-D}^{i-1}, Y_{i+1}^{i+D}\right) \tag{7}$$

almost surely, where $L(x_1^n | y_1^n)$ is the code length to compress sequence x_1^n with side information y_1^n.

Proof: Let $P_c(x_1^n | x_{-D+1}^0, y_{-D+1}^{n+D})$ be the coding probability of x_1^n with side information y_1^n. (We assume both encoder and decoder also know $(x,y)_{-D+1}^0$ and y_{n+1}^{n+D}.) Let

$$P\left(x_1^n | x_{-D+1}^0, y_{-D+1}^{n+D}, d\right) = \prod_{i=1}^{n} P\left(x_i | y_i, x_{i-d}^{i-1}, y_{i-d}^{i-1}, y_{i+1}^{i+d}\right)$$

where $P(\cdot|\cdot)$ is the actual conditional probability.

Let $P_e(\cdot)$ be a function of an $|\mathcal{X}|$-dimensional vector defined as follows: $P_e(0, 0, \ldots, 0) = 1$ and

$$P_e(c_1, c_2, \ldots, c_l + 1, \ldots, c_{|\mathcal{X}|}) = \frac{c_l + \frac{1}{2}}{\mathcal{C} + \frac{1}{2}|\mathcal{X}|} P_e(c_1, c_2, \ldots, c_l, \ldots, c_{|\mathcal{X}|}) \tag{8}$$

where

$$\mathcal{C} = \sum_{j=1}^{|\mathcal{X}|} c_j. \tag{9}$$

For any $1 \le d \le D$

$$\log \frac{1}{P_c\left(x_1^n | x_{-D+1}^0, y_{-D+1}^{n+D}\right)} - \log \frac{1}{P\left(x_1^n | x_{-D+1}^0, y_{-D+1}^{n+D}, d\right)}$$

$$= \log \frac{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))}{P_c\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}\right)} + \log \frac{P\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}, d\right)}{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))} \tag{10}$$

where $\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))$ is the estimated probability of node s, $c_s(\cdot,\cdot)$ are the counts stored in node s, and $c_s(\cdot,y)$ is a vector of $|\mathcal{X}|$ integers.

The second term in (10) can be bounded by

$$\begin{aligned}
&\log \frac{P\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}, d\right)}{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))} \\
&= \sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}} \log \frac{\prod_{x\in\mathcal{X}} p(x|y,s)^{c_s(x,y)}}{P_e(c_s(\cdot,y))} \\
&\le \sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}:\sum_{x\in\mathcal{X}} c_s(x,y)>0} \\
&\quad\times \left(\frac{|\mathcal{X}|-1}{2}\log\left(\sum_{x\in\mathcal{X}} c_s(x,y)\right)+1\right) \qquad (11)\\
&\le |\mathcal{Y}|^{2d+1}|\mathcal{X}|^d(|\mathcal{X}|-1) \\
&\quad\times \sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}} |\mathcal{Y}|^{-(2d+1)}|\mathcal{X}|^{-d}\gamma\left(\sum_{x\in\mathcal{X}} c_s(x,y)\right) \\
&\le |\mathcal{Y}|^{2d+1}|\mathcal{X}|^d(|\mathcal{X}|-1)\gamma\left(|\mathcal{Y}|^{-(2d+1)}|\mathcal{X}|^{-d}n\right) \qquad (12)
\end{aligned}$$

where

$$\gamma(z) = \begin{cases} z & : \ 0\le z\le 1 \\ \frac{1}{2}\log z + 1 & : \ z>1. \end{cases} \tag{13}$$

The first inequality (11) follows from of [14, eq. (11)] and

$$P_e(c_1, c_2, \ldots, c_{|\mathcal{X}|}) \ge \frac{1}{2}\frac{1}{\mathcal{C}^{(|\mathcal{X}|-1)/2}}\prod_{1\le i\le|\mathcal{X}|}\left(\frac{c_i}{\mathcal{C}}\right)^{c_i} \tag{14}$$

where $\mathcal{C}\ge 1$ and $c_i\ge 0$, for $1\le i\le|\mathcal{X}|$, which is proved in Appendix A. The inequality (12) follows from the fact that function $\gamma(\cdot)$ is convex.

The first term in (10) can be bounded by

$$\log \frac{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))}{P_c(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D})} \le \frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1}. \tag{15}$$

To see this, by the recursive weighing formula (3), we have

$$\log \frac{\prod_{s\in\mathcal{S}}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))}{P_c\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}\right)} \le \frac{r|\mathcal{S}|-1}{r-1} \tag{16}$$

where $|\mathcal{S}|$ is the number of states (leaf nodes) of the tree source S, and r is the number of children of any internal node. The right-hand side of (16) is the number of nodes (including internal nodes and leaves) of S. To obtain (15), we just let $r=|\mathcal{Y}|^2|\mathcal{X}|$ and $|\mathcal{S}|=(|\mathcal{Y}|^2|\mathcal{X}|)^d$.

