Let $G = (V, E)$ be an instance of GNC, where $V = \{v_0, v_1, \dots, v_{K-1}\}$ and $E = \{e_0, e_1, \dots, e_{M-1}\}$. Without loss of generality, assume that V does not have isolated vertices. To reduce this to an instance of RPACK, consider a two-level binary tree and let R be a 4 by M rectangle. We label the M columns of R by the edges, i.e., cliques with two vertices. Define K sets of rectangles, $\{r_i : i = 0, 1, \dots, K-1\}$, all of height 1, such that r_i occupies column e_j if and only if vertex v_i is on edge e_j .

The essence of the reduction is to treat the four possible locations of $\{r_i\}$ as four colors. Choosing a location for r_i can be interpreted as picking a color for v_i . By the definition of $\{r_i\}$, r_{i_1} and r_{i_2} can be placed in the same location if and only if v_{i_1} and v_{i_2} are not joined by any edge, i.e., v_{i_1} and v_{i_2} can be labeled by the same color. Therefore, G is colorable by four colors if and only if $\{r_i\}$ can be packed into R in a nonoverlapping way. Since the reduction from GNC to RPACK can be done easily in polynomial time, therefore ZIC is NP-complete. $\square$

Fig. 3 shows an example of reducing GNC to RPACK.

ACKNOWLEDGMENT

The authors wish to thank the reviewers for their comments. In particular, suggestions from one of the reviewers helped generalize the results in [4] to the present form. The authors also wish to thank Prof. Alon Orlitsky for referring them to [2] and Prof. Ning Cai for providing a copy of [6].

REFERENCES

- [1] 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.
- [2] N. Alon and A. Orlitsky, "Source coding and graph entropies," *IEEE Trans. Inf. Theory*, vol. 42, no. 5, pp. 1329–1339, Sep. 1996.
- [3] 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.
- [4] Y.-O. Yan and T. Berger, "On instantaneous codes for zero-error coding of two correlated sources," in *Proc. IEEE Int. Symp. Information Theory*, Sorrento, Italy, Jun./Jul. 2000, p. 344.
- [5] P. Koulgi, E. Tuncel, S. L. Regunathan, and K. Rose, "On zero-error source coding with decoder side information," *IEEE Trans. Information Theory*, vol. 49, no. 1, pp. 99–111, Jan. 2003.
- [6] N. Cai and Z. Zhang, "On variable-length coding with side information for discrete memoryless sources," *ACTA Math. Appl. Sinica*, vol. 3, no. 4, pp. 289–297, Oct. 1987.
- [7] A. Kh., A. Jabri, and S. Al-Issa, "Zero-error codes for correlated information sources," *Proc. 6th IMA Int. Conf. Cryptography and Coding*, pp. 17–22, Dec. 1997.
- [8] Q. Zhao and M. Effros, "Lossless and near-lossless source coding for multiple access networks," *IEEE Trans. Inf. Theory*, vol. 49, no. 1, pp. 112–128, Jan. 2003.
- [9] H. Sato, "On the instantaneous code for multiple information sources," in *Proc. IEEE Int. Symp. Information Theory*, Les Arcs, France, Jun. 1982, p. 103.
- [10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

Faster Sequential Universal Coding Via Block Partitioning

Dror Baron, *Member, IEEE*, and Richard G. Baraniuk, *Fellow, IEEE*

Abstract—Rissanen provided a sequential universal coding algorithm based on a block partitioning scheme, where the source model is estimated at the beginning of each block. This approach asymptotically approaches the entropy at the fastest possible rate of $\frac{1}{2} \log(n)$ bits per unknown parameter. We show that the complexity of this algorithm is $\Omega(n \log(n))$, which is comparable to existing sequential universal algorithms. We provide a sequential $O(n \log(\log(n)))$ algorithm by modifying Rissanen's block partitioning scheme. The redundancy with our approach is greater than with Rissanen's block partitioning scheme by a multiplicative factor $1 + O(1/\log(\log(n)))$, hence it asymptotically approaches the entropy at the fastest possible rate.

