ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN

Môn học: Tính toán đa phương tiện

Học kỳ II (2019-2020)

ĐỀ TÀI

Nghiên cứu giải thuật Arithmetic Coding

Giảng viên: Thầy Nguyễn Vinh Tiệp

Lớp: CS232.K23.KHCL

Sinh viên thực hiện :

Hoàng Văn Hùng - 18520794

Trần Anh Khôi – 18520946

Lê Công Lực – 18521070

Tp.Hồ Chí Minh, tháng 7, 2020

Mục lục:

I. Giới thiệu thuật toán Arithmetic Coding và ngữ cảnh xuất hiện:

Phương pháp mã hóa số học(Arithmetic Coding) là một dạng mã hóa entropy được sử dụng trong nén bảo toàn dữ liệu. Arithmetic Coding là phương pháp hiệu quả nhất để mã hóa các ký hiệu theo xác suất xuất hiện của chúng.

Phương pháp này giống với mã hóa Huffman ở chỗ nó cũng dựa trên bảng chữ cái và tần số xuất hiện của từng chữ, nó cũng có thể áp dụng dạng động hoặc dạng tĩnh dựa vào việc thay đổi các khoảng tần số trong quá trình mã hóa.

+ Ngữ cảnh xuất hiện:

Giả sử rằng ta cần nén một thông tin T nằm trong bốn khả năng:

$$T \in \{a,b,c,d\}$$

Một phương pháp đơn giản để lưu trữ thông tin là ta sẽ dùng $\log_2(4)$ bits thông tin để lưu trữ T. Ta xây dựng một bảng dùng để encode và decode thông tin như sau:

Symbol	Code
a	00
b	01
c	10

Symbol	Code
d	11

Như vậy ta cần 2 bits để lưu trữ *T* trong cả 4 giá trị có thể nhận của nó.

Tuy nhiên, ta nhận thấy cách thức mã hóa như trên chỉ phù hợp khi xác suất *T* nhận một giá trị bất kì là bằng nhau cho mọi khả năng:

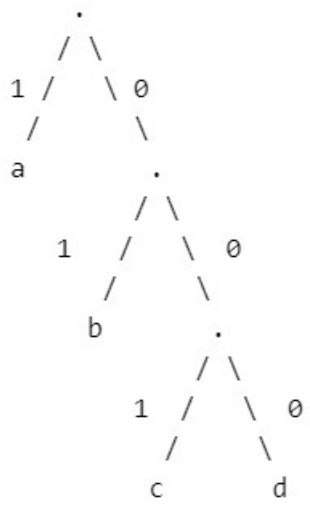
$$P(T=a)=P(T=b)=P(T=c)=P(T=d)=14$$

Nếu phân phối xác suất của a,b,c,d không đồng đều thì ta có một phương pháp tốt hơn. Giả dụ, ta có bảng phân phối xác suất như sau:

Symbol	**Probability**
a	0.9
b	0.05
c	0.025
d	0.025

Thuật toán tối ưu được dùng ở đây là Huffman Coding.

Sau khi sử dụng Huffman Coding, ta sẽ xây dựng được một cây nhị phân như sau:



Tuy nhiên một giới hạn của Huffman coding là nó chỉ hoạt động khi T nhận giá trị từ một tập hữu hạn các khả năng có thể. Trong thực tế, ta muốn nén một file có độ dài bất kỳ. Nói cách khác, ta muốn tìm một phương pháp nén mà T có thể nhận vô hạn các khả năng khác nhau.

Thậm chí nếu ta giới hạn T là một chuỗi hữu hạn 512 bits, thì Huffman Coding cần xây dựng một cây nhị phân có 2^{512} lá. Điều này là không khả thi trong thực tế.

Còn nếu ta áp dụng thuật toán Huffman Coding cho mỗi một chuỗi 5 bits, thì ứng với mỗi 5 bits này, ta có thể bị phung phí 1 bit so với giới hạn entropy của 5 bits này. Số lượng bits bị phung phí sẽ cộng dồn càng nhiều với một file có độ dài càng lớn.

Thêm vào đó, phân phối xác suất của bit hiện thời rất nhiều khả năng là phụ thuộc vào giá trị của các bits xuất hiện trước nó trong chuỗi bit. Cho nên ta cần xây dựng một phương pháp nén khai thác việc phụ thuộc này.

Ta xét ví dụ:

Xét một đồng xu X bị thiên vị:

$$P(X=1)=0.9$$

và

$$P(X=0)=0.1$$

Ta muốn lưu trữ một chuỗi 512 bits $T = x1,x512$ sao cho:

$$x\text{i} \sim P(X)$$

Nếu ta dùng Huffman Coding, thì ta cần xây dựng 1 cây nhị phân 2^{512} lá. Còn nếu ta dùng Huffman Coding cho mỗi xi , thì ta cần 512 bits để lưu trữ vì mỗi xi cần 1 bit để mã hóa.