By using the arithmetic coder, the codeword length $L(x_1^n|y_1^n)$ divided by the sequence length n is upper-bounded by

$$\frac{L\left(x_1^n|y_1^n\right)}{n} \le \frac{1}{n}\log\frac{1}{P_c\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}\right)} + \frac{2}{n}. \tag{17}$$

By (10), (12), (15), and (17) we have

$$\begin{aligned}
\frac{L\left(x_1^n|y_1^n\right)}{n} &\le \frac{1}{n}\log\frac{1}{P\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}, d\right)} \\
&\quad + \frac{1}{n}\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1} \\
&\quad + \frac{|\mathcal{Y}|^{2d+1}|\mathcal{X}|^d(|\mathcal{X}|-1)}{n} \\
&\quad \cdot\gamma(|\mathcal{Y}|^{-(2d+1)}|\mathcal{X}|^{-d}n) + \frac{2}{n}. \qquad (18)
\end{aligned}$$

Since

$$\begin{aligned}
&\frac{1}{n}\log\frac{1}{P\left(x_1^n|x_{-D+1}^0, y_{-D+1}^{n+D}, d\right)} \\
&\qquad \to H\left(X_i|Y_i, X_{i-d}^{i-1}, Y_{i-d}^{i-1}, Y_{i+1}^{i+d}\right) \qquad (19)
\end{aligned}$$

almost surely, we have

$$\limsup_{n\to\infty}\frac{L\left(x_1^n|y_1^n\right)}{n} \le H\left(X_i|Y_i, X_{i-d}^{i-1}, Y_{i-d}^{i-1}, Y_{i+1}^{i+d}\right) \tag{20}$$

almost surely, for any $1\le d\le D$. □

Following the analysis in [16], we now give a result on the conditional compressor based on extended CTW which eliminates the restriction on the maximal memory length.

Theorem 2: For jointly stationary and ergodic (X,Y), using the conditional extended CTW with unbounded memory length, we have

$$\limsup_{n\to\infty} L\left(x_1^n|y_1^n\right)n \le H(X|Y) \quad \text{a.s.} \tag{21}$$

where $L(x_1^n|y_1^n)$ is the code length to compress sequence x_1^n with side information y_1^n.

Proof: For any $d\ge 1$, due to the weighting formula (6), we have

$$\begin{aligned}
\tilde{P}_c(x_1^n|y_1^n) = \tilde{P}_w^{\lambda} &\ge 2^{-\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1}}|\mathcal{X}|^{-\Delta_d(x_1^n|y_1^n)} \\
&\quad\cdot\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y)) \qquad (22)
\end{aligned}$$

where $\tilde{P}_c(x_1^n|y_1^n)$ is the coding probability of x_1^n with side information y_1^n, and $\Delta_d(x_1^n|y_1^n)=2d$ is the number of symbols in x_1^n for which the full context is not available, including the first d symbols and the last d symbols. We have

$$\begin{aligned}
&\frac{1}{n}\log\frac{1}{\tilde{P}_c\left(x_1^n|y_1^n\right)} \\
&\quad\le \frac{1}{n}\log\frac{1}{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))} \\
&\qquad + \frac{\Delta_d(x_1^n|y_1^n)\log|\mathcal{X}|}{n} + \frac{1}{n}\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1} \\
&\quad= \frac{1}{n}\log\frac{1}{\prod_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\prod_{y\in\mathcal{Y}} P_e(c_s(\cdot,y))} \\
&\qquad + \frac{2d\log|\mathcal{X}|}{n} + \frac{1}{n}\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1}
\end{aligned}$$

$$\le -\frac{1}{n}\sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} c_s(x,y)\log\frac{c_s(x,y)}{\sum_{x'}c_s(x',y)}$$
$$+\frac{1}{n}\sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}:\sum_{x\in\mathcal{X}}c_s(x,y)>0}$$
$$\times\left(\frac{|\mathcal{X}|-1}{2}\log\left(\sum_{x\in\mathcal{X}}c_s(x,y)\right)+1\right)$$
$$+\frac{2d\log|\mathcal{X}|}{n}+\frac{1}{n}\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1}$$
$$\le -\frac{1}{n}\sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} c_s(x,y)\log\frac{c_s(x,y)}{\sum_{x'}c_s(x',y)} \quad (23)$$
$$+\frac{|\mathcal{Y}|^{2d+1}|\mathcal{X}|^d(|\mathcal{X}|-1)}{n}\cdot\gamma(|\mathcal{Y}|^{-(2d+1)}|\mathcal{X}|^{-d}n)$$
$$+\frac{2d\log|\mathcal{X}|}{n}+\frac{1}{n}\frac{(|\mathcal{Y}|^2|\mathcal{X}|)^{d+1}-1}{|\mathcal{Y}|^2|\mathcal{X}|-1}. \quad (24)$$

