

A Universal Lossless Compressor with Side Information based on Context Tree Weighting

Haixiao Cai Sanjeev R. Kulkarni Sergio Verdú
Dept. of Electrical Engineering
Princeton University, Princeton, NJ 08544, USA

Abstract—This paper proposes a new algorithm based on the Context-Tree Weighting 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.

I. INTRODUCTION

We study the problem of universal lossless compression with side information: we wish to encode 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 **A** and **B** have identical copies of a file and **A** wants to convey an edited version of the file to **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 multi-resolution image coding where one may use low-resolution images as side information for high-resolution images [10], 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].

This work was supported in part by ARL MURI under Grant number DAAD19-00-1-0466, Draper Laboratory under IR&D 6002 Grant DL-H-546263, and the National Science Foundation under Grant CCR-0312413.

In almost lossless compression, the celebrated Slepian-Wolf-Cover result [9], [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].

Universal compression with side information known to both the encoder and decoder has been studied in [11], where a “conditional” version of the Lempel-Ziv algorithm was proposed. Note that asymptotic optimality has not been proved for this algorithm.

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.

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 i th 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.

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

The context tree uses the joint symbol $(x, y) \in \mathcal{X} \times \mathcal{Y}$ (see Figure 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 d -th order context, we have to take branches according to $(x_{i-k}, y_{i-k}, y_{i+k})$ for $1 \leq k \leq d$. 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} .

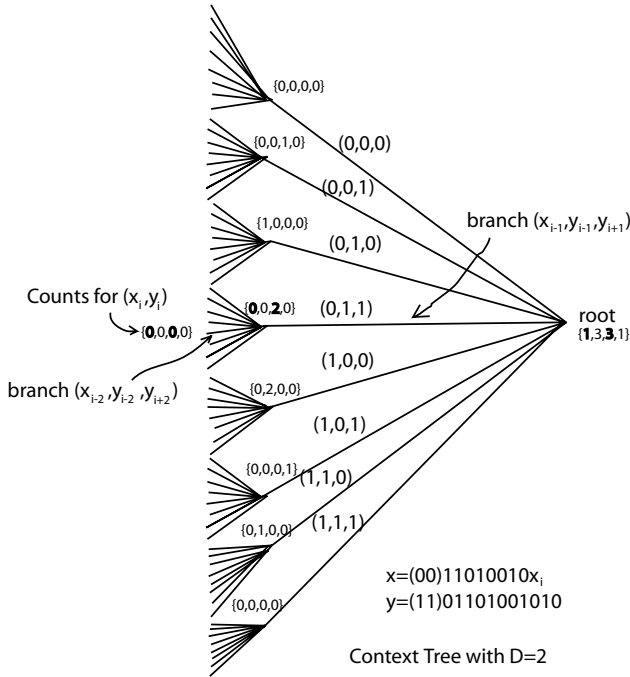


Fig. 1. Context Tree with $D = 2$ of joint source (x, y) . We encode x_i , where $i = 9$. A branch from depth $d - 1$ to depth d corresponds to $(x_{i-d}, y_{i-d}, y_{i+d})$, $d = 1, 2$.

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

- 1) Travel through the context tree according to $(x_{i-k}, y_{i-k}, y_{i+k})$, $k = 1, 2, \dots, D$, until a leaf node is reached. Notice 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:

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

The count $c_s(x_i, y_i)$ in node s is increased by 1. Then the weighted probability P_w^s of node s is updated:

$$P_w^s := \begin{cases} \frac{1}{2}P_e^s + \frac{1}{2} \prod_{v \in \text{Child}(s)} P_w^v & : 0 \leq l(s) \leq D \\ P_e^s & : l(s) = D, \end{cases} \quad (2)$$

where $l(s)$ is the depth of node s , and $\text{Child}(s)$ 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 Figure 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 Figure 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 $(x, y)_{i-1}^{i-1}$, the current symbol y_i and the future symbols y_{i+1}^n . For the current symbol (x_i, y_i) , $i = 1, 2, \dots, n$

- 1) Travel through the extended context tree according to $(x_{i-k}, y_{i-k}, y_{i+k})$, $k = 1, 2, \dots$ 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:

$$P_e^s := \frac{c_s(x_i, y_i) + \frac{1}{2}}{\sum_{x \in \mathcal{X}} c_s(x, y_i) + \frac{1}{2}|\mathcal{X}|} \cdot P_e^s \quad (3)$$

The count $c_s(x_i, y_i)$ in node s is increased by 1. Then the weighted probability \tilde{P}_w^s of node s is updated:

$$\tilde{P}_w^s := \begin{cases} \frac{1}{|\mathcal{X}|} & : \text{if } s \text{ is unique} \\ \frac{1}{2}P_e^s + \frac{1}{2} \prod_{v \in \text{Child}(s)} \tilde{P}_w^v, & : \text{otherwise} \end{cases} \quad (4)$$

where $Child'(s)$ is the set of children nodes of s in the extended context tree. Note that there are two special nodes in $Child'(s)$ 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

Omitting proofs because of space limitations, in this section we give our main results on the optimality of the compression algorithms with side information proposed in this paper. 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*

$$\limsup_{n \rightarrow \infty} \frac{L(x_1^n | y_1^n)}{n} \leq H(X_i | Y_i, X_{i-D}^{i-1}, Y_{i-D}^{i-1}, Y_{i+1}^{i+D}) \quad (5)$$

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 .