Index Terms—Lossless source coding, redundancy, sequential coding, universal coding.

I. LIMITATIONS OF SEQUENTIAL UNIVERSAL CODING

Universal lossless source coding has made tremendous gains using context models [1]–[5]. The key idea is that the context, which consists of the last several symbols of the input sequence, often imposes constraints on the next symbol. For example, in English text a "q" is almost always followed by a "u." In the universal case, where the source statistics are unknown, one must estimate the conditional distribution of the symbol that follows the context. This estimation process accumulates statistics on which symbols follow other appearances of the context. Allowing large context depths during the context accumulation phase may yield better source models, but may also require excessive computation. Fortunately, for a length- n sequence generated by a stationary ergodic source, almost all repeating subsequences are of length $O(\log(n))$,¹ and thus $\Theta(\log(n))$ context depths yield near-optimal compression. Furthermore, even unbounded depth context searches are (usually) not too costly.

Previous authors exploited these properties to develop sequential universal algorithms with moderate computational requirements. For example, algorithm "context" [2] introduced context trees as a useful tool for source coding; the complexity for maintaining a context tree of depth $O(\log(n))$ is $O(n \log(n))$. Willems [3] described a sequential implementation of an unbounded depth mixture; for stationary ergodic sources, the typical complexity of his approach is $\Theta(n \log(n))$. In contrast, nonsequential algorithms such as the Burrows–Wheeler transform (BWT) [6] enable context accumulation to be performed in $O(n)$ time. For example, Baron and Bresler [4] provided an $O(n)$ BWT-based universal encoder. Martin *et al.* [5] used related tools to also provide an $O(n)$ universal decoder. Despite

Manuscript received December 3, 2004; revised December 16, 2005. This work was supported by AFOSR under Grant FA9550-05-1-0148, NSF under Grant CCF-0431150, ONR under Grant N00014-02-1-0353, and the Texas Instruments Leadership University Program.

The authors are with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA (e-mail: drorb@rice.edu; richb@rice.edu).

Communicated by S. A. Savari, Associate Editor for Source Coding.
Digital Object Identifier 10.1109/TIT.2006.871041

¹For two functions $f(n)$ and $g(n)$, $f(n) = O(g(n))$ if $\exists c, n_0 \in \mathbb{R}^+$, $0 \leq f(n) \leq cg(n)$ for all $n > n_0$. Similarly, $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$, and $f(n) = \Theta(g(n))$ if $\exists c_1, c_2, n_0 \in \mathbb{R}^+$, $0 \leq c_1g(n) \leq f(n) \leq c_2g(n)$ for all $n > n_0$.

these impressive advances in nonsequential coding, the development of low-complexity sequential universal coding algorithms is of great interest.

In the paper “Fast Universal Coding with Context Models,” Rissanen [1] provided a universal coding scheme based on partitioning the input into blocks. This work opened up a potential for low-complexity sequential coding. The model estimate is updated at the beginning of each block, and the symbols of each block are encoded based on the model estimate at the beginning of the block. In Rissanen’s scheme there are $\Theta(i)$ blocks in super block i , where the length of super block i is 2^i . (Super blocks are only used to define the initial locations of blocks; they serve no algorithmic purpose.) For a length- n sequence, Rissanen proved [1] that these block lengths asymptotically achieve the lowest possible universal redundancy, i.e., $\frac{1}{2} \log(n)$ bits per unknown parameter [7].

Combined with an $O(n \log(\log(n)))$ context accumulation algorithm, the aggregate complexity was claimed to be $O(n \log(\log(n)))$, including model estimation at the beginning of all the blocks [1]. This result improved upon previous $O(n \log(n))$ algorithms [2], [3].