The first inequality follows from (22), the second inequality follows from (14), and the third inequality follows from the fact that the function $\gamma(\cdot)$ is convex. For jointly stationary ergodic sources, we have

$$\lim_{n\to\infty}-\frac{1}{n}\sum_{s\in\mathcal{Y}^{2d}\times\mathcal{X}^d}\sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} c_s(x,y)\log\frac{c_s(x,y)}{\sum_{x'}c_s(x',y)}$$
$$=H(X_i|Y_i,X_{i-d}^{i-1},Y_{i-d}^{i-1},Y_{i+1}^{i+d}) \quad \text{a.s.} \quad (25)$$

By using the arithmetic coder, we have

$$\frac{L(x_1^n|y_1^n)}{n}\le\frac{1}{n}\log\frac{1}{\tilde{P}_c(x_1^n|y_1^n)}+\frac{2}{n}. \quad (26)$$

Therefore,

$$\limsup_{n\to\infty}\frac{L(x_1^n|y_1^n)}{n}\le H\left(X_i|Y_i,X_{i-d}^{i-1},Y_{i-d}^{i-1},Y_{i+1}^{i+d}\right) \quad (27)$$

almost surely, for any $d\ge 1$. Since

$$\lim_{d\to\infty}H(X_i|Y_i,X_{i-d}^{i-1},Y_{i-d}^{i-1},Y_{i+1}^{i+d})=H(X|Y) \quad (28)$$

(see Appendix B), we have that the limsup of the conditional compression rate is upper-bounded by the conditional entropy rate

$$\limsup_{n\to\infty}\frac{L(x_1^n|y_1^n)}{n}\le H(X|Y) \quad \text{a.s.} \quad (29)$$

□

Notice that even if (X,Y) forms a finite-order Markov chain, X_i still depends on an infinite number of future symbols Y_i^∞, and the upper bound $H(X_i|Y_i,X_{i-D}^{i-1},Y_{i-D}^{i-1},Y_{i+1}^{i+D})$ is larger than $H(X|Y)$. In practice, the basic CTW method with finite memory length performs almost as well as the extended CTW method.

IV. Implementation

For sources with large alphabets, the number of links (to children nodes) and the number of counts stored in each node are very large. (Assuming an alphabet size of 27, the number of links stored in a node is $27^3=19683$ and the number of counts stored in a node is $27^2=729$.) We can dynamically allocate space for nodes, links, and counts, but it still takes a large amount of memory to build the context tree even with a moderate memory length D. In practice, the CTW approach may exhibit poor performance if the alphabet size is too large [2], [13]. In fact, the redundancy bounds in (18) and (24) will be too large and practically useless.

There are several techniques discussed in [6], [15] to improve the CTW method for sources with large alphabets, which can also be used in the implementation of the conditional CTW algorithm.

1) Since the CTW method works best for binary sources, it is appealing to use a multilevel approach where we decompose symbols into bits, with separate context trees for each bit of the symbol. The context of each bit consists of all earlier bits of the current symbol as well as all earlier symbols. For the multilevel CTW [15], the root of the context tree for the ith bit has 2^{i-1} branches, while the number of branches of an internal node equals the alphabet size. The counts of 0's and 1's are stored in each node. Weighting takes place at internal nodes, which are symbol boundaries. For the multilevel conditional CTW, we build a context tree for each bit of the symbol X and each different symbol of Y (so there are totally $|\mathcal{Y}|\lceil\log_2|\mathcal{X}|\rceil$ context trees), and the number of branches of an internal node equals $|\mathcal{X}||\mathcal{Y}|^2$ (see Fig. 2).
2) Hashing can be used to reduce the required memory and save space for pointers to children nodes.
3) The zero-redundancy estimator for binary sources

$$P_{e,ZR}(c_1,c_2):=\begin{cases}\frac{1}{2}P_e(c_1,c_2) & : \text{ for } c_1>0,c_2>0\\ \frac{1}{2}P_e(c_1,0)+\frac{1}{4} & : \text{ for } c_1>0,c_2=0\\ \frac{1}{2}P_e(0,c_2)+\frac{1}{4} & : \text{ for } c_1=0,c_2>0\\ 1 & : \text{ for } c_1=c_2=0\end{cases} \quad (30)$$

can be used to replace the Krichevski–Trofimov estimator in order to reduce the parameter redundancy for a source that generates 0's and 1's only.