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

$$\limsup_{n \rightarrow \infty} \frac{L(x_1^n | y_1^n)}{n} \leq H(X|Y) \quad a.s. \quad (6)$$

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

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], [12].

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 i -th 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 Figure 2).
- 2) Hashing can be used to reduce the required memory and save space for pointers to children nodes.
- 3) Zero-redundancy estimator for binary sources

$$P_{e,ZR}(c_1, c_2) := \begin{cases} \frac{1}{2} P_e(c_1, c_2) & : c_1 > 0, c_2 > 0, \\ \frac{1}{2} P_e(c_1, 0) + \frac{1}{4} & : c_1 > 0, c_2 = 0, \\ \frac{1}{2} P_e(0, c_2) + \frac{1}{4} & : c_1 = 0, c_2 > 0, \\ 1 & : c_1 = c_2 = 0 \end{cases} \quad (7)$$

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.

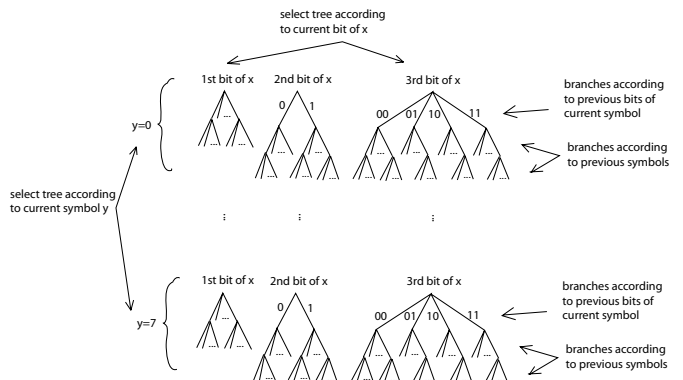


Fig. 2. Example of Multilevel Context Tree. $|\mathcal{X}| = |\mathcal{Y}| = 8$, $D = 2$.

In the experimental results in Section V, we find these techniques very useful in dealing with sources with large alphabets. 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 . Experiments show that this modification also improves the compression ratio.

V. SIMULATIONS

Example 1. We test the same example 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 $X_i = Y_i \oplus W_i$ where W_i is i.i.d. with the probability of symbol 0 being p . We have $H(X|Y) = H(W)$. When $p = 0.9$ and $q = 0.8$, $H(X|Y) = 0.469$. Figure 3 shows the compression ratio as a function of data size of both our algorithm and the CPM algorithm. The compression ratio can also be seen as an estimate of the conditional entropy rate.

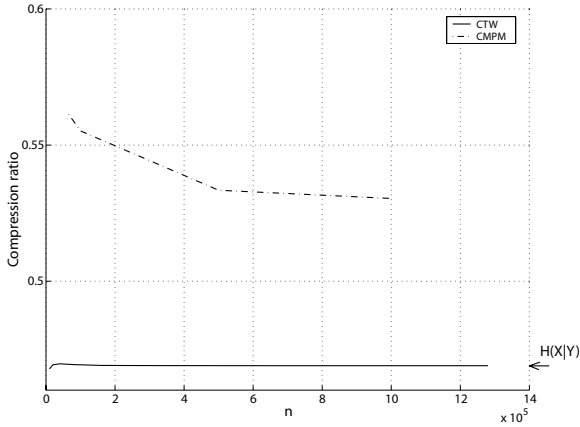


Fig. 3. Example 1. comparison of compression rates with side information. Conditional entropy $H(X|Y) = 0.469$.