Unfortunately, the $O(n \log(\log(n)))$ claim is somewhat ambitious [1, p. 1069, top of column 2]. Because there are $\log(n) + \Theta(1)$ super blocks, the last super block consists of $\Theta(\log(n))$ blocks. Therefore, the last $\Theta(\log(n))$ blocks all begin after $t = \Theta(n)$ symbols have already been processed. Furthermore, after having processed t symbols, the model estimation cost at the beginning of each block is $\Omega(t)$. We conclude that the aggregate computation required for model estimation is $\Omega(n \log(n))$. This $\Omega(n \log(n))$ aggregate complexity of Rissanen’s model estimation is similar to the complexity of previous sequential universal coding algorithms [2], [3]. Therefore, the development of low-complexity sequential universal coding algorithms is still of great interest.

In the remainder of this correspondence, we provide an $O(n \log(\log(n)))$ sequential universal algorithm by modifying Rissanen’s block partitioning scheme. The redundancy with our approach is greater than with Rissanen’s block partitioning scheme by a multiplicative factor $1 + O(1/\log(\log(n)))$; hence, it asymptotically approaches the entropy at the fastest possible rate.

II. FASTER BLOCK PARTITIONING SCHEME

In order to provide a faster sequential coding algorithm, we use a modified block partitioning scheme. We propose the following series of break points

$$\begin{aligned} t_{i,0} &= 2^{i-1} \\ d_i &= 2^{i-1 - \lfloor \log(\log(i)) \rfloor} \\ t_{i,j} &= t_{i,0} + j d_i = 2^{i-1} \left[1 + j 2^{-\lfloor \log(\log(i)) \rfloor} \right] \end{aligned}$$

where $t_{i,0}$ is the initial location of super block i , $t_{i,j}$ is the initial location of the j th block within that super block, $i \in \{1, 2, \dots\}$, $j \in \{0, 1, \dots, 2^{\lfloor \log(\log(i)) \rfloor} - 1\}$, $\lfloor \cdot \rfloor$ denotes the floor operator, and $\log(\cdot)$ is the base-two logarithm. For example, the first several blocks begin at $t_{1,0} = 1$, $t_{2,0} = 2$, $t_{3,0} = 4$, $t_{4,0} = 8$, $t_{4,1} = 12$, and $t_{5,0} = 16$.

In this block partitioning scheme, there are $\Theta(\log(i))$ blocks in super block i . Therefore, for a length- n input, there are at most $O(\log(\log(n)))$ blocks in each super block. Because the model estimation in super block i is $O(2^i)$ per block, the complexity for model estimation in super block i is upper bounded by $O(2^i \log(\log(n)))$.

Therefore, the complexity for model estimation over all super blocks is

$$O(n \log(\log(n))) \left[1 + \frac{1}{2} + \frac{1}{4} + \dots \right] = O(n \log(\log(n))).$$

Furthermore, context accumulation is $O(n \log(\log(n)))$ [1] or even $O(n)$ [4], [5], and so the aggregate computational complexity of our scheme is $O(n \log(\log(n)))$.

III. REDUNDANCY WITH FASTER SCHEME

Following Rissanen [1], we consider a tree source with K states. We define the redundancy R as the excess coding length in bits above the entropy H . In order to compute R for an algorithm based on such a block partitioning, we provide the following result [1].

Lemma 1: [1] For a source model with K unknown parameters, the redundancy $R_{i,j}$ incurred in the block whose initial location is $t_{i,j}$ satisfies

$$R_{i,j} \leq \frac{K(1+\epsilon)}{2 \ln(2)} \left(\frac{d_i}{t_{i,j}} + \frac{C_\epsilon}{t_{i,j}} \right)$$

where ϵ is any positive number, $t_{i,j}$ is greater than some $t(\epsilon)$, and C_ϵ is a constant.

With our specific block partitioning scheme, given ϵ there exists $\tilde{C}_\epsilon$ such that the result holds for all blocks, even those for which $t_{i,j} < t(\epsilon)$.

Lemma 2: Using our block partitioning scheme, for a source model with K unknown parameters, the redundancy $R_{i,j}$ incurred in the block whose initial location is $t_{i,j}$ satisfies

$$R_{i,j} \leq \frac{K(1+\epsilon)}{2 \ln(2)} \left(\frac{d_i}{t_{i,j}} + \frac{\tilde{C}_\epsilon}{t_{i,j}} \right).$$

Using this lemma, we can bound the redundancy of the entire scheme as follows.

