Context-based Binary Arithmetic Coding

國立陽明交通大學
NATIONAL YANG MING CHIAO TUNG UNIVERSITY

Chun-Jen Tsai
NYCU
5/1/2025

Homework Goal

- For this homework, you have to use context-based Binary Arithmetic Coding (BAC) to compress the same AlexNet model file of HW#1
  - You must upload the source to E3 for TA to check your result
  - You must write a short report on your discovery. Only PDF format, please.
  - In your report, please compare your results from HW#1 with the results in this HW
- Please upload your report to E3 by 5/27, 11:55pm.

Techniques You Can Consider

- BAC only allows two code symbols, 0 and 1. The binarization process that converts the original alphabet to binary code symbols may affect efficiency
  - You can begin with 1) natural binary code, or 2) unary code for 8-bit data binarization, then see if you can design other better binarization method to improve the performance
- For probability modeling, you can try:
  - Fixed probability model (none context-based)
  - PPM with max order of 2

About the Binarization Process

- Data source alphabet usually has more than 2 symbols
  - For example, a 8-bit data source has 256 symbols
- For BAC, the input data sequence are sequences of 0's and 1's $\rightarrow$ must convert data to binary before BAC
- A bad example of binarization:
  - $\Omega_x = \{ a, b, c \}$, p.m.f. $= \{ 0.1, 0.8, 0.1 \}$
  - If we use Gray code $\{ 00, 01, 11 \}$ to binarize a, b, c, the binarized distribution of 0 and 1 is uniform $\rightarrow$ bad for BAC
    - Context-based coding helps, but has its limitation
  - If unary code, a, b, c, are mapped to 10, 0, and 110 $\rightarrow$ very biased distribution of 0 and 1

Experiments You Should Try

- You should at least try the following four algorithms and compare their coding efficiency:
  - Natural binary code BAC + fixed probability model
  - Natural binary code BAC + PPM
  - Unary code BAC + fixed probability model
  - Unary code BAC + PPM
- In the textbook, the PPM algorithm has different design parameters to choose from, please discuss your implementation in detail in your report

Comments on Implementations

- You do not need to define an EOS symbol for AC since we already know the size of the data file
- For the fixed-pmf experiments, you can estimate the probability of each symbol from the data file offline
- You do not have to implement the decoder. Encoder alone is good enough to analyze its coding efficiency
  - However, if you have also implemented the encoder, make sure you highlight this in your report!
- We do not care about program execution speed.
  - But you should still include execution time in your report

Coding Instructions

- You must use C/C++ or Python for the HW, using open-source library is fine
  - You have to upload your source code in an archive file (in zip or tgz format).
  - You should put a readme file in the package with detail instructions about how to build and run your programs.
  - If the readme file is not written clearly enough for the TAs to reproduce your results, you will get 0 points for this HW!

Evaluation of the Homework

- Grading is based on your report
  - The report shall be in two-column format, single-space text, font size 10, with up to 4 pages. There is a report template (in both word and pdf format) on E3 for your reference
- Grading policy:
  - 50% on writing style and the completeness of your study
    - That is, the things you have tried
  - 50% on the difficulty of your implementation
    - The amount of code you write or modifications you made to open-source code
    - You MUST specify clearly which part of your code is from open-source projects