Universal Data Compression with LDPC Codes

Giuseppe Caire⋆, Shlomo Shamai† and Sergio Verdú‡
⋆ Eurecom, Sophia-Antipolis, France
† Technion, Haifa, Israel
‡ Princeton University, Princeton NJ, USA

Abstract: *We present a new universal noiseless compressor of sources with memory based on the concatenation of the Burrows–Wheeler block sorting transform (BWT) with the syndrome former of an LDPC code. The proposed scheme makes use of a library of LDPC parity-check matrices of different rates and of a simple method to estimate and encode the tree source model from the BWT output. Unlike existing works that use error-correcting codes for data compression, our scheme can deal with sources with memory and achieves lossless compression. Our method offers competitive performance over existing methods such as Lempel-Ziv (*`gzip`*) and standard BWT–based schemes (*`bzip`*), while being amenable to joint source-channel decoding.*

Keywords: LDPC codes, universal data compression, Burrows-Wheeler transform

1 Introduction

In [1], we proposed an explicit scheme for fixed-length data compression/decompression based on the concatenation of the Burrows-Wheeler block sorting transform (BWT) (see [2] and references therein) with the syndrome former of a linear error-correcting code. Using a sequence of capacity-achieving codes we obtain an asymptotically optimal scheme for a large class of finite-memory stationary and ergodic sources. In practice, however, we wish to have encoding and decoding schemes with linear complexity in the blocklength. Therefore, in the proposed scheme we replace the asymptotically optimal sequence of codes by low-density parity-check (LDPC) codes (see [3] and references therein) and the asymptotically optimal decoding scheme (e.g., ML or typical-set decoding) by iterative Belief-Propagation (BP) decoding as the linear code component.

By relaxing the fixed-length requirement, the scheme of [1] can be made *exactly* lossless at the cost of a small variance of the output length. The basic scheme of [1] is not universal, as it assumes that the source statistics is known to both the encoder and the decoder. Since universal encoding is variable-length by its very nature, as it applies to sources with possibly different entropy, here we focus only on the variable-length option of the scheme of [1] and we describe an augmented scheme to handle universal data compression.

The reminder of this paper is organized as follows. Section 2 recalls the basics of the universal data compression problem and describes the proposed coding scheme in general. Section 3 recalls the variable-length LDPC encoding-decoding scheme of [1], which is the basic building block of the proposed universal scheme. Section 4 describes the details of the universal scheme, and Section 5 presents some numerical comparisons with standard compression methods.

2 Universal data compression

Consider a class of stationary ergodic sources $\{P_\theta : \theta \in \Lambda\}$ indexed by some parameter θ and defined over the alphabet $\mathcal{X}$. For $\mathbf{x} \in \mathcal{X}^n$ define the empirical entropy

$$\widehat{H}_\theta(\mathbf{x}) = -\frac{1}{n}\log_2 P_\theta(\mathbf{x}) \tag{1}$$

and the entropy rate $H_\theta(\mathcal{X}) = \lim_{n\to\infty} \mathbb{E}_\theta[\widehat{H}_\theta(\mathbf{x})]$ (where $\mathbb{E}_\theta[\cdot]$ indicates expectation w.r.t. the probability measure P_θ). Consider a coding strategy to encode n-sequences $\mathbf{x}$ with output length $\ell_n(\mathbf{x})$. The pointwise and expected redundancy per input letter are defined by

$$\delta_n(\mathbf{x}|\theta) = \frac{1}{n}\ell_n(\mathbf{x}) - \widehat{H}_\theta(\mathbf{x}) \tag{2}$$

and by $\delta_n(\theta) = \frac{1}{n}\mathbb{E}_\theta[\ell_n(\mathbf{x})] - H_\theta(\mathcal{X})$, respectively. For Λ being a compact set in $\mathbb{R}^K$, both $\delta_n(\mathbf{x}|\theta)$ and $\delta_n(\theta)$ are lower bounded by $(K/2)\log_2(n)/n + O(1/n)$ [4]. On the other hand, this rate of convergence is shown to be achievable and, for the class of stationary finite memory sources (FMS) with S states, coding schemes achieving pointwise redundancy $\leq S(|\mathcal{X}|-1)/2\log_2(n)/n + O(1/n)$ have been proposed (see [5] and references therein).