The computational complexity of the basic algorithm in Fig. 1 is $O(nD|\mathcal{X}||\mathcal{Y}|(|\mathcal{X}|-1))$, because for each symbol, we have to update D tree nodes, and there are $|\mathcal{X}||\mathcal{Y}|$ counts in each tree node. The factor $(|\mathcal{X}|-1)$ is due to the arithmetic coding. The computational complexity of the improved algorithm in Fig. 2 is $O(nD\lceil\log_2|\mathcal{X}|\rceil)$, because for each symbol, we have to update $\lceil\log_2|\mathcal{X}|\rceil$ context trees, but each one stores counts of 0's and 1's only. The improved algorithm takes more space, because it essentially keeps $|\mathcal{X}||\mathcal{Y}|$ context trees and the number of pointers to children in each tree node is the leading term $(|\mathcal{X}||\mathcal{Y}|^2)$. However, if a hashing technique is used to store tree nodes, then we do not have to store pointers and the number of counts in each tree node is 2 instead of $|\mathcal{X}||\mathcal{Y}|$, in which case the improved algorithm does not use much more space. For the extended algorithm with unbounded memory, the tree depth D should be replaced by the sequence length n in the worst case in the computational complexity, and the space needed grows linearly with n.

In the experimental results in Section V, we find these techniques very useful in dealing with sources with large alphabets.

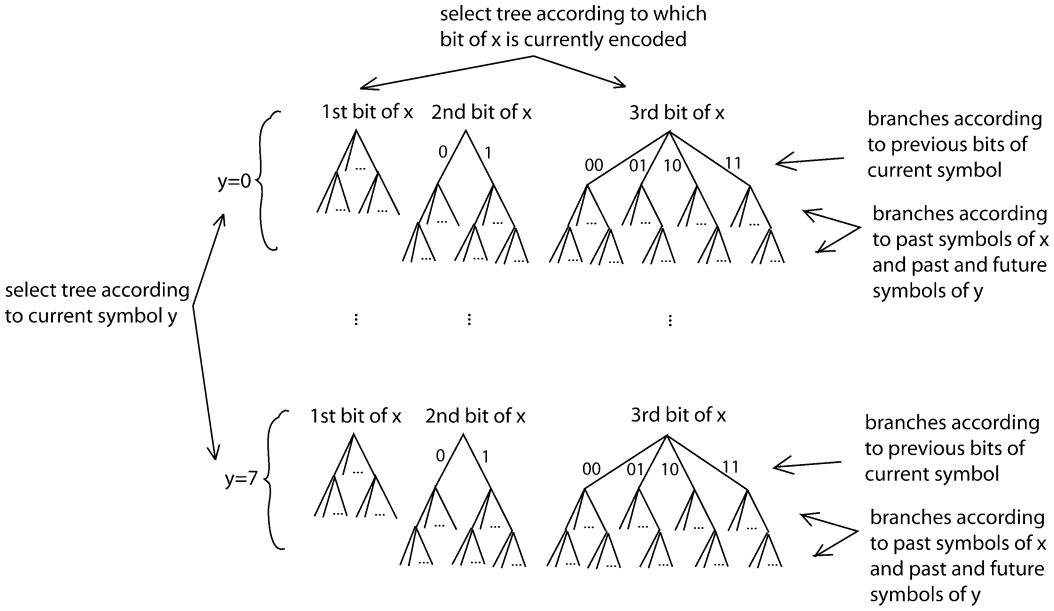

Fig. 2. Example of multilevel context tree. $|\mathcal{X}| = |\mathcal{Y}| = 8, D = 2$.

In the special case that we know Y is X observed through a discrete memoryless channel and want to compress X given Y, instead of conditioning on the past symbols of (X, Y), we condition only on the past symbols of X and future symbols of Y. In that case, the number of branches of an internal node is reduced to $|\mathcal{X}||\mathcal{Y}|$, since we condition on the past symbols of X and the current and future symbols of Y. We show in Section V that this modification also improves the compression ratio.

V. Simulations

Example 1: We test the same example as in [18], and compare our method with the algorithm therein. Y is a binary Markov chain with the transition matrix

$$\begin{bmatrix} 1-q & q \\ q & 1-q \end{bmatrix}$$

and we construct the hidden Markov chain $X_i = Y_i \bigoplus W_i$ where W_i is independent and identically distributed (i.i.d.) with the probability of symbol 0 being p. We have $H(X|Y) = H(W)$. When $p = 0.9$ and $q = 0.8, H(X|Y) = 0.469$. Fig. 3 shows the compression ratio as a function of data size of both our algorithm and the CMPM algorithm. The compression ratio can also be seen as an estimate of the conditional entropy rate.

Example 2: With the same processes used in Example 1, we now interchange the roles of X and Y, with X taking the role of side information. Note that $H(Y|X) = 0.3075 < H(X|Y) = h(p)$. In Fig. 4, we test the case when there is a lag between both sequences: "sync+k" means that we have advanced the sequence x by k positions.