Arithmetic Coding giải quyết bài toán trên bằng cách encode mỗi khả năng x của T, trong 2^{512} khả năng, bằng một khoảng $[a,b]\subset[0,1]$ sao cho độ dài của khoảng này b−a bằng chính xác suất của x .

Nói cách khác, ta chia khoảng số thực [0,1] ra thành 2^{512} khoảng nhỏ, sao cho độ dài mỗi khoảng là xác suất T nhận giá trị tương ứng.

II. Ý tưởng thuật toán và cải tiến:

Mục tiêu của Arithmetic Coding là tìm ra một khoảng duy nhất thể hiện một chuỗi ký tự có độ dài cố định, tiếp theo cần chọn trong khoảng này một số thập phân thích hợp, và coi đấy là mã biểu diễn cho chuỗi ký tự trên. Quá trình mã hóa được khởi tạo với khoảng ban đầu là [0, 1).

Arithmetic Coding thường có tỷ lệ nén tốt hơn phương pháp Huffman, vì nó tạo ra một mã hiệu duy nhất biểu diễn cho cả chuỗi ký tự thay vì một mã riêng biệt cho từng ký tự.

III. Chi tiết thuật toán và các bước thực hiện:

Thuật toán tổng quát:

Cho bảng chữ cái S = (s1 , s2 , … , sn) và xác suất xuất hiện từng ký tự là P = (p1 , p2 , … , pn).

Bước 1: Dựa vào xác suất xuất hiện để chia khoảng [0,1) thành n khoảng thể hiện cho n ký tự.

Bước 2: Mở rộng khoảng của ký tự đầu tiên được mã hóa. Chia khoảng này ra thành n khoảng thể hiện cho n ký tự.

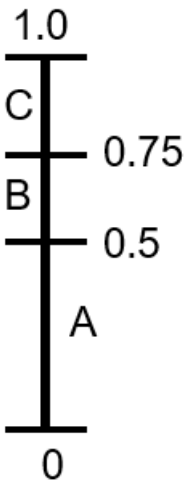
Bước 3: Tiếp tục thực hiện cho đến khi ký tự cuối cùng được mã hóa.

Ví dụ:

+Nén chuỗi "BACA" với xác suất xuất hiện của từng ký tự là P(A) = 0.5, P(B) = 0.25 và P(C) = 0.25.

Chia khoảng [0,1) thành các khoảng phụ của tất cả các kí tự đầu vào. Cho các khoảng phụ phù hợp với xác suất xuất hiện của các kí tự đầu vào:

$$A \in [0,0.5),\ B \in [0.5,0.75),\ C \in [0.75,1)$$



Mở rộng khoảng của ký tự đầu tiên trong chuỗi là "B". Tính D bằng công thức:

D = cận trên – cận dưới

sau đó tính lại khoảng cho từng ký tự bằng công thức:

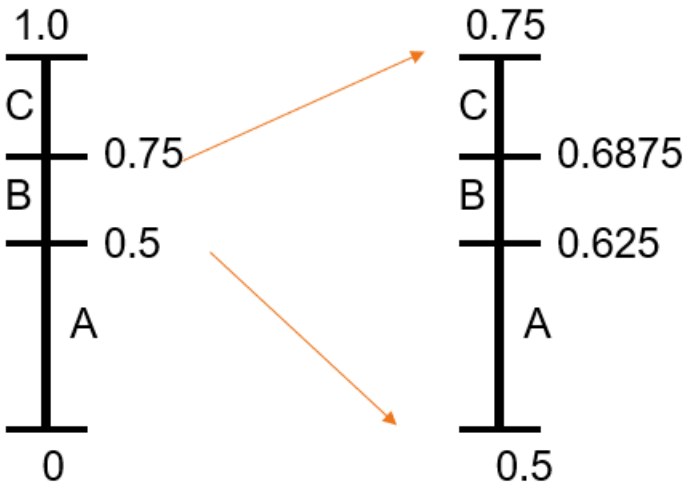
*Khoảng của "ký tự" = cận dưới -> cận dưới + D*P(ký tự)*

D(B) = 0.75 – 0.5 = 0.25.