For simplicity, we focus on binary FMSs. We now give a general overview of our scheme. Consider a collection of Q LDPC ensembles [1] (defined by the left and right degree distributions (λ_q, ρ_q) for $q = 1, \ldots, Q$) of rates $0 < R_1 < \ldots < R_Q < 1$, and, for each q-th ensemble, a set of c parity-check matrices $\mathcal{H}_q = \left\{\mathbf{H}_{i,q} \in \mathbb{F}_2^{m_q \times n} : i = 1, \ldots, c\right\}$ such that $m_q/n = 1 -$

[1] We assume that the reader is familiar with the basics of LDPC coding and iterative decoding, with their Tanner graph representation and with standard terminology such as "bitnodes", "checknodes", "edges" and left and right "degree sequences". This background can be found, for example, in the special issue of *IEEE Trans. on Inform. Theory* [3].

R_q and each matrix $\mathbf{H}_{i,q}$ is independently and randomly generated over the ensemble (λ_q, ρ_q).

Define a finite set of possible source models $\{P_\theta : \theta \in \mathcal{S}\}$, and assume that the model $\widehat{\theta}$ can be described by $L_n(\theta)$ bits. Finally, define the length function

$$M_n : \{1, \ldots, Q\} \times \mathcal{S} \times \mathbb{F}_2^n \to \{1, \ldots, n\} \qquad (3)$$

such that for each binary sequence $\mathbf{x} \in \mathbb{F}_2^n$, source model P_θ and integer q, $M_n(q, \theta, \mathbf{x})$ is the output length of the basic LDPC compression scheme of [1] (see next section) applied to $\mathbf{x}$, with the set of LDPC parity-check matrices $\mathcal{H}_q$ and assuming the source probabilities P_θ.

Then, in order to encode $\mathbf{x}$, the proposed scheme finds the model in $\mathcal{S}$ and the set of parity-check matrices $\mathcal{H}_q$ that minimize the overall output length, i.e., it finds

$$(\widehat{q}, \widehat{\theta}) = \arg \min_{q,\theta} \{L_n(\theta) + M_n(q, \theta, \mathbf{x})\} \qquad (4)$$

It then encodes the source model $\widehat{\theta}$ using $L_n(\widehat{\theta})$ bits, and the sourceword $\mathbf{x}$ with the basic LDPC compression scheme using the parity-check matrices in $\mathcal{H}_{\widehat{q}}$.

The minimization in (4), although feasible in principle, is generally too complex for any practical purpose. In Section 4 we provide a heuristic but effective practical method to approximate such minimization with linear complexity in the block length n.

3 LDPC lossless compression of FMSs

In this section we recall the basic LDPC lossless data compression algorithm of [1], which is the main building block of the universal scheme outlined before.

For a parity-check matrix $\mathbf{H} \in \mathbb{F}_2^{m \times n}$, the corresponding syndrome-based fixed-length source encoder is given by the linear mapping $e : \mathbb{F}_2^n \to \mathbb{F}_2^m$ such that $e(\mathbf{x}) = \mathbf{H}\mathbf{x}$. The corresponding ML source decoder is the mapping $d_{\rm ml} : \mathbb{F}_2^m \to \mathbb{F}_2^n$ such that, for each $\mathbf{z} \in \mathbb{F}_2^m$, $d_{\rm ml}(\mathbf{z}) = \arg\max_{\mathbf{u} \in \mathbb{F}_2^n : \mathbf{H}\mathbf{u} = \mathbf{z}} P_X(\mathbf{u})$. When $\mathbf{H}$ is an LDPC (sparse) matrix, the syndrome-based encoding $e : \mathbf{x} \mapsto \mathbf{z}$ has linear complexity in the blocklength n. On the contrary, the ML decoder has generally exponential complexity in the blocklength. For LDPCs, the suboptimal BP iterative decoding proved to yield very good results on a variety of memoryless channels [3]. In the framework of source decoding, BP is more conveniently stated as a noise vector estimation problem.

