ĐẠI HỌC THÁI NGUYÊN
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

NGUYỄN THÀNH DƯƠNG

TÌM HIỂU MỘT SỐ THUẬT TOÁN MÃ HÓA VÀ NÉN DỮ LIỆU, XÂY DỰNG ỨNG DỤNG ĐỂ NÉN DỮ LIỆU ẢNH

LUẬN VĂN THẠC SỸ KHOA HỌC

Thái Nguyên - 2012

MỤC LỤC

PHẦN MỞ ĐẦU

Nén dữ liệu hiện đang được sử dụng hầu như ở mọi nơi. Tất cả các hình ảnh mà chúng ta xem hoặc sao chép được từ các trang web là các tệp hình ảnh đã được nén, thông thường trong định dạng JPEG hoặc GIF; đa số các modem đều sử dụng tính năng nén dữ liệu; truyền hình độ phân giải cao (HDTV) sử dụng phương pháp nén theo chuẩn MPEG-2. Một số hệ thống quản lý tệp tin tự động nén các tệp tin khi lưu trữ và chúng ta cũng thường sử dụng các chương trình nén khác nhau để nén tệp dữ liệu. Quá trình làm giảm kích thước của một tệp dữ liệu được gọi một cách phổ biến là *nén dữ liệu* (data compresion), còn tên gọi trong lý thuyết thông tin là *mã hóa nguồn* (source coding). Trong khoa học máy tính và lý thuyết thông tin, *nén dữ liệu* (hoặc *mã hóa nguồn*) là việc mã hóa thông tin bằng số ít bit hơn so với biểu diễn ban đầu.

Có thể chia các phương pháp nén ra hai lớp: *nén không mất thông tin* và *nén có mất thông tin*. Nén không mất thông tin làm giảm bit số bít biểu diễn bằng cách xác định và loại bỏ độ dư thừa thống kê trong cách biểu diễn ban đầu. Như tên gọi, thông tin không bị mất trong quá trình nén *không mất thông tin*. Nén có mất thông tin cố gắng giảm số bit bằng cách xác định thông tin không quan trọng và loại bỏ chúng. Nếu nói ngắn gọn về bản chất nén, đó là tập hợp các thuật toán, bao gồm từ phân loại, hàm băm, cho đến biến đổi Fourier nhanh (FFT), ... Ngoài ra các thuật toán dựa trên nền tảng lý thuyết vững chắc đóng một vai trò quan trọng trong các ứng dụng thực tế.

Nén dữ liệu hữu ích vì giúp giảm tài nguyên sử dụng như không gian lưu trữ dữ liệu hoặc dung lượng truyền. Vì dữ liệu nén phải được giải nén trước khi sử dụng, điều này đòi hỏi thêm chi phí tính toán để giải nén. Ví dụ, một chương trình nén cho video có thể yêu cầu phần cứng đắt tiền cho video được giải nén đủ nhanh để được xem như là nó đang được giải nén, và tùy chọn để giải nén video đầy đủ trước khi xem nó có thể là bất tiện hoặc yêu cầu lưu trữ bổ sung. Việc thiết kế các chương trình nén dữ liệu liên quan đến việc dung hòa các yếu tố khác nhau, bao

gồm cả mức độ nén, lượng thông tin bị mất khi sử dụng phương pháp nén dữ liệu có mất thông tin và các nguồn lực tính toán cần thiết để nén và giải nén dữ liệu.

Thuật ngữ tương đương *thông điệp, bản tin* hay *dãy tin* được sử dụng chung cho các đối tượng cần nén. Nhiệm vụ của nén dữ liệu bao gồm hai thành phần: một thuật toán *mã hóa* nhận bản tin ban đầu (mà ta gọi là *bản tin gốc*) và biểu diễn nó dưới dạng "nén" (hy vọng với ít bit hơn), và thuật toán *giải mã* được dùng để tái tạo lại bản tin ban đầu hoặc xấp xỉ của bản tin ban đầu từ bản tin đã được nén. Hai thành phần này thường được xây dựng gắn kết với nhau.

Nén không mất thông tin và *nén có mất thông tin:* Nén không mất thông tin thường được sử dụng cho văn bản và nén có mất thông tin thường được sử dụng để nén các tệp âm thanh và hình ảnh khi việc mất một số bit thông tin về độ phân giải thường là không thể phát hiện được hoặc ít nhất là chấp nhận được. Tuy nhiên nén có mất thông tin không có nghĩa là bị mất các pixel một cách ngẫu nhiên, thay vào đó có nghĩa là sự mất mát một đại lượng như một thành phần tần số, hoặc nhiễu. Chẳng hạn, người ta có thể nghĩ rằng nén văn bản có mất thông tin là không thể chấp nhận được bởi vì họ nghĩ tới việc mất hoặc chuyển đổi các ký tự. Thay vì đó ta có thể nghĩ tới một hệ thống các câu chuẩn, hoặc các từ thay thế bằng từ đồng nghĩa, nhờ đó có thể nén tập tin tốt hơn. Về mặt kỹ thuật nén mất dữ liệu có thể gây ra sự thay đổi của văn bản, nhưng ý nghĩa và tính rõ ràng của văn bản vẫn có thể được giữ nguyên hoặc thậm chí cải thiện.

