Cơ sở Lý thuyết Truyền tin-2004

Hà Quốc Trung[1]

[1]Khoa Công nghệ thông tin
Đại học Bách khoa Hà nội

Chương 5: Mã hóa nguồn

1. Mã hóa nguồn rời rạc không nhớ
2. Mã hóa cho nguồn dừng rời rạc
3. Cơ sở lý thuyết mã hóa nguồn liên tục
4. Các kỹ thuật mã hóa nguồn liên tục

Khái niệm chung

- Là phép biến đổi đầu tiên cho nguồn tin nguyên thủy
- Đầu vào của phép biến đổi này có thể là: nguồn tin rời rạc hoặc nguồn tin liên tục
- Trong cả hai trường hợp mục đích chính của phép mã hóa nguồn là biểu diễn thông tin với tài nguyên tối thiểu
- Các vấn đề cần nghiên cứu
  - Mã hóa nguồn rời rạc
  - Mã hóa nguồn liên tục
  - Nén dữ liệu

1.2.Mã hóa nguồn

- Nguồn thông tin tạo ra các đầu ra một cách ngẫu nhiên
- Nguồn rời rạc: tạo ra một chuỗi các ký hiệu ngẫu nhiên
  - Nguồn không nhớ: các ký hiệu xuất hiện một cách độc lập với nhau
  - Nguồn có nhớ: các ký hiện xuất hiện phụ thuộc vào các ký hiệu đã xuất hiện trước đo
  - Nguồn dừng các mối liên hệ thống kê giữa các thời điểm không phụ thuộc vào thời gian
- Với nguồn rời rạc, vấn đề cơ bản là thay đổi bảng chữ cái và phân bố xác suất để giảm bớt số lượng ký hiệu cần dùng
- Nguồn liên tục tạo ra một tín hiệu, một thể hiện của một quá trình ngẫu nhiên
  - Nguồn liên tục có thể được biến thành một chuỗi các biến ngẫu nhiên (liên tục) bằng phép lấy mẫu
  - Lượng tử hóa cho phép biến đổi các biến ngẫu nhiên này thành các biến ngẫu nhiên rời rạc, với sai số nhất định
  - Các kỹ thuật mã hóa nguồn tương tự

2. Mã hóa nguồn rời rạc không nhớ

1. Mã hóa nguồn rời rạc không nhớ
   - Mô hình toán học nguồn thông tin
   - Mã hóa với từ mã có độ dài cố định
   - Mã hóa với từ mã có độ dài thay đổi
2. Mã hóa cho nguồn dừng rời rạc
3. Cơ sở lý thuyết mã hóa nguồn liên tục
4. Các kỹ thuật mã hóa nguồn liên tục

Mô hình toán học nguồn rời rạc

- Với nguồn rời rạc cần quan tâm
  - Entropy của nguồn tin nguyên thủy
  - Entropy của nguồn sau khi mã hóa
  - Hiệu quả của phép mã hóa
  - Giới hạn của hiệu quả mã hóa
  - Xét một nguồn rời rạc không nhớ, sau một thời gian t_s tạo ra ký hiệu x_i trong L ký hiệu với các xác suất xuất hiện là $P(i)$
  - Để cho đơn giản, chỉ xét trường hợp mã hiệu nhị phân. Khi đó: lượng tin=lượng bít= số ký hiệu nhị phân
  - Với mã hiệu có cơ số lớn hơn 2, có thể mở rộng các kết quả thu được.

2.2.Mã hóa với từ mã có độ dài cố định

- Nguyên tắc: Mã hóa một ký hiệu nguồn thành một chuỗi ký hiệu mã có độ dài xác định R
- Để đảm bảo phép mã hóa là 1-1, một ký hiệu nguồn tương ứng với 1 chuỗi ký hiệu nhị phân. Số lượng chuỗi nhị phân phải lớn hơn số ký hiệu nguồn

$$2^R \geq L \text{ hay } R \geq \log_2 L$$

- Nếu L là lũy thừa của 2 thì giá trị nhỏ nhất của R là $\log_2 L$
- Nếu L không là lũy thừa của 2, giá trị đó là $\lfloor \log_2 L \rfloor + 1$
- Như vậy