Fix the realization of the input to the decoder, $\mathbf{z}$. The set of checknodes in which the bitnode $k \in \{1, \ldots n\}$ participates is denoted by $\mathcal{A}_k \subset \{1, \ldots m\}$, and the set of bitnodes which are connected to checknode $j \in \{1, \ldots m\}$ is denoted by $\mathcal{B}_j \subset \{1, \ldots n\}$. Define the a priori source log-ratios

$$\mathcal{L}_k = \log \frac{1 - p_k}{p_k}, \quad \forall \ \ k = 1, \ldots, n \qquad (5)$$

where, for the time being, we do not specify how the probabilities p_k are related to the joint source probability assignment P_X. For each iteration $t = 1, 2, \ldots$, the algorithm computes the value of the bitnodes

$$\widehat{x}_k = \text{sign} \left\{ \mathcal{L}_k + \sum_{j \in \mathcal{A}_k} \mu_{j \to k}^{(t)} \right\}$$

by updating the messages sent by the checknodes to their neighboring bitnodes and by the bitnodes to their neighboring checknodes, denoted respectively by $\mu_{j \to k}^{(t)}$ and by $\nu_{k \to j}^{(t)}$, according to the message-passing rules

$$\nu_{k \to j}^{(t)} = \mathcal{L}_k + \sum_{j' \in \mathcal{A}_k - \{j\}} \mu_{j' \to k}^{(t-1)} \qquad (6)$$

and

$$\mu_{j \to k}^{(t)} = (-1)^{z_j} 2 \tanh^{-1} \left(\prod_{k' \in \mathcal{B}_j - \{k\}} \tanh \left(\frac{\nu_{k' \to j}^{(t)}}{2} \right) \right) \qquad (7)$$

with the initialization $\mu_{j \to k}^{(0)} = 0$ for all $j \in \{1, \ldots, m\}$. In source coding the encoder has the luxury of running the decoding algorithm and check if successful decoding is achieved. Therefore, it can take several countermeasures to drive the decoder to successful decoding at the cost of a small additional redundancy [1]. In this work we consider the use of *Closed Loop Iterative Doping*. This technique, proposed in [1], consists of feeding to the BP decoder the source symbol with least reliability, i.e., for which $\left| \mathcal{L}_k + \sum_{j \in \mathcal{A}_k} \mu_{j \to k}^{(t)} \right|$ is minimum, every $D \geq 1$ iterations. The algorithm converges to error-free decoding in $t(\mathbf{x}) \leq nD$ iterations, and requires $d(\mathbf{x}) = \lfloor t(\mathbf{x})/D \rfloor$ redundancy bits in addition to the syndrome bits. The variance of the number of doped bits is greatly reduced by using $c > 1$ parity-check matrices and taking the minimum of the resulting $d(\mathbf{x})$.

It is immediate to show that estimating $\mathbf{x}$ via the above BP is equivalent to decoding the all-zero codeword transmitted through a time-varying BSC with noise realization $\mathbf{x}$. It follows that efficient LDPC parity-check matrices for the time-varying BSC are also efficient for the source coding problem at hand. In particular, if $p_k = p$ for all k, source coding rates close to the binary entropy $\mathcal{H}(p)$ can be obtained by designing LDPC codes for the BSC with (channel) coding rate R close to the BSC capacity $1 - \mathcal{H}(p)$.