Theorem 1: For a source model with K unknown parameters, the aggregate redundancy R incurred by our scheme satisfies

$$R < \frac{K(1+\epsilon)}{2} \left[1 + O\left(\frac{1}{\log(\log(n))}\right) \right] \log(n).$$

Proof: An increase in the input length from n to $2n$ increases R by roughly $\frac{1}{2}$ bit per unknown parameter, which is negligible with respect to $O(\log(n)/\log(\log(n)))$. Therefore, we assume without loss of generality that $n = 2^k$, where k is an integer. Summing over all blocks and incorporating Lemma 2, we have

$$\begin{aligned} R &= \sum_{i=1}^k \sum_{j=0}^{2^{\lfloor \log(\log(i)) \rfloor} - 1} R_{i,j} \\ &\leq \frac{K(1+\epsilon)}{2 \ln(2)} \sum_{i=1}^k \sum_{j=0}^{2^{\lfloor \log(\log(i)) \rfloor} - 1} \left(\frac{d_i}{t_{i,j}} + \frac{\tilde{C}_\epsilon}{t_{i,j}} \right). \end{aligned}$$

There are $O(\log(i))$ terms of the form $\tilde{C}_\epsilon/t_{i,j}$ in super block i , for a total of $O(\log(i)/2^i)$, hence their aggregate contribution over k super

blocks is $O(1)$ bits. Therefore, we concentrate on terms of the form $d_i/t_{i,j}$. Summing inside super block i , we have

$$\begin{aligned} & \sum_{j=0}^{2^{\lfloor \log(\log(i)) \rfloor} - 1} \frac{d_i}{t_{i,j}} \\ &= \sum_{j=0}^{2^{\lfloor \log(\log(i)) \rfloor} - 1} \frac{2^{i-1 - \lfloor \log(\log(i)) \rfloor}}{2^{i-1} [1 + j 2^{-\lfloor \log(\log(i)) \rfloor}]} \\ &= \sum_{j=0}^{2^{\lfloor \log(\log(i)) \rfloor} - 1} \frac{1}{2^{\lfloor \log(\log(i)) \rfloor} + j} \\ &< \ln \left(\frac{2 \cdot 2^{\lfloor \log(\log(i)) \rfloor} - 1}{2^{\lfloor \log(\log(i)) \rfloor} - 1} \right) \end{aligned} \quad (1)$$

$$< \ln(2) + \frac{1}{2[2^{\lfloor \log(\log(i)) \rfloor} - 1]} \quad (2)$$

where (1) is based on integrating $f(j) = (2^{\lfloor \log(\log(i)) \rfloor} + j)^{-1}$ from -1 to $2^{\lfloor \log(\log(i)) \rfloor} - 1$, and (2) relies on the inequality $\ln(1+x) \leq x$ for $x > 0$. Note also that the second term of the upper bound (2) is not defined for $i \in \{1, 2, 3\}$, for which the example of Section II indicates that $d_i/t_{i,j} = 1$. Summing over all k super blocks, we have

$$\begin{aligned} R &< \frac{K(1+\epsilon)}{2 \ln(2)} \left[3 + \sum_{i=4}^k \ln(2) \right. \\ &\quad \left. + \frac{1}{2(2^{\lfloor \log(\log(i)) \rfloor} - 1)} \right] + O(1) \\ &= \frac{K(1+\epsilon)}{2} \left[k + O(1) + \sum_{i=4}^k O\left(\frac{1}{\log(i)}\right) \right]. \end{aligned} \quad (3)$$