Example 3: We test the algorithm on English texts. Let X be the original copy of a novel, and Y be a noisy version of the novel (X observed through a discrete memoryless channel). We use the algorithm to estimate both $H(Y|X)$ and $H(X|Y)$, assuming $Y = X + W$ where W is i.i.d. noise independent of X. In this case, it is easy to calculate $H(Y|X) = H(W)$. Since we do not know the statistics of X we do not know $H(X|Y)$, but expect it to be less than $H(Y|X)$. In our experiment, the corrupted symbol is either the original symbol with probability p, or corrupted to any other symbol with equal probabilities.

The novel we use is *Emma* by Jane Austen. Conditional compression ratios versus sequence lengths are shown in Fig. 5, where $p = 0.99$. We have found that other novels yield similar results (not shown here), and the estimate of $H(Y|X)$ does converge to the true value. In addition, the estimate of $H(X|Y)$ is consistently smaller than the estimate of $H(Y|X)$.

Appendix A

We prove that

$$P_e(c_1, c_2, \ldots, c_{|\mathcal{X}|}) \geq \frac{1}{2} \frac{1}{\mathcal{C}^{(|\mathcal{X}|-1)/2}} \prod_{1 \leq i \leq |\mathcal{X}|} \left(\frac{c_i}{\mathcal{C}}\right)^{c_i} \tag{31}$$

where $\mathcal{C} \geq 1$ and $c_i \geq 0$, for $1 \leq i \leq |\mathcal{X}|$ by extending the proof in [14, Appendix B] to the nonbinary alphabet case.

Let

$$\Delta(c_1, c_2, \ldots, c_{|\mathcal{X}|}) \triangleq \frac{P_e(c_1, c_2, \ldots, c_{|\mathcal{X}|})}{\frac{1}{\mathcal{C}^{(|\mathcal{X}|-1)/2}} \prod_{1 \leq i \leq |\mathcal{X}|} \left(\frac{c_i}{\mathcal{C}}\right)^{c_i}}. \tag{32}$$

For any k, consider

$$\frac{\Delta(c_1, c_2, \ldots, c_k + 1, \ldots, c_{|\mathcal{X}|})}{\Delta(c_1, c_2, \ldots, c_k, \ldots, c_{|\mathcal{X}|})} = e^{f(c_k)+g(\mathcal{C})} \tag{33}$$

where

$$f(t) \triangleq \ln \frac{t^t(t+1/2)}{(t+1)^{t+1}} \tag{34}$$

and

$$g(t) \triangleq \left(t + \frac{1}{2}(|\mathcal{X}| - 1)\right) \ln \frac{t+1}{t} + \ln \left(\frac{t+1}{t + \frac{1}{2}|\mathcal{X}|}\right) \tag{35}$$