Next, let us consider a stationary ergodic FMS. As we saw, it is easy to incorporate the knowledge about the source marginals in the BP decoding algorithm. However, for sources with memory the marginals alone do not suffice for efficient data compression. In this case, in [1] we proposed to use a one-to-one transformation, called the block-sorting transform or Burrows-Wheeler transform (BWT) [2] which performs the following operation: after adding

a special End-of-file symbol, it generates all cyclic shifts of the given data string and sorts them lexicographically. The last column of the resulting $(n+1) \times (n+1)$ matrix is the BWT from which the original data string can be recovered if we just add the row location of the original string in the BWT array, at the cost of $\log_2 n$ bits. By applying the BWT to the time-reversed source sequence $\mathbf{x}$, the symbols in the BWT output $\mathbf{y} = \mathrm{BWT}(\mathbf{x})$ are ordered lexicographically by their prefix. Hence, symbols having the same preceding context are grouped together. We refer to these groups of consecutive symbols in $\mathbf{y}$ with the same context in $\mathbf{x}$ as *segments*.

Note that the BWT performs no compression. Fashionable universal data compression algorithms (e.g. `bzip`) have been proposed which are quite competitive with the Lempel-Ziv algorithm. This is accomplished by exploiting the fact that the BWT output $\mathbf{y}$ (as the blocklength grows) is asymptotically piecewise i.i.d. (p.i.i.d.) [2]. For stationary ergodic tree sources and finite blocklength, the length, location, and distribution of the segments depend on the statistics of the source and on the source sequence realization, i.e., the transition points T_i separating the segments in the BWT output block are random variables. However, T_i/n converges to a deterministic limit for all i as $n \to \infty$. The universal BWT-based methods for data compression all hinge on the idea of compression for a memoryless source with an adaptive procedure which learns implicitly the local distribution of the segments, while forgetting the effect of distant symbols.

Our approach is to let the BWT be the front-end. Then we apply the LDPC parity-check matrix to the BWT output $\mathbf{y}$, as explained before. At the decoder, we exploit the fact that $\mathbf{y}$ is $\approx$ p.i.i.d., and treat the symbols as independent with probability $P(y_k = 1) = p_i$, for $k \in [T_{i-1}, T_i)$. The transitions T_i between the segments and the prior probabilities p_i are measured by the encoder by observing the source sequence $\mathbf{x}$ and its BWT $\mathbf{y}$, and must be communicated explicitly to the decoder.

At the decoder, once the compressed sequence has been processed by the BP decoder we apply the inverse BWT to recover the original source sequence. We note that the inverse BWT does not degrade gracefully with respect to errors. If only one input symbol is in error, the inverse BWT output will be seriously erroneous. Thus, even more than for memoryless sources, in this case the block-error rate is the main performance indicator for the decompression algorithm.

4 The universal scheme

If the encoder knew that the source is Markov with S states, then it could detect the segment transitions T_i just by looking at the first $M = \log_2 S$ columns of the BWT array. In this case, each probability p_i can be estimated by the empirical frequency of ones in the i-th segment. If the source memory is not known a priori, then all memories $M = 0, 1, \ldots,$ up to some maximum L can be considered, and the corresponding empirical frequencies can be obtained hierarchically, on a tree.

A coarse description of the segmentation and associated log-ratios $\mathcal{L}_i$ is supplied to the iterative decoder. The discrete set of possible sources $\mathbb{S}$ communicated to the decompressor corresponds to all possible quantized probabilities and quantized transition points, for all possible source memories. We denote the quantized parameters of a particular instance of the source model by $\theta \in \mathbb{S}$. We wish to encode $\{T_i\}$ and $\{\mathcal{L}_i\}$ by using

$$L_n(\theta) = \frac{S(\theta)}{2} \log_2 n + O(1) \quad \text{bits} \tag{8}$$

where $S(\theta)$ is the number of states in the model θ. Apart from the value of the constant term, this is the best possible redundancy to encode the source model, according to the minimum description length principle [4].