The summation in (3) satisfies

$$\begin{aligned} \sum_{i=4}^k \frac{1}{\log(i)} &= \sum_{i=\lfloor k^{\frac{1}{2}} \rfloor + 1}^k \frac{1}{\log(i)} + \sum_{i=4}^{\lfloor k^{\frac{1}{2}} \rfloor} \frac{1}{\log(i)} \\ &< \frac{k}{\log(k^{\frac{1}{2}})} + \frac{k^{\frac{1}{2}}}{\log(4)} \\ &= \frac{2k}{\log(k)} + \frac{k^{\frac{1}{2}}}{2} \\ &= O\left(\frac{k}{\log(k)}\right) \end{aligned} \quad (4)$$

where i) the first summation of (4) relies on the inequalities $i \geq k^{\frac{1}{2}}$ and thus $\log(i) > \log(k^{\frac{1}{2}})$; ii) the second summation of (4) uses $k^{\frac{1}{2}} \geq \lfloor k^{\frac{1}{2}} \rfloor$ and $\log(i) \leq \log(4)$; and iii) the outcome (5) arises because $k^{\frac{1}{2}} = O(k/\log(k))$. Because $\log(n) = k$, the summation in (3) is $O(\log(n)/\log(\log(n)))$, which completes the proof. $\square$

We can also show that ϵ decays to zero faster than $1/\log(\log(n))$. To do so, consider Appendix A in Rissanen [1]. Choose α , the deviation from the conditional symbol probability, to be $O(\sqrt{\log(n)/n})$. With this value for α , Rissanen's inner summation [1, p. 1070, eq. (24)] exceeds its ideal value by $O(\sqrt{\log(n)/n^{\frac{3}{2}}})$, which increases ϵ by $O(\sqrt{\log(n)/n})$. In a similar manner, the deviation β allowed from the state probability is chosen to be $O(\log(n)/\sqrt{n})$. The probability that β is exceeded vanishes polynomially in n , and so the first term in Rissanen's expression [1, p. 1070, eq. (30)] is negligible. Rissanen's second term increases by a multiplicative factor $O(\beta)$, which increases ϵ by $O(\log(n)/\sqrt{n})$. This rate of decay for ϵ is faster than $1/\log(\log(n))$. Therefore, the redundancy of our scheme is $\frac{1}{2} \log(n) + O(\log(n)/\log(\log(n)))$ bits per unknown parameter.

Theorem 1 shows that our block partitioning scheme suffers from a minor degradation of the redundancy, whereas the complexity was reduced from $O(n \log(n))$ to $O(n \log(\log(n)))$. One could easily develop block partitioning schemes that decrease the computation further, thus fulfilling the potential of Rissanen's block partitioning scheme [1]. However, $\log(\log(n))$ is quite small for sequences of practical interest. Therefore, in order to accelerate sequential coding algorithms further, there are more gains to be made on the practical side (reducing the constants) rather than the asymptotic aspect of reducing the order terms themselves. In contrast, the $O(\log(n)/\log(\log(n)))$ redundancy term may be quite large in practice. Therefore, we feel that future work on low-complexity sequential coding should focus on lowering the redundancy.

ACKNOWLEDGMENT

The author would like to thank to Jorma Rissanen for useful discussions, Gil Shamir for extensive comments on the manuscript, and Yoram Bresler for useful suggestions. They also thank the reviewers for their comments, which led to great improvements in the manuscript.

REFERENCES

- [1] J. Rissanen, "Fast universal coding with context models," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1065–1071, May 1999.
- [2] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory*, vol. IT-29, no. 5, pp. 656–664, Sep. 1983.
- [3] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.
- [4] D. Baron and Y. Bresler, "An $O(N)$ semipredictive encoder via the BWT," *IEEE Trans. Inf. Theory*, vol. 50, no. 5, pp. 928–937, May 2004.
- [5] A. Martin, G. Seroussi, and M. J. Weinberger, "Linear time universal coding and time reversal of tree sources via FSM closure," *IEEE Trans. Inf. Theory*, vol. 50, no. 7, pp. 1442–1468, Jul. 2004.
- [6] M. Burrows and D. J. Wheeler, "A Block-Sorting Lossless Data Compression Algorithm," Digital Systems Research Center, Palo Alto, CA, SRC Res. Rep. 124, May 1994.
- [7] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Inf. Theory*, vol. IT-30, no. 4, pp. 629–636, Jul. 1984.