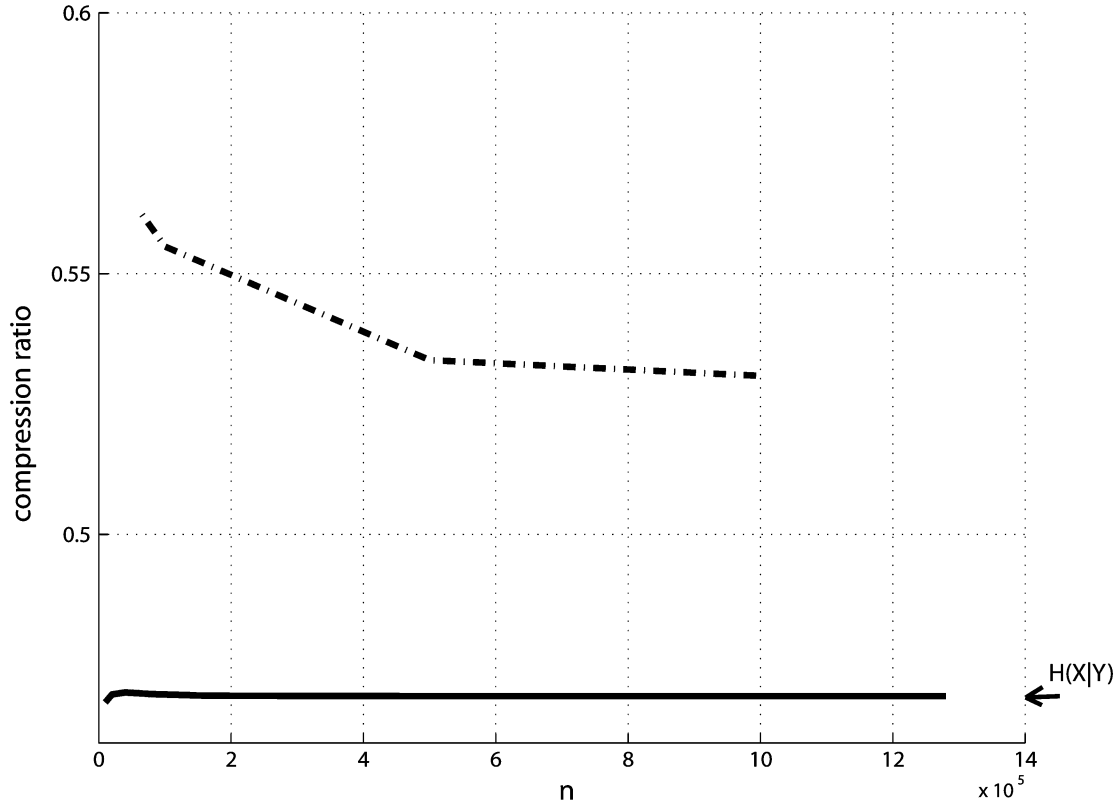

Fig. 3. Compression rates with side information of the processes in Example 1, as a function of length. Conditional entropy $H(X|Y) = 0.469$. The dashed (upper) curve corresponds to the CMPM algorithm and the solid (lower) curve corresponds to our CTW-based algorithm.

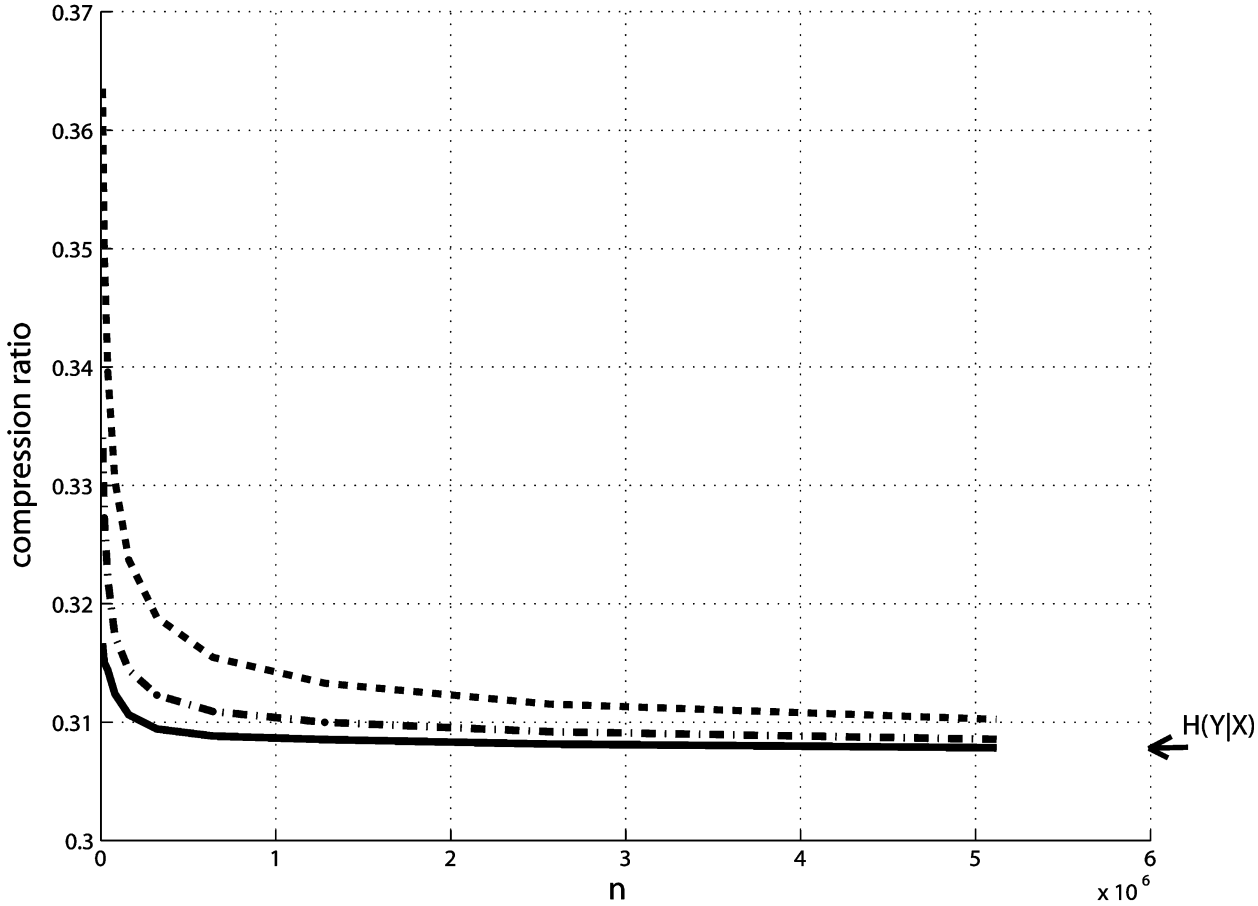

Fig. 4. Example 2. Compression ratio via CTW. Conditional entropy $H(Y|X) = 0.3075$. The lower (solid) curve corresponds to the case where both sequences are synchronized. The upper (dashed) curves correspond to the case where the side information is advanced by one and two samples, respectively.

