

CSE 490 GZ

Assignment 3

Due Friday, January 30, 2004

1. For this problem we compare adaptive Golomb and adaptive arithmetic coding for a very special class of inputs in the alphabet $\{0, 1\}$. For the adaptive Golomb code we use the one described in problem 3 of Assignment 2. For the adaptive arithmetic coder we use the simple one that starts with each symbol having frequency 1. After an input symbol is encoded the frequency count for the symbol is increased by 1. The new frequencies are then used to encode the next symbol.
 - (a) Encode 0^71 and $0^{15}1$ with the adaptive Golomb coder.
 - (b) Encode 0^71 and $0^{15}1$ with the adaptive arithmetic coder. Assume that the decoder knows that the 1 indicates the end of the input.
 - (c) Compute the length, as a function of n , of the encoding of $0^{n-1}1$ for n a power of two for the adaptive Golomb coder.
 - (d) Estimate the length, as a function of n , of the encoding of $0^{n-1}1$ for n a power of two for the adaptive arithmetic coder. You can use information theory for this by figuring out the size W of the interval representing the string. The arithmetic code will have length approaching $\lceil \log_2(1/W) \rceil + 1$.
2. Let us try LZW on a special class of inputs too. Again assume the two symbol alphabet $\{0, 1\}$. Consider the following strategy for encoding the dictionary symbols from LZW. Start with a dictionary of size 2 and use just one bit to transmit a symbol. When the dictionary fills up we double its size to 4 and use two bits to transmit a word in the dictionary. This doubling happens whenever the dictionary fills.
 - (a) Encode 0^6 and 0^{28} with this version of LZW.
 - (b) Compute the length, as a function of n , of the encoding of 0^n with this version of LZW. (You may restrict yourself to easy n 's to work with if that helps.)
 - (c) Encode 0^6 and 0^{28} using the γ -code to represent the dictionary symbols from LZW on the strings 0^6 and 0^{28} .
 - (d) Compute the length, as a function of n , of the encoding of 0^n using the γ -code to represent the dictionary symbols of LZW. (You may restrict yourself to easy n 's to work with if that helps.)
 - (e) How does LZW compare to adaptive Golomb and adaptive arithmetic coding for compressing binary strings with very few 1s? Explain the comparison in one brief paragraph.