A simple algorithm approaching this lower bound is the following. For each given memory M, let the exact transition points of the segments identified by the first M columns of the BWT array be denoted by $T_i(M)$, for $i = 1, \ldots, 2^M - 1$. We can write $T_i(M) = \kappa_i \sqrt{n} + \zeta_i$, where $\zeta_i = T_i(\theta)$ modulo $\sqrt{n}$ and $\kappa_i = \lfloor T_i(M)/\sqrt{n} \rfloor$. Then, we quantize the reminder ζ_i by using b_1 bits and the transition point $T_i(M)$ is known up to a maximum error of $\sqrt{n} 2^{-b_1}$. We let $\widehat{T}_i(M)$ denote the quantized value corresponding to $T_i(M)$. Notice that after quantization some of the originally distinct transition points might have been quantized to the same value, i.e., some segments after quantization have disappeared from the resulting source model. Let $\{\widehat{T}_j(\theta) : j = 1, \ldots, S(\theta) - 1\}$, denote the set of *distinct* transition points after quantization, where $S(\theta)$ denotes the number of states in the source model θ. Notice that, by construction, $S(\theta) \leq 2^M$. Let p_j denote the empirical frequency of ones in the j-th segment identified by the transition points $\{\widehat{T}_j(\theta)\}$. We use b_2 bits to encode the log-ratios $\mathcal{L}_j(\theta) = \log(1-p_j)/p_j$. The decoder will apply the (quantized) log-ratio $\mathcal{L}_j(\theta)$ on the positions of segment j. Clearly, each κ_i can be encoded by $\frac{1}{2} \log_2 n$, therefore the description length for the model θ is

$$L_n(\theta) = (S(\theta) - 1) \left(\frac{1}{2} \log_2 n + b_2 \right) + S(\theta) b_1$$

which is compliant with the minimum description length principle (8).

The degrees of freedom available to the encoder are the model to be described $\theta \in \mathbb{S}$ and the ensemble of LDPC matrices to be chosen. It is problematic to evaluate $M_n(q, \theta, \mathbf{x})$ in (4), as this would require the application of the Closed-loop Iterative Doping Algorithm to $\mathbf{x}$, for all parity-check matrices $\{\mathbf{H}_{i,q} : q = 1, \ldots, Q, i = 1, \ldots, c\}$ and all models $\theta \in \mathbb{S}$. However, we make the following observation. Let $\widehat{H}_\theta(\mathbf{x})$ be the empirical entropy of $\mathbf{x}$ according to the probability model θ. Then, if $R_q > 1 - \widehat{H}_\theta(\mathbf{x})$, the number of doping bits $d(\mathbf{x})$ is very large. On the contrary, if $R_q < 1 - \widehat{H}_\theta(\mathbf{x})$, $d(\mathbf{x})$ is very small with respect to n. Hence, in (4) we shall replace $M_n(q, \theta, \mathbf{x})$ by

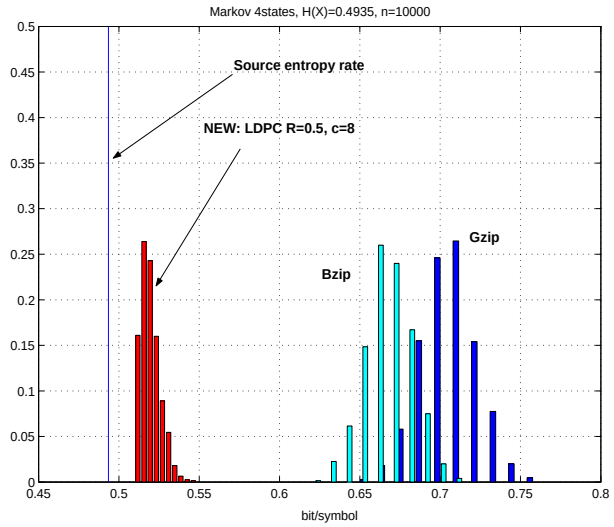

Figure 1: Histogram of normalized output lengths for a fixed Markov source with entropy = 0.4935 bit/symbol.