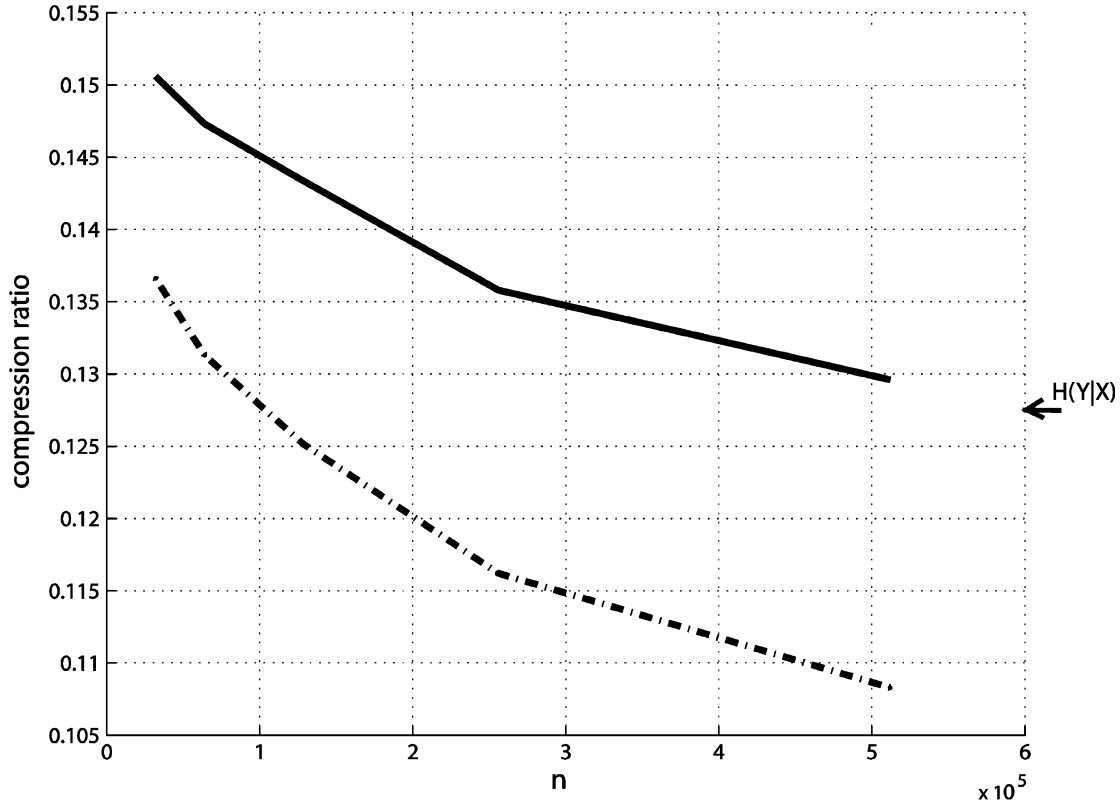

Fig. 5. Example 3. Compression ratio of *Emma* via multilevel conditional CTW. The channel parameter is $p = 0.99$, and the entropy of the noise is $H(W) = H(Y|X) = 0.1278$. The lower (dashed) curve corresponds to encoding the original source with noisy side information, while the upper (solid) curve corresponds to encoding the noisy source with the original source as side information.

for $t \in (0, \infty)$. The derivatives are

$$\frac{df(t)}{dt} = \ln \frac{t}{t+1} + \frac{1}{t+1/2} \leq 0 \tag{36}$$

for $t > 0$, and

$$\begin{aligned}\frac{dg(t)}{dt} &= \ln \frac{t+1}{t} - \frac{t+\frac{1}{2}(|\mathcal{X}|-1)}{t(t+1)} + \frac{\frac{1}{2}|\mathcal{X}|-1}{(t+1)(t+\frac{1}{2}|\mathcal{X}|)} \\ &\leq \frac{t+1/2}{t(t+1)} - \frac{t+\frac{1}{2}(|\mathcal{X}|-1)}{t(t+1)} + \frac{\frac{1}{2}|\mathcal{X}|-1}{(t+1)(t+\frac{1}{2}|\mathcal{X}|)} \\ &= -\frac{\frac{1}{2}|\mathcal{X}|-1}{t(t+1)} + \frac{\frac{1}{2}|\mathcal{X}|-1}{(t+1)(t+\frac{1}{2}|\mathcal{X}|)} \leq 0 \end{aligned} \tag{37}$$

for $t > 0$ and $|\mathcal{X}| \geq 2$. Since $\lim_{t\to\infty} f(t) = -1$ and $\lim_{t\to\infty} g(t) = 1$, we have

$$\begin{aligned}&\Delta(c_1, c_2, \ldots, c_k + 1, \ldots, c_{|\mathcal{X}|}) \\ &\qquad \geq \Delta(c_1, c_2, \ldots, c_k, \ldots, c_{|\mathcal{X}|})\end{aligned} \tag{38}$$

for $c_k \geq 0$. Therefore,

$$\Delta(c_1, c_2, \ldots, c_{|\mathcal{X}|}) \geq \Delta(1, 0, \ldots, 0) = \frac{1}{2}. \tag{39}$$

