

Universal Estimation of Entropy and Divergence via Block Sorting¹

Haixiao Cai, Sanjeev R. Kulkarni, and Sergio Verdú
 Dept. of EE, Princeton University, Princeton NJ 08544
 {hcai, kulkarni, verdu}@ee.princeton.edu

In this paper, we present a new algorithm to estimate both entropy and divergence of two finite-alphabet, finite-memory tree sources, using only information provided by a realization from each of the two sources. Our algorithm outperforms a previous LZ-based method proposed in [3]. It is motivated by data compression based on the Burrows-Wheeler Block Sorting Transform, using the fact proved in [1] that if the input is a finite-memory tree source, then the divergence between the output distribution and a piecewise stationary memoryless distribution vanishes as the length of the input sequence goes to infinity. Let's denote the alphabet set by χ , and the state set by S .

The BWT helps us estimate the conditional probabilities in each state by grouping letters in the same state together, removing the need to know the state set or the memory length explicitly when estimating the probability of the sequence \mathbf{z} . The entropy estimator for $H(Z)$ has four steps:

- Run the BWT on the reversed sequence \mathbf{z}' . The output sequence $\text{BWT}(\mathbf{z}')$ is like a concatenation of i.i.d. segments.
- Divide the output sequence $\text{BWT}(\mathbf{z}')$ into segments properly.
- Estimate the conditional probabilities within each segment, and calculate the logarithm of the probability of each segment.
- Average the individual estimates to get the estimate of $H(Z)$.

In order to estimate the divergence $D(q_z \| p_x)$, we have to deal with the term $p_x(z^n)$. We can gather letters having the same context from both \mathbf{z} and \mathbf{x} together by applying the BWT on the concatenation of the two sequences, i.e. $\mathbf{z}+\mathbf{x}$. Letters from \mathbf{x} and letters from \mathbf{z} are treated equivalently in the sorting procedure of the BWT, but we assign a bit for each letter in the output sequence to indicate whether it comes from \mathbf{x} or from \mathbf{z} . To illustrate this, we use upper case letters for \mathbf{x} , and lower case letters for \mathbf{z} . Our divergence estimator for $D(q_z \| p_x)$ has three major steps:

- Estimate $H_z = -\frac{1}{n} \log q_z(z^n)$.
- Estimate the cross term $-\frac{1}{n} \log p_x(z^n)$.

$$p_x(j, A) = N_j(A) / \sum_{B \in \chi} N_j(B), \forall A \in \chi \quad (1)$$

$$\log p_x(j) = \sum_{b \in \chi} N_j(b) \log p_x(j, B). \quad (2)$$

$$-\frac{1}{n} \log p_x(z^n) = -\frac{1}{n} \sum_{j=1}^{T_x} \log p_x(j). \quad (3)$$

Note, $q_z(\cdot)$ in step a and $p_x(\cdot)$ in step b are estimated in two different segmentations. In step b, we divide $\text{BWT}(\mathbf{z} + \mathbf{x}')$ into T_x segments according to upper case letters from \mathbf{x} . And $p_x(j, \cdot)$ is estimated in each segment j . Lower case letters do not influence the estimation of $p_x(j, \cdot)$, but we know which segment each letter belongs to. And we count $N_j(b)$, which is the number of lower case letters 'b' in segment j . $\log p_x(j)$ is

the logarithm of the probability of lower case letters from \mathbf{z} in segment j , where the probability is according to the estimates $p_x(j, \cdot)$ determined by upper case letters from \mathbf{x} .

- Subtract the entropy term from the cross term.

The remaining problem is how to do the segmentation efficiently. A simple way is to divide the BWT output sequence into equal-length segments. The segment-length should be chosen such that those segments containing transitions are negligible, but the estimates within each segment are reliable. We further develop an adaptive algorithm using the transition detection method introduced in [2]. We make a new segment only when we detect a transition.

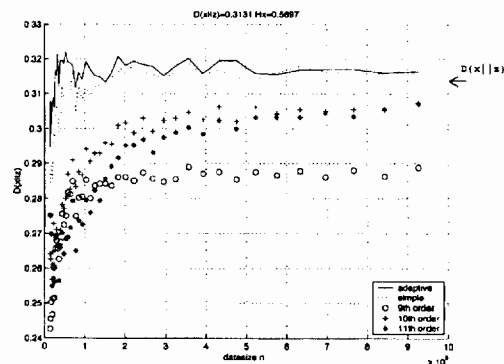
If we assume correct segmentation, then the MSE's of the estimators for Markov sources are

$$E[(\hat{H}(Z) - H(Z))^2] = \frac{1}{n} \text{Var}^2[\log_2 q(Z|S)] + O\left(\frac{1}{n^2}\right). \quad (4)$$

$$E[(\hat{D}(q||p) - D(q||p))^2] = \frac{1}{n} \text{Var}^2\left[\log_2 \frac{q(Z|S)}{p(Z|S)}\right] + \frac{1}{n \ln^2 2} \sum_{s \in S} \frac{q^2(s)}{p(s)} \left[\sum_{i \in \chi} \frac{q^2(i|s)}{p(i|s)} - 1\right] + O\left(\frac{1}{n^2}\right). \quad (5)$$

Similar results for i.i.d. sources are obtained by simply removing the conditioning on the states.

We compare the performance of the new algorithms and the empirical distribution plug-in schemes assuming Markov models of different orders. The LZ-based method converges very slowly (not shown in the figure). At data size of 10^6 , its divergence estimate is 0.22, while our estimate is 0.316.



REFERENCES

- M. Effros, K. Visweswariah, S. R. Kulkarni, S. Verdú. "Universal Lossless Source Coding with the Burrows Wheeler Transform", to appear in *IEEE Trans. Inform. Theory*, 2002.
- Gil I. Shamir. "Asymptotically optimal low-complexity sequential lossless coding for piecewise-stationary memoryless sources – Part I: the regular case". *IEEE Trans. Inform. Theory*, vol. 46, No. 7, pp. 2444–2467, Nov. 2000.
- J. Ziv, N. Merhav. "A measure of relative entropy between individual sequences with application to universal classification". *IEEE Trans. Inform. Theory*, vol. 39, No. 4, pp. 1270–1279, July 1993.

¹This work was supported in part by the National Science Foundation under contract number ECS-9873451 and by the Army Research Office under grant number DAAD19-00-1-0466.