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 Figure 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 generate a random sequence from a joint Markov chain with the following transition matrix:

$$\begin{aligned} & \{P[(X_{i+1}, Y_{i+1}) = (l_1, l_0) | (X_i, Y_i) = (j_1, j_0)]\} \\ &= \begin{bmatrix} 0.12 & 0.32 & 0.14 & 0.42 \\ 0.23 & 0.32 & 0.33 & 0.12 \\ 0.1 & 0.1 & 0.3 & 0.5 \\ 0.43 & 0.16 & 0.21 & 0.2 \end{bmatrix}, \end{aligned} \quad (8)$$

where (j_1, j_0) is the binary representation of the j -th row, and (l_1, l_0) is the binary representation of the l -th column. We have $H(Y) = 0.9421$, $H(X) = 0.9882$, and $H(X, Y) = 1.8254$. $I(X; Y) = 0.1049$ and $H(X|Y) = 0.8833$. The long-dash line corresponds to the case where sequence x and sequence

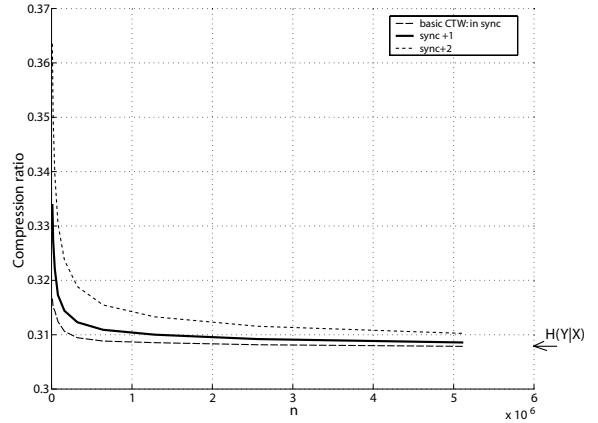


Fig. 4. Example 2. Conditional compression ratio via CTW. X takes the role of side information. “in sync” means sequence x and y are synchronized; “sync +1” means that we have advanced the sequence x by 1. Conditional entropy $H(Y|X) = 0.3075$.

y are synchronized. The solid line corresponds to the case where sequence y is advanced by 1; and the short-dash line corresponds to the case where sequence y is advanced by 2. (See Figure 5.)

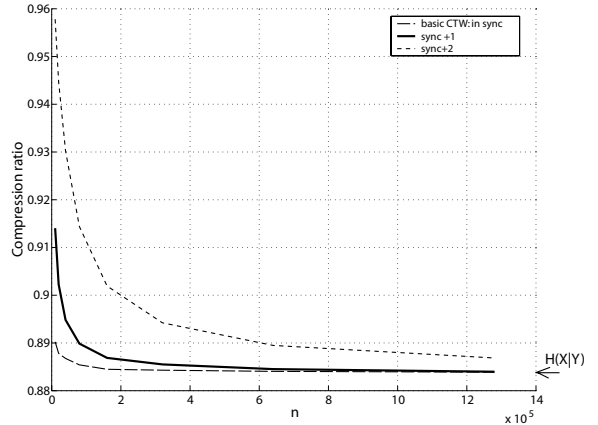


Fig. 5. Example 3. Conditional compression ratio via CTW: $H(X|Y) = 0.8833$. “in sync” means that sequence x and y are synchronized; “sync +1” means that we have advanced the sequence y by 1.

Example 4. 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. Conditional compression ratios with different values of p (where $p \in [0.8, 0.999]$) are shown in Figure 6,

where the novel used is “War and peace” by Leo Tolstoy. We have found that other novels yield similar results (not shown here). 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 $H(Y|X)$.

Example 5. Let Y be a noisy version of the original English text X as in the previous example. Now we further assume that the algorithm has the knowledge that Y is X observed through a discrete memoryless channel with known transition probability matrix. We compare compression ratios of the algorithm in Figure 7. The knowledge that Y is X observed through a discrete memoryless channel does help to improve the speed of convergence, because we condition only on the past symbols of X and future symbols of Y , and not on the past symbols of Y . It is interesting to compare with the following scheme: first use the Backward-Forward Product algorithm [19] (which is an alternative to the DUDE algorithm [13]) to obtain a denoised sequence Y' of Y and then use Y' as the side information to compress X . This is based on the heuristic that Y' has a lower symbol error rate than Y with respect to the original X . Although $H(X|Y')$ may be larger than $H(X|Y)$, this algorithm performs very well in practice when the symbol error rate of the denoised sequence is low. In our experiment, the symbol error rate of the denoised version Y' decreases from 9.2% to 6%, when the data size increases from 32K to 3000K symbols. (The symbol error rate of Y is 10%.)