$$R \geq H(X)$$

. Hiệu suất của phép mã hóa $\frac{H(X)}{R} \leq 1$

- Tốc độ lập tin đầu ra sẽ lớn hơn tốc độ lập tin đầu vào

Tăng hiệu quả mã hóa

- Hiệu quả mã hóa đạt giá trị cực đại khi
  - L là lũy thừa của 2
  - Nguồn tin ban đầu đẳng xác suất
- Nếu nguồn tin ban đầu đẳng xác suất, nhưng L không là lũy thừa của 2, số lượng ký hiệu nhỏ nhất sẽ là $\lfloor H(X) \rfloor + 1$. Hiệu quả của nguồn là

$$\frac{H(X)}{\lfloor H(X) \rfloor + 1} \geq \frac{H(X)}{H(X) + 1}$$

- Để tăng hiệu quả, cần tăng lượng tin cho mỗi lần mã hóa: mã hóa cùng một lúc J ký hiệu. Hiệu quả mã hóa

$$\frac{JH(X)}{\lfloor JH(X) \rfloor + 1} \geq \frac{JH(X)}{JH(X) + 1}$$

Biểu thức trên tiến tới 1 khi J tiến tới vô cùng

- Kết quả này chỉ đúng cho nguồn đẳng xác suất.
- Phép mã hóa không có sai số, mỗi chuỗi ký hiệu nguồn luôn luôn tương ứng với 1 từ mã duy nhất.

Tăng hiệu quả bằng mã hóa có sai số

- Trong trường hợp nguồn không đẳng xác suất, để có thể tiệm cận với hiệu quả tối đa (1), cần chấp nhận một sai số nào đó
- Xét L^J chuỗi ký hiệu nguồn có độ dài J, mã hóa bằng chuỗi các ký hiệu nhị phân có độ dài $R, 2^R < L^J$
- Như vậy còn $L^J - 2^R$ tổ hợp ký hiệu nguồn không có từ mã tương ứng
- Sử dụng $2^R - 1$ từ mã mã hóa $2 - 1$ chuỗi ký hiệu nguồn
- Các chuỗi ký hiệu nguồn còn lại (chọn các chuỗi có xác suất nhỏ nhất), được mã hóa bằng 1 từ mã chung
- Nếu nguồn phát một chuỗi các ký hiệu trùng với các chuỗi ký hiệu có xác suất thấp, sẽ có sai số. Gọi xác suất sai số là P_e
- Liên quan giữa P_e, R, J?

Định lý mã hóa nguồn 01

Theorem

- *Cho U là một nguồn tin có Entropy hữu hạn. Mã hóa các khối J ký hiệu của nguồn thành các từ mã N ký hiệu nhị phân. ϵ là một số dương bất kỳ*
- *Xác suất sai số có thể nhỏ tùy ý nếu*

$$R = \frac{N}{J} \geq H(U) + \epsilon$$

- *Ngược lại, nếu*

$$R = \frac{N}{J} \leq H(U) - \epsilon$$

thì sai số sẽ tiến tới 1 khi J tiến tới vô hạn

- Tốc độ lập tin của đầu ra luôn luôn lớn hơn của đầu vào

Chứng minh định lý

Chứng minh.

- Phần thuận
  Coi tập hợp các chuỗi ký hiệu nguồn mà

$$|\frac{I(u_J)}{J} - H(U)| \geq \epsilon$$

  là các chuỗi ký hiệu nguồn ánh xạ vào cùng một từ mã.
  Cần chứng minh
  1. Xác suất xuất hiện của các từ mã nói trên có thể bé tùy ý khi L lớn tùy ý (hiển nhiên, $\lim_{J\to\infty} \frac{I(u_J)}{J} = H(U)$)
  2. Các chuỗi ký hiệu còn lại có thể được mã hóa chính xác (1-1) với $R = \frac{N}{J} \geq H(X) + \epsilon$
- Phần đảo: Chứng minh xác suất sai số tiến đến 1 (?)

□

Chứng minh định lý

Chứng minh.

