Homework Project #03: Context-based Binary Arithmetic Coding of “jokes.txt”

Due Date: 11/23/2014

Binary Arithmetic Coding (BAC)

- □ In this HW project, you must write a context-based binary arithmetic coding program to compress the data source "jokes.txt"
- □ Note that the alphabet of a binary arithmetic coder is {0, 1}, while the data source jokes.txt has a alphabet set {"A" ~ "Z", " ", "$"}, while "$" should be appended to "jokes.txt" as the end-of-sequence symbol.

The Binarization Process

- □ To convert the original data source to a binary source, please use the following binarization rule:
  - ■ The binary code of “A” ~ “Z” equal 00001 ~ 11010
  - ■ The binary code of “ “ equals 11011
  - ■ The binary code of “$” equals 00000
- □ After the data source “jokes.txt” is converted to a binary source, you can use a BAC coder to encode the sequence

Implementation of the BAC Coder

- You can write the BAC coder based on the example described in section 4.6 of the textbook
- Note that you do not have to optimize your coder using the QM, MQ, or M coders techniques described in section 4.6.1, 4.6.2, and 4.6.3 of the textbook
  - These coding techniques does not produce higher compression ratio; they simply reduce the complexity of a BAC

Context-based Implementation of BAC

- □ For context-based BAC, please adopt the PPM algorithm described in the textbook, with the maximal context order N equals 3.

Hand-in for the Homework

□ After encoding “jokes.txt” into a compressed sequence, please calculate the average bits per symbol (BPS), and compare it with the entropies estimated by the IID model, the 1st-order Markov model, and the Zip tool.

□ Please write a 4-page report to summarize and discuss your experiments

Optional Suggestions for your HW

- □ Can you find a better way to binarize the data source so that you get a smaller BPS?
- □ What are the differences of BPS when the maximal context orders are 3, 2, or 1?
- □ If you repeat the data source jokes.txt three times, would you see differences in BPS?