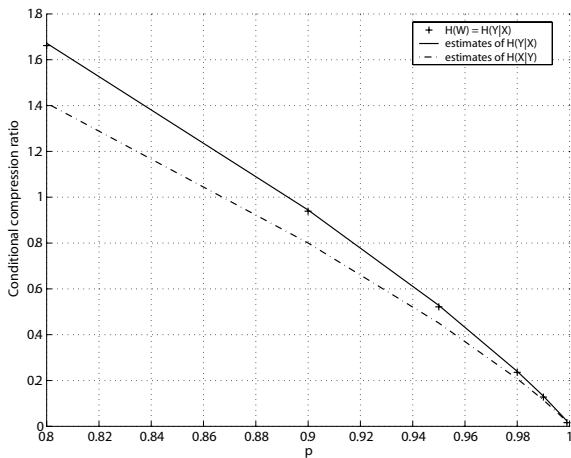


Fig. 6. Example 4. Conditional compression ratio of “War and Peace” via CTW. Sequence length $n = 480000$. The channel parameter p is the probability of correct symbol in the corrupted version, which is in the range of $[0.8, 0.999]$. The true entropy of the noise W is plotted with ‘+’.

REFERENCES

- [1] N. Alon and A. Orlitsky. “Source coding and graph entropies,” *IEEE Trans. Inform. Theory*, vol. 42, pp 1329-1339, Sept. 1996.
- [2] R. Begleiter and R. El-Yaniv. “On Prediction Using Variable Order Markov Models,” *Journal of Artificial Intelligence Research* 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 Processing*, vol. 12, pp. 132-139, Feb. 2003.

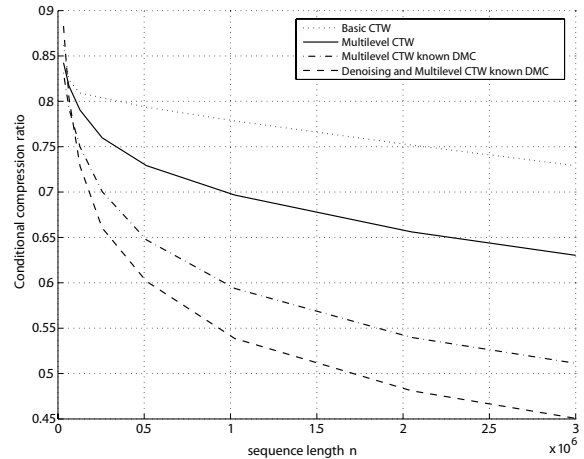


Fig. 7. Example 5. Conditional compression ratio of “War and Peace” via CTW. The channel parameter is $p = 0.9$, and the true entropy of the noise $H(W) = H(Y|X) = 0.939$. Comparison of four algorithms: the conditional CTW with side information Y , the multilevel conditional CTW with side information Y , the multilevel conditional CTW with knowledge that Y is X observed through a discrete memoryless channel, and the multilevel conditional CTW using the denoised sequence Y' as side information.

- [4] G. Caire, S. Shamai and S. Verdú. “Practical Schemes for Interactive Data Exchange,” in *Proc. ISITA 2004*, Parma, Italy, Oct. 10-13, 2004.
- [5] T. M. Cover. “A proof of the data compression theorem of Slepian and Wolf for ergodic sources,” *IEEE Trans. Inform. Theory*, vol. 22, pp. 226-228, Mar. 1975.
- [6] CTW implementation v0.1 <http://www.ele.tue.nl/ctw/>
- [7] J. Kieffer and En-hui Yang. “Grammar-based lossless universal refinement source coding,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 1415-1424, July 2004.
- [8] A. Orlitsky. “Average-case interactive communication,” *IEEE Trans. Inform. Theory*, vol. 38, pp. 1534-1547, July 1992.
- [9] D. Slepian and J. K. Wolf. “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, pp. 471-480, 1973.
- [10] R. Stites and J. Kieffer. “Resolution scalable lossless progressive image coding via conditional quadrisection,” in *Proc. ICIP 2000*, Vancouver, B.C., Canada.
- [11] P. Subrahmanya and T. Berger. “A sliding window Lempel-Ziv algorithm for differential layer encoding in progressive transmission,” in *Proc. IEEE Int. Symp. Inform. Theory*, Whistler, B.C., Canada, p. 266, 1995.
- [12] P. Volf. “Weighting Techniques in Data Compression: Theory and Algorithm,” Ph.D. thesis, Technische Universiteit Eindhoven, 2002.
- [13] 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.
- [14] 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.
- [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] F. M. J. Willems. “The Context Tree Weighting Method: Extensions,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 792-798, Mar. 1998.
- [17] H. S. Witsenhausen. “The zero-error side information problem and chromatic numbers,” *IEEE Trans. Inform. Theory*, vol. 22, pp. 592-593, Sept. 1976.
- [18] En-hui Yang, A. Kaltchenko and J. Kieffer. “Universal lossless data compression with side information by using a conditional MPM Grammar Transform,” *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2130-2150, Sept. 2001.
- [19] J. Yu and S. Verdú. “Schemes for bi-directional modeling of discrete stationary sources,” in *Proc. 2005 Conference on Information Sciences and Systems*, John Hopkins University, Mar. 2005.