- Phần thuận
  Coi tập hợp các chuỗi ký hiệu nguồn mà

$$|\frac{I(u_J)}{J} - H(U)| \geq \epsilon$$

  là các chuỗi ký hiệu nguồn ánh xạ vào cùng một từ mã.
  Cần chứng minh
  1. Xác suất xuất hiện của các từ mã nói trên có thể bé tùy ý khi L lớn tùy ý (hiển nhiên, $\lim_{J\to\infty} \frac{I(u_J)}{J} = H(U)$)
  2. Các chuỗi ký hiệu còn lại có thể được mã hóa chính xác (1-1) với $R = \frac{N}{J} \geq H(X) + \epsilon$
- Phần đảo: Chứng minh xác suất sai số tiến đến 1 (?)

□

Chứng minh định lý

Chứng minh.

- Phần thuận
  Coi tập hợp các chuỗi ký hiệu nguồn mà

$$|\frac{I(u_J)}{J} - H(U)| \geq \epsilon$$

  là các chuỗi ký hiệu nguồn ánh xạ vào cùng một từ mã.
  Cần chứng minh
  1. Xác suất xuất hiện của các từ mã nói trên có thể bé tùy ý khi L lớn tùy ý (hiển nhiên, $\lim_{J\to\infty} \frac{I(u_J)}{J} = H(U)$)
  2. Các chuỗi ký hiệu còn lại có thể được mã hóa chính xác (1-1) với $R = \frac{N}{J} \geq H(X) + \epsilon$
- Phần đảo: Chứng minh xác suất sai số tiến đến 1 (?)

□

Chứng minh định lý

Chứng minh.

- Phần thuận
  Coi tập hợp các chuỗi ký hiệu nguồn mà

$$|\frac{I(u_J)}{J} - H(U)| \geq \epsilon$$

  là các chuỗi ký hiệu nguồn ánh xạ vào cùng một từ mã.
  Cần chứng minh
  1. Xác suất xuất hiện của các từ mã nói trên có thể bé tùy ý khi L lớn tùy ý (hiển nhiên, $\lim_{J\to\infty} \frac{I(u_J)}{J} = H(U)$)
  2. Các chuỗi ký hiệu còn lại có thể được mã hóa chính xác (1-1) với $R = \frac{N}{J} \geq H(X) + \epsilon$
- Phần đảo: Chứng minh xác suất sai số tiến đến 1 (?)

Chứng minh định lý

Chứng minh.

- Phần thuận
  Coi tập hợp các chuỗi ký hiệu nguồn mà

$$|\frac{I(u_J)}{J} - H(U)| \geq \epsilon$$

  là các chuỗi ký hiệu nguồn ánh xạ vào cùng một từ mã.
  Cần chứng minh
  1. Xác suất xuất hiện của các từ mã nói trên có thể bé tùy ý khi L lớn tùy ý (hiển nhiên, $\lim_{J\to\infty} \frac{I(u_J)}{J} = H(U)$)
  2. Các chuỗi ký hiệu còn lại có thể được mã hóa chính xác (1-1) với $R = \frac{N}{J} \geq H(X) + \epsilon$
- Phần đảo: Chứng minh xác suất sai số tiến đến 1 (?)

Chứng minh phần thuận

- Gọi tập hợp các ký hiệu còn lại là T. Với mỗi $u_J \in T$ có

$$\frac{I(u_J)}{J} - H(U) \leq \epsilon$$

$$H(U) - \epsilon \leq \frac{I(u_J)}{J} \leq H(U) + \epsilon$$

$$2^{-J(H(U)-\epsilon)} \geq P(u_J) \geq 2^{-J(H(U)+\epsilon)}$$

Chú ý

$$1 \geq P(T) \geq M_T min(P(u_J)) \geq M_T 2^{-J(H(U)+\epsilon)}$$

Có

$$M_T \leq 2^{J(H(U)+\epsilon)}$$

Vậy nếu chọn chuỗi nhị phân có độ dài tối thiểu là

$$N_min = \log_2 2^{J(H(U)+\epsilon)} = J(H(U) + \epsilon)$$

sẽ có ánh xạ 1-1 giữa T và tập các từ mã N ký hiệu nhị phân Phép ánh xạ chung sẽ có sai số nhỏ tùy ý