Khi xét các thuật toán nén, điều quan trọng là cần phân biệt giữa hai thành phần: mô hình và bộ mã hóa. *Mô hình* cho biết phân phối xác suất của các dãy tin bằng cách nhận biết hoặc phát hiện cấu trúc của đầu vào. Bộ mã hóa tạo ra các dãy mã dựa trên các xác suất tạo ra mô hình. Để có hiệu quả nén, bộ mã hóa thường tạo ra các dãy mã dài cho các dãy tin có xác suất thấp và gán dãy mã ngắn cho các dãy tin có xác suất cao. Ví dụ, trong bảng chữ cái của một ngôn ngữ tự nhiên thường có một vài chữ cái xuất hiện trong các văn bản viết với xác suất cao hơn các chữ cái khác, điều này còn rõ ràng hơn với các cặp chữ cái. Khi đó bộ mã hóa sẽ gán từ mã có độ dài ngắn cho chữ cái xuất hiện với xác suất cao và ngược lại. Thông thường

sự tách biệt giữa mô hình và thành phần mã hóa không phải luôn luôn được xác định một cách rõ ràng.

Lý thuyết thông tin là lĩnh vực có thể gắn mô hình với thành phần mã hóa. Nó cho lý thuyết rất tốt sự liên quan giữa xác suất và độ dài từ mã. Lý thuyết này phù hợp với thực tế gần như hoàn hảo, và chúng ta có thể đạt được độ dài mã gần như giống hệt với những gì lý thuyết dự đoán.

Trong trường hợp mã hóa có mất thông tin, ta có thể lấy tiêu chuẩn đánh giá là thời gian nén, thời gian để tái tạo lại dãy tin ban đầu (giải mã) kích thước của tệp nén. Trong trường hợp nén có mất thông tin, các tiêu chuẩn thường là phức tạp hơn, chẳng hạn xấp xỉ dãy tin ban đầu như thế nào được gọi là chấp nhận được. Thông thường cần dung hòa giữa kích thước nén, thời gian chạy, và chất lượng dãy tin được giải mã.

Nội dung luận văn bao gồm 3 chương:

Chương 1: CƠ SỞ LÝ THUYẾT

Trình bày các khái niệm cơ bản, lý thuyết chung về mã hóa, nén dữ liệu, các định lý cơ bản về nén dữ liệu, lý thuyết về xử lý ảnh số.

Chương 2: MỘT SỐ THUẬT TOÁN MÃ HÓA VÀ NÉN DỮ LIỆU.

Chương này trình bày ý tưởng và các thuật toán mã hóa và nén dữ liệu như: RLE, HUFFMAN, JPEG, H.264/ACV, AIC.

Chương 3: XÂY DỰNG ỨNG DỤNG THỬ NGHIỆM

Chương này trình bày các kết quả cài đặt và chạy thử nghiệm của các thuật toán mã hóa và nén dữ liệu như: RLE, HUFFMAN, JPEG. Các kết quả so sánh với các phần mềm hiện có.

Chương 1
CƠ SỞ LÝ THUYẾT

1.1. Mã hóa thông tin

Để tìm hiểu về mã hóa thông tin, ta bắt đầu từ những khái niệm cơ bản sau:

Bảng chữ cái: Bảng chữ cái là tập bất kỳ hữu hạn các phần tử, khác rỗng. Mỗi phần tử của bảng chữ cái gọi là kí tự.

Bản tin: Cho bản chữ cái A = $\{a_1,a_2,...a_n\}$, dãy X gồm các kí tự của A gọi là bản tin. Bản tin theo nghĩa rộng nó có thể là bức ảnh, có thể là băng ghi âm thanh v.v…, tuy nhiên khi thực hiện số hóa để lưu trữ hay truyền đi vẫn phải sử dụng bảng chữ cái nào đó.

Mã hoá: Giả sử có bảng chữ cái A = $\{a_1,a_2,...,a_n\}$, X là một bản tin trên bảng chữ cái A. Ta gọi bản tin Y trên bảng chữ cái B = $\{b_1,b_2,...,b_m\}$ là bản mã của bản tin X nếu tồn tại ánh xạ f sao cho Y = f(X). Khi đó f được gọi là phép mã hóa.