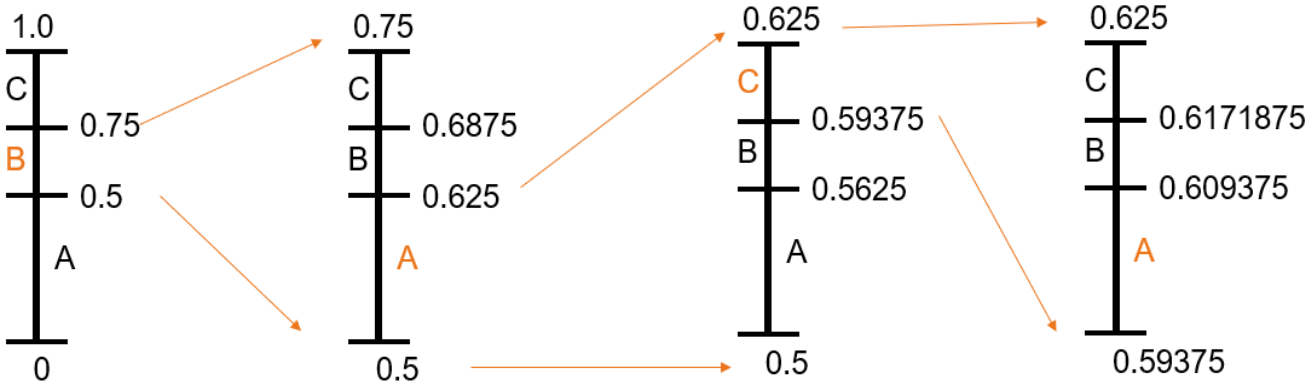
Khoảng của "A" = 0.5 -> 0.5 + 0.25*0.5 = 0.5 -> 0.625

Khoảng của "B" = 0.625 -> 0.625 + 0.25*0.25 = 0.625 -> 0.6875

Khoảng của "C" = 0.6875 -> 0.6875 + 0.25*0.25 = 0.6875 -> 0.75



Tiếp tục thực hiện các bước tương tự cho các ký tự còn lại.



Sau khi kết thúc ta sẽ được một khoảng mã hiệu biểu thị cho chuỗi vừa nén:

0.59375<mã hiệu<0.625

Chọn mã hiệu thể hiện chuỗi vừa nén:

Mã hiệu = (cận trên khoảng mã hiệu + cận dưới khoảng mã hiệu) / 2

= (0.625 + 0.59375) / 2 = **0.60937**

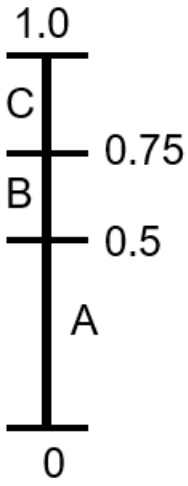
+Giải mã:

Đầu vào của quá trình giải mã là một mã hiệu và các xác suất xuất hiện của từng ký tự, đầu ra là chuỗi ký tự ban đầu, quá trình mã hóa và giải mã đối với thuật toán này là tương đồng.

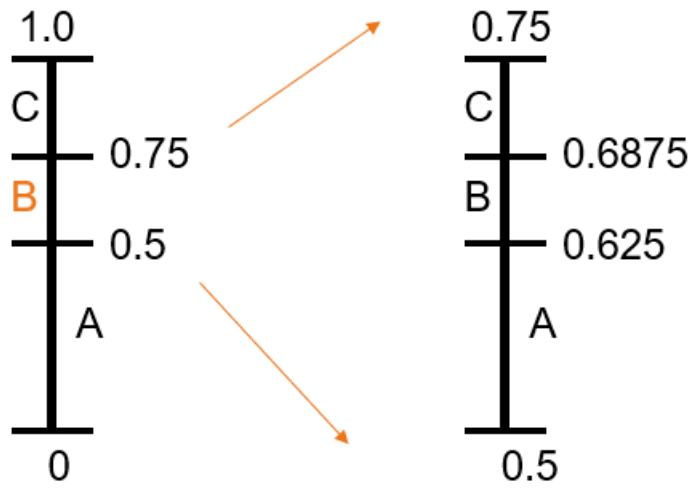
Ví dụ:

Giải nén mã hiệu 0.609375 với PA = 0.5, PB = 0.25 và PC = 0.25

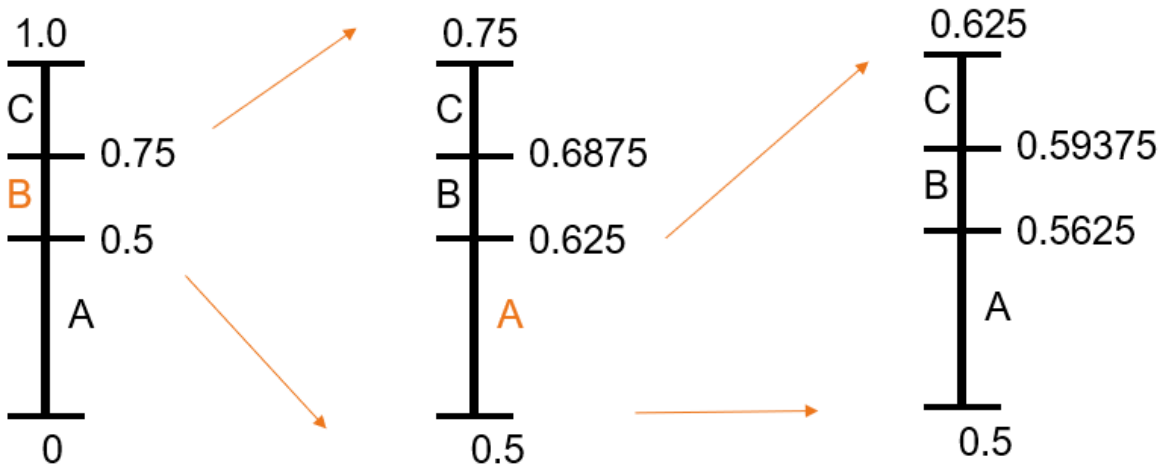
Chia khoảng [0,1) thành các khoảng phụ với xác suất xuất hiện của các kí tự đầu vào.



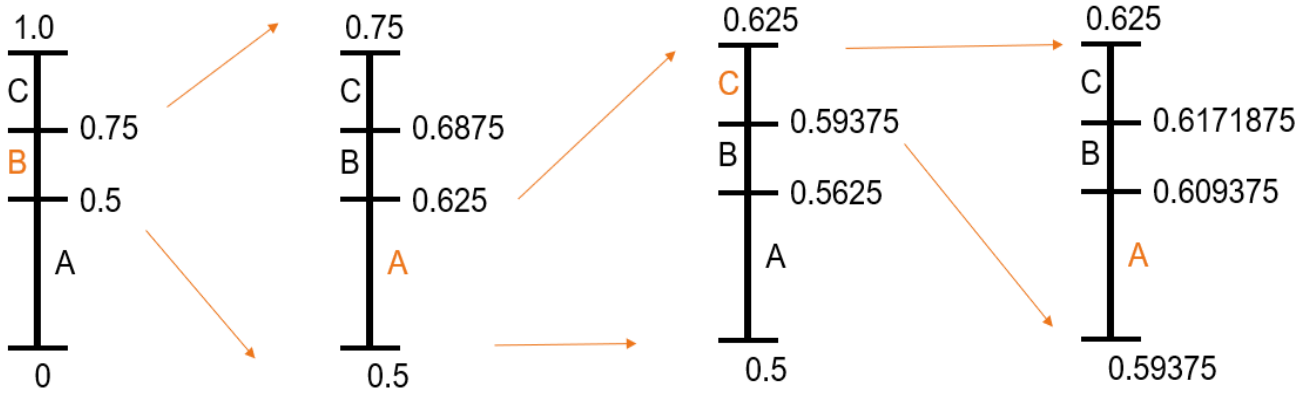
Xét mã hiệu 0.609375 nằm trong khoảng 0.5 -> 0.75 => thuộc khoảng ký tự "B". Mở rộng khoảng "B" và chia lại khoảng cho từng ký tự tương tự như bước mã hóa.



Tiếp tục xét mã hiệu 0.609375 nằm trong khoảng 0.5 -> 0.625 => thuộc khoảng ký tự "A". Mở rộng khoảng "A" và chia lại khoảng cho từng ký tự.



Tiếp tục thực hiện các bước trên cho đến khi ta giải mã được chuỗi ban đầu thì giải thuật giải mã dừng lại.



IV. Ưu, nhược điểm của Arithmetic Coding:

+ Ưu điểm:

- Là giải thuật nén không mất mát.

- Thường có hệ số nén tốt hơn so với các giải thuật nén entropy khác.

+ Nhược điểm:

- Cài đặt phức tạp hơn các giải thuật entropy khác.

- Dễ bị lỗi trong quá trình giải mã nếu tín hiệu mã hóa truyền vào bị sai lệch.

- Tồn tại nhiều bằng sáng chế về Arithmetic Coding nên việc sử dụng thuật toán có thể bị tính phí bản quyền.

V.Tài liệu tham khảo:

-Nén dữ liệu sử dụng phương pháp mã hóa số học – Hà Diệu Thúy – Đại học Thái Nguyên – Trường đại học CNTT & Truyền thông - 2013.

- Arithmetic Coding- A Reliable Implementation - Lakshmi Sasilal - Dr. V. K. Govindan - International Journal of Computer Applications - Volume 73– No.7, July 2013.