Speeding up Arithmetic Coding using Greedy Re-normalization

Yunwei Jia En-hui Yang Da-ke He Steven Chan
Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
E-mails: {yjia,ehyang,dhe,schan}@bbcr.uwaterloo.ca

A typical adaptive arithmetic coder consists of three steps: range calculation, re-normalization, and probability model updating. A method, called greedy re-normalization, is given in this paper that significantly reduces the computational complexity of the re-normalization step of arithmetic coding. This is achieved by reducing both the number of re-normalizations required to encode a sequence and the number of operations within each re-normalization.

Following the notations in [1], the internal state of an arithmetic coder is represented by an interval $[L, L+R)$ where L is the base of the interval and R the length, and both L and R are b-bit integers. To reduce the number of operations within each re-normalization, the greedy re-normalization method generates as many code bits as possible from an interval and updates the interval correspondingly; moreover, both of the tasks are done at one time without using a loop. This is one of the differences between the greedy re-normalization method and the conventional method in [1] where a loop is used to generate code bits and update an interval. Let $m(L, R)$ represent the maximal number of code bits with known polarities that can be generated from $[L, L+R)$ and $k(L, R)$ the maximum number of *outstanding* code bits (see [1] for the meaning of this term). We show that $m(L, R) = b - 1 - \lfloor \log t \rfloor$, and $k(L, R) = \lfloor \log t \rfloor - \lceil \log(2^{1+\lfloor \log t \rfloor} - t) \rceil$ if $R \neq 1$ and $\lceil \log(2^{1+\lfloor \log t \rfloor} - t) \rceil \geq \lfloor \log R \rfloor$, and $\lfloor \log t \rfloor - \lfloor \log R \rfloor$ otherwise. (log stands for the logarithm with base 2.) The maximal number of bits that can be shifted out of L and R is given by $m(L, R) + k(L, R)$.

To reduce the number of re-normalizations in encoding a sequence, the proposed method adopts a dynamic re-normalization criterion: $Rp_{min} < 1$, where p_{min} stands for the probability of the least probable symbol in the source alphabet. Since the value p_{min} may change from time to time, this criterion is a dynamic one. This is another difference between the proposed method and the conventional method where a fixed re-normalization threshold is usually used.

Experimental results show that the proposed greedy re-normalization method indeed improves the speed of arithmetic coding. For example, by simply replacing the re-normalization procedure in the binary arithmetic coder in [1] with the proposed greedy re-normalization, more than 20% gain in coding speed is observed in experiments. By combining the greedy re-normalization method with some ideas of the QM-Coder, about 30% average gain over the QM-coder is observed for binary *i.i.d.* sources with the probability of the less probable symbol ranging from 0.01 to 0.45. Compression performance is not compromised in either of the two cases. The greedy re-normalization method can also be used in multi-symbol arithmetic coding.

References

[1] A. Moffat, R. M. Neal, and I. H. Witten, "Arithmetic coding revisited", *ACM Transactions on Information Systems*, Vol. 16, pp. 256–294, 1998.

Proceedings of the Data Compression Conference (DCC'03)
1068-0314/03 $17.00 © 2003 **IEEE**