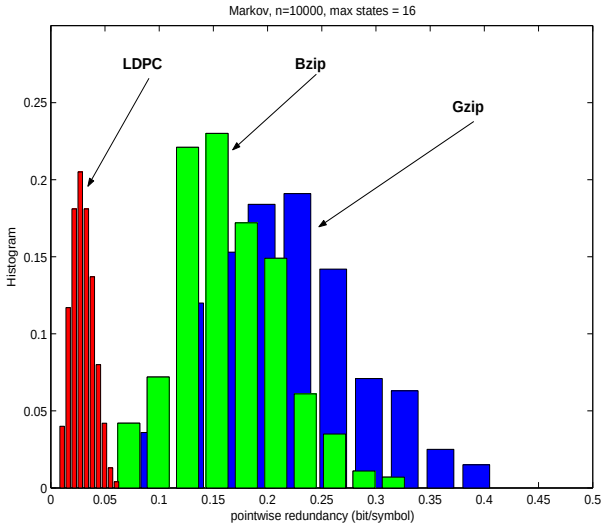

Figure 2: Histogram of codign redundancies for a random ensemble of Markov sources with blocklength equal to 10,000.

$m_{q(\theta)} = \max\{1 \le q \le Q : R_q < 1 - \widehat{H}_\theta(\mathbf{x})\}$. In other words, we choose the LDPC ensemble with largest rate not above the *normalized information density* of the deterministically time-varying BSC determined by θ with noise realization $\mathbf{x}$.

The BWT and the recursive segment determination can be obtained with complexity linear in n by using suffix-trees methods. This makes the overall complexity of our algorithm linear with n, although the (constant with n) complexity due to BP is large with respect to other universal linear complexity algorithms based on sequential arithmetic coding. A source of suboptimality of our algorithm with respect to the optimal pointwise redundancy is the discretization of the coding rate levels R_q. Although the maximum redundancy $\max |R_q - R_{q+1}|$ does not go to zero with n, it can be made as small as desired by choosing a sufficiently large Q.

5 Experiments

We compare the performance of the proposed algorithm with the standard compression software `gzip` (based on the Lempel-Ziv algorithm) and `bzip` (based on postprocessing the BWT output using Move-to-Front run-length coding and adaptive arithmetic coding.

Fig.1 shows the histogram of the normalized output lengths obtained from 2000 independent trials for a four-state binary Markov source with entropy rate 0.4935 bit/symbols, for the new scheme, `gzip` and `bzip`, for block length $n = 10,000$. We used $b_2 = 3$ quantization bits for the log-ratios, $b_1 = 3$ quantization bits for the transition points and a collection of irregular LDPC ensembles with rates equally spaced from 0.005 to 0.0995. For each ensemble, $c = 8$ parity-check matrices were randomly generated.

Instead of testing a given source, we also considered an ensemble of randomly generated binary Markov sources with number of states equally likely to be 1, 2, 4, 8 and 16 (i.e., with memory equal to 0, 1, 2, 3 and 4). The Markov source ensemble is obtained by generating independently the memory length, and then conditional distributions are also generated randomly, and it is restricted to produce sources with entropy ranging from 0.05 to 0.75 bit/symbol. Fig.2 examine histograms of the normalized redundancy (2) for our universal scheme, `gzip` and `bzip` for blocklengths equal to 10,000. For our scheme, the parameters b_1 and b_2 that govern the quantization coarseness in the description of the segmentation are adapted using the minimum description length principle described above.

REFERENCES

[1] G. Caire, S. Shamai and S. Verdu, "A new data compression algorithm for sources with memory based on error correcting codes," *Inform. Theory Workshop, ITW 2003*, Paris, April 2003.

[2] M. Effros, K. Visweswariah, S. Kulkarni, and S. Verdú, "Data compression based on the Burrows-Wheeler transform: Analysis and optimality," *IEEE Trans. on Information Theory*, vol. 48, pp. 1061–1081, May 2002.

[3] "Special issue on iterative decoding," *IEEE Trans. Inform. Theory*, vol. 47, Feb. 2001.

[4] J. Rissanen, "Universal coding, Information Prediction and Estimation," *IEEE Trans. on Information Theory*, vol. 30, No. 4, pp. 629–636, July 1984.

[5] A. Barron, J. Rissanen and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. on Information Theory*, Vol. 44, No. 6, pp. 2743-2760, Oct 1998.