Cách ghi mã: Có nhiều cách ghi mã, giả sử mã văn bản người ta hay sử dụng những nhóm ký hiệu được phân cách bởi một dấu *Space*, cách mã như vậy gọi là mã bằng phương pháp từ. Mã chỉ sử dụng hai ký tự "0" và "1" để biểu diễn gọi là mã nhị phân. Loại mã dùng ký hiệu bằng một nhóm ký tự có độ dài nhất định cho mỗi từ mã là mã có độ dài cố định. Loại mã này ta luôn giải mã được. Nhưng nếu lưu trữ như vậy sẽ rất tốn kém, nên người ta thường dựa vào tần suất xuất hiện các chữ cái để mã, với tần suất càng nhiều mã càng ngắn. Mã như vậy gọi là mã có độ dài thay đổi. Tuy nhiên nếu độ dài của từ mã thay đổi thì không phải với ánh xạ mã nào cũng có thể giải mã được.

Xét ví dụ ánh xạ mã: $a \to 100$; $b \to 1000$; $c \to 0$

Mã của "ac" và "b" đều là dãy bit "1000". Như vậy khi nhận được chuỗi bit 1000 chúng ta không thể biết được rằng văn bản ban đầu là "b" hay là "ac". Cho nên khi mã hoá sử dụng mã có độ dài thay đổi cần có tính chất là giải mã được, đó là tính phân tách. Tính phân tách được đưa ra dưới đây sẽ đảm bảo cho tính giải được của mã.

Xét A và B là hai đoạn mã tạo ra từ các bít 0/1. Ta nói A là đầu của B nếu như có một đoạn C sao cho B = A + C. Một tập hợp M tạo ra được gọi là phân tách nếu không có đoạn nào là đầu của đoạn khác.

Như vậy, mã có độ dài từ mã cố định là mã phân tách.

1.2. Nén dữ liệu

Dữ liệu: Giả sử có bảng chữ cái A = $\{x_1,x_2,...x_n\}$, X là một bản tin trên bảng chữ cái A. Ta gọi bản tin Y trên bảng chữ cái nhị phân B = $\{0,1\}$ là bản mã của bản tin X, nếu tồn tại ánh xạ f sao cho Y = f(X). Khi đó Y được gọi là dữ liệu của bản tin X.

Nén dữ liệu: Ta kí hiệu L(Y) là số bít của bản tin Y. Giả sử $L_f(Y)$ là dung tích dữ liệu của bản tin X với phép mã hóa f, việc tìm phép mã hóa g sao cho $L_g(Y) \leq L_f(Y)$ gọi nén dữ liệu.

Từ các khái niệm, định nghĩa nêu trên chúng ta dễ dàng nhận ra bản chất của việc nén dữ liệu là đi tìm phép mã hóa bản tin sao cho dung tích dữ liệu của nó càng nhỏ càng tốt. Một file dữ liệu không thể nén đến bao nhiêu tuỳ ý vẫn cần đảm bảo sự tồn tại của dữ liệu đó. Một file dữ liệu chỉ có thể nén đến một giới hạn nhất định, giới hạn ấy gọi *Entropy*. *Entropy* chỉ phụ thuộc vào dữ liệu, không phụ thuộc vào thuật toán.

1.3. Entropy

* Độ đo Logarit của thông tin: Giả sử có hai biến ngẫu nhiên X và Y; X có thể nhận các giá trị trong tập $\{x_1,x_2,...,x_n\}$ và Y có thể nhận giá trị trong tập $\{y_1,y_2,...,y_m\}$. Chúng ta cần xác định về mặt định lượng thông tin của sự kiện $X = x_i$ khi đã biết $Y = y_j$. Rõ ràng là nếu X và Y là hai biến độc lập thì việc biết trước $Y = y_j$ thì không cho lượng thông tin nào về việc xảy ra $X = x_i$. Mặt khác nếu X và Y phụ thuộc nhau đầy đủ thì khi $Y = y_j$ xác định được $X = x_i$ thì nội dung thông tin (Information Content) đơn giản được cho bởi $X = x_i$. Khi đó thông tin có được về việc xảy ra sự kiện $X = x_i$ nhờ đã xảy ra sự kiện $Y = y_j$ được tính bằng:

$$I(x_i, y_j) = \log \frac{p(x_i / y_j)}{p(x_i)} \qquad 1.1$$

ở đây kí hiệu: $p(x_i|y_j) = p(X=x_i|Y=y_j)$ và $p(X=x_i) = p(x_i)$

Trong đó: $I(x_i,y_j)$ số đo thông tin liên quan giữa x_i và y_j.

