Fast Codes for Large Alphabet Sources and Its Application to Block Encoding *

Boris Ryabko[†] and Jaakko Astola [‡]

[†] Siberian State University of Telecommunication and Computer Science, Russia
[‡] Technical University of Tampere, Finland

For many adaptive codes the speed of coding depends substantially on the alphabet size. Thus, the number of operations of an obvious (or naive) method of updating the cumulative probabilities is proportional to the alphabet size N. Jones and Ryabko have independently suggested two different algorithms of updating, which perform all the necessary transitions between individual and cumulative probabilities in $O(\log N)$ operations, where N is the alphabet size. Later many such algorithms have been developed and investigated in numerous papers.

In this paper we suggest a method for speeding up codes based on the following main idea. Letters of the alphabet are put in order according to their probabilities (or frequencies of occurrence), and the letters with probabilities close to each others are grouped in subsets (as new super letters), which contain letters with small probabilities. The key point is the following: equal probability is ascribed to all letters in one subset, and, consequently, their codewords have the same length. This gives a possibility to encode and decode them much faster than if they are different. Then each subset of the grouped letters is treated as one letter in the new alphabet, whose size is much smaller than the original alphabet. Such a grouping can increase the redundancy of the code. It turns out, however, that a large decrease in the alphabet size may cause a relatively small increase in the redundancy. More exactly, the number of the groups as a function of the redundancy (δ) increases as $c(\log N + 1/\delta) + c_1$, where N is the alphabet size and c, c_1 are constants.

The suggested algorithm is applied to block coding. The main problem of block encoding is that the number of blocks grows exponentially when the block length grows. In fact, it means exponential increasing of the input alphabet. In order to surmount this obstacle we suggest applying the described method of grouping to the set of all possible blocks in such a way that the redundancy caused by grouping is relatively small whereas the size of new alphabet is much less than the number of the possible blocks. As a result, the average number of operations per a source letter will be much less than for letterwise code.

*The first author was supported by INTAS grant no. 00-738 and Russian Foundation for Basic Research under Grant no. 03-01-00495. The part of results was presented at 2003 International Symposium on Information Theory, Yokohama, Japan .