APPENDIX B

First, we prove the upper bound

$$\begin{aligned}\frac{1}{d} H\left(X_0^{d-1}|Y_0^{d-1}\right) &= \frac{1}{d}\left(H(X_0|Y_0^{d-1}) + H(X_1|X_0 Y_0^{d-1})\right. \\ &\quad \left. + \cdots + H\left(X_{d-1}|X_0^{d-2}Y_0^{d-1}\right)\right) \\ &\geq H\left(X_0|X_{-d+1}^{-1}Y_{-d+1}^{d-1}\right). \end{aligned} \tag{40}$$

On the other hand, the limit

$$\lim_{d\to\infty} H\left(X_i|Y_i, X_{i-d}^{i-1}, Y_{i-d}^{i-1}, Y_{i+1}^{i+d}\right) \tag{41}$$

exists, and we have

$$\begin{aligned}&\frac{1}{d}\left(H(X_0|Y_0) + H(X_1|X_0Y_0^2) + \cdots \right. \\ &\qquad \left. + H(X_{d-1}|X_0^{d-2}Y_0^{2d-2})\right) \geq \frac{1}{d} H(X_0^{d-1}|Y_0^{2d-2}). \end{aligned} \tag{42}$$

The left-hand side of (42) converges to the same limit as in (41), when $d \to \infty$. Therefore,

$$\lim_{d\to\infty} H\left(X_i|Y_i, X_{i-d}^{i-1}, Y_{i-d}^{i-1}, Y_{i+1}^{i+d}\right) = H(X|Y). \tag{43}$$

REFERENCES

[1] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1329–1339, Sep. 1996.

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

[3] D. Brunello, G. Calvagno, G. Mian, and R. Rinaldo, "Lossless compression of video using temporal information," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 132–139, Feb. 2003.

[4] G. Caire, S. Shamai (Shitz), and S. Verdú, "Practical schemes for interactive data exchange," in *Proc. Int. Symp. Information Theory and Its Applications (ISITA 2004)*, Parma, Italy, Oct. 2004.

[5] T. M. Cover, "A proof of the data compression theorem of Slepian and Wolf for ergodic sources," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 2, pp. 226–228, Mar. 1975.

[6] "CTW implementation v0.1," [Online]. Available: http://www.ele.tue.nl/ctw/

[7] J. Kieffer and E.-h. Yang, "Grammar-based lossless universal refinement source coding," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1415–1424, Jul. 2004.

[8] P. Koulgi, E. Tuncel, S. Regunathan, and K. Rose, "On zero-error source coding with decoder side information," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 99–111, Jan. 2003.

[9] A. Orlitsky, "Average-case interactive communication," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1534–1547, Jul. 1992.

[10] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[11] R. Stites and J. Kieffer, "Resolution scalable lossless progressive image coding via conditional quadrisection," in *Proc. Int. Conf. Image Processing (ICIP 2000)*, Vancouver, BC, Canada, Sep. 2000, vol. 1, pp. 976–979.

[12] P. Subrahmanya and T. Berger, "A sliding window Lempel-Ziv algorithm for differential layer encoding in progressive transmission," in *Proc. 1995 IEEE Int. Symp. Information Theory*, Whistler, BC, Canada, Jun. 1995, p. 266.

[13] P. Volf, "Weighting techniques in data compression: theory and algorithm," Ph.D. dissertation, Tech. Univ. Eindhoven, Eindhoven, The Netherlands, 2002.

[14] 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.

[15] F. M. J. Willems and T. J. Tjalkens, Complexity reduction of the Context-Tree Weighting Algorithm: A Study for KPN Research Complexity reduction of the Context-Tree Weighting Algorithm: A Study for KPN Research, EIDMA rep.t ser.: EIDMA-RS.97.01, Jan. 1997.

[16] F. M. J. Willems, "The context tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.

[17] H. S. Witsenhausen, "The zero-error side information problem and chromatic numbers," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 5, pp. 592–593, Sep. 1976.

[18] E.-h. Yang, A. Kaltchenko, and J. Kieffer, "Universal lossless data compression with side information by using a conditional MPM Grammar Transform," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2130–2150, Sep. 2001.