Đơn vị của $I(x_i,y_j)$ được xác định bởi cơ số của Logarit người ta thường lấy là 2 hoặc *e*, nếu cơ số là 2 ta gọi đơn vị của I là bit, nếu là *e* ta gọi là đơn vị tự nhiên. Công thức chuyển đổi giữa các đơn vị là :

$$\mathbf{lna = ln2log_2a = 0{,}69315log_2a} \qquad 1.2$$

Trường hợp X và Y là hai biến độc lập thì $p(x_i|y_j) = p(x_i)$ khi đó từ công thức *1.1* suy ra $I(x_i,y_j) = 0$. Khi sự kiện $Y = y_j$ xảy ra, mà chắn chắn sự kiện $X = x_i$ xảy ra thì: $p(x_i|y_j) = 1$. Khi đó công thức *1.1* có dạng:

$$\mathbf{I(x_i,y_j) = log\ 1/p(x_i)\ hay\ I(x_i,y_j) = -\ log\ p(x_i)} \qquad 1.3$$

Công thức *1.3* chính là thông tin của sự kiện $X = x_i$, có thể viết công thức *1.3* ở dạng:

$$\mathbf{I(x_i) = -\ log\ p(x_i)} \qquad 1.4$$

Cần chú ý rằng từ *1.4* suy ra sự kiện có xác suất càng cao thì lượng thông tin mang lại ít hơn sự kiện có xác suất thấp. Rõ ràng với sự kiện x bất kỳ mà $p(x) = 1$ thì $I(x) = 0$, nghĩa là việc xảy ra sự kiện x không mang lại lượng thông tin nào.

Xét ví dụ sau: Giả sử có nguồn rời rạc phát đi các bit 0, 1 với xác suất bằng nhau bằng 1/2 trong t giây thông tin đưa ra từ nguồn là:

$I(x_i) = -\log_2 1/2 = 1$ bit, *ở đây $x_i = 0$ hoặc $x_i = 1$*

Hoặc ví dụ khác: Giả sử xét mô hình thống kê độc lập. Xét dãy k bít của nguồn phát đi, rõ ràng có tất cả $M = 2^k$ dãy k bit khác nhau do vậy các dãy này có xác suất xuất hiện bằng nhau và bằng $1/2^k$. Khi đó:

$I(x_i) = -\log_2 1/2^k = k$ bit trong khoảng thời gian k.t

Như vậy có thể thấy độ đo Logarit của thông tin có tính chất cộng khi ta coi đầu ra của nguồn ra là một dãy.

Bây giờ chúng ta chú ý tới đẳng thức sau:

$$p(x_i|y_j)/p(x_i) = p(x_i|y_j)p(y_j)/p(x_i)p(y_j)$$
$$= p(x_i,y_j)/p(x_i)p(y_j)$$
$$= p(y_j|x_i)/p(y_j)$$

Từ đây suy ra: $\mathbf{I(x_i,y_j) = I(y_j,x_i)}$ *1.5*

Như vậy thông tin về sự kiện $X = x_i$ khi xảy ra sự kiện $Y = y_j$ đã xảy ra bằng thông tin về sự kiện $Y = y_j$ khi sự kiện $X = x_i$ đã xảy ra.

Ngoài ra từ định nghĩa thông tin phụ thuộc lẫn nhau (Mutual Information) và thông tin độc lập (Self Information) được dùng để xác định thông tin có điều kiện (Condition self - Information)

$$\mathbf{I(x_i|y_j) = \log [1/ p(x_i|y_j)] = -\log p(x_i|y_j)} \quad 1.6$$

Kết hợp các đẳng thức *1.1* và *1.4* ta có:

$$\mathbf{I(x_i,y_j) = I(x_i) - I(x_i|y_j)} \quad 1.7$$

Từ *1.7* có thể suy ra thông tin phụ thuộc lẫn nhau giữa các cặp sự kiện có thể dương, bằng 0 hoặc âm.

Trung bình của thông tin phụ thuộc: Từ định nghĩa thông tin phụ thuộc lẫn nhau của các cặp sự kiện (x_i, y_j) của hai biến ngẫu nhiên X và Y, khi đó ta có thể nhận được giá trị trung bình của thông tin phụ thuộc của hai biến ngẫu nhiên X, Y có dạng:

$$I(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)I(x_i,y_j) \quad 1.8$$

Hay : $$I(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i,y_j)\frac{P(x_i,y_j)}{p(x_i)p(y_j)} \quad 1.9$$

Ta thấy I(X,Y) = 0 khi X và Y độc lập, vậy một đặc trưng quan trọng của I(X,Y) là $I(X,Y) \geq 0$. Tương tự như vậy chúng ta định nghĩa thông tin trung bình:

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) = -\sum_{i=1}^{n} P(x_i)\log P(x_i) \quad 1.10$$

Khi X là bảng chữ cái bao gồm các kí tự sinh ra nguồn, khi đó H(X) là trung bình thông tin trên các kí tự.