$P_e = |\frac{I(u_J)}{J} - H(U)| \geq \epsilon$

Chứng minh phần đảo

- Chọn $N \leq J(H(U) - 2\epsilon)$. Xét một phép mã hóa bất kỳ

$$P(T) + P(\overline{T}) + P_e = 1$$

Trong đó

- $P(T)$ là xác suất để mỗi một chuỗi ký hiệu trong T có một từ mã
- $P(\overline{T})$ là xác suất để một chuỗi ký hiệu ngoài T có một từ mã
- Xác suất lỗi (tồn tại chuỗi ký hiệu không có từ mã)

Tổng cộng có 2^N từ mã, mỗi từ mã sẽ tương ứng với một từ trong T có xác suất nhỏ hơn $2^{-J(H(U)-\epsilon)}$, vậy xác suất để một từ trong T có một từ mã là

$$P(T) = 2^{-J(H(U)-\epsilon)}2^N \leq 2^{-J(H(U)-\epsilon)}2^{-J(H(U)-2\epsilon)} = 2^{-J\epsilon}$$

Chú ý $P(\overline{T})$ tiến tới 0 khi j tiến tới vô cùng. Vậy P_e tiến tới 1

Ý nghĩa định lý

- Phép mã hóa với từ mã có độ dài không đổi nói chung bảo toàn độ bất định của nguồn
- H(U) là số ký hiệu nhị phân nhỏ nhất có thể sử dụng để biểu diễn nguồn tin nguyên thủy một cách chính xác
- Trong trường hợp tổng quát, số ký hiệu nhỏ nhất đó có thể đạt được khi mã hóa một khối có chiều dài vô tận các ký hiệu nguồn
- Định lý có thể mở rộng cho mã hiệu cơ số lớn hơn 2.

2.3.Mã hóa với từ mã có độ dài thay đổi

- Mục tiêu: mã hóa ký hiệu với số lượng ký hiệu nhị phân tối thiểu
- Xét trường hợp nguồn có phân bố xác suất không đều
- Các ký hiệu nguồn có xác suất xuất hiện lớn cần được mã hóa với các từ mã có độ dài nhỏ và ngược lại. Số ký hiệu trung bình cho mỗi ký hiệu của nguồn:

$$\overline{R} = \sum_{1}^{L} n_k P(u_k)$$

sẽ có giá trị tối ưu

- Mã hiệu sử dụng trong trường hợp này cần có tính prefix (giải mã được) được thể hiện bằng bất đẳng thức Kraft (McMillan)

2.3.Mã hóa với từ mã có độ dài thay đổi

Theorem

Điều kiện cần và đủ để tồn tại một mã hiệu nhị phân có tính prefix với các từ mã có độ dài $n_1 \leq n_2 \leq \ldots \leq n_L$ *là*

$$\sum_{k=1}^{L} 2^{-n_k} \leq 1$$

Chứng minh phần thuận

- Xây dựng một cây mã nhị phân có $2^n, n = n_L$ nút cuối
- Chọn một nút bậc n_1. Đường dẫn tới nút đó lấy làm từ mã. Toàn bộ cây con trên nút đó coi là đã sử dụng (gồm 2^{n-n_1} nút cuối)
- Tiếp tục chọn một nút ở mức n_2. Loại bỏ toàn bộ cây con của nút đó (gồm 2^{n-n_1} nút cuối).
- Nếu vẫn còn nút cuối chưa sử dụng, còn có thể chọn được một nút ở mức bất kỳ
- Khi chọn nút n_j số lượng các nút đã sử dụng là

$$\sum_{k=1}^{L} 2^{n-n_k} = 2^n \sum_{k=1}^{L} 2^{-n_k} \leq 2^n$$

Vậy luôn luôn có thể chọn được một nút cho đến khi $n_j > n = n_L$. Các từ mã tương ứng sẽ tạo ra một mã hiệu có tính prefix.

Chứng minh phần đảo

- Biểu diễn mã hiệu prefix bằng cây nhị phân.
- Mỗi một từ mã tương ứng với một nút
- Không có từ mã nào nằm trong cây con của từ mã nào
- Hai cây con của hai từ mã bất kỳ rời nhau
- Tính số lượng các nút cuối thuộc về cây con của mỗi từ mã 2^{n-n_j}
- Tính tổng các nút thuộc về các cây con, có bất đẳng thức Kraft

$$\sum_{k=1}^{L} 2^{n-n_k} \leq 2^n \text{ hay } \sum_{k=1}^{L} 2^{-n_k} \leq 1$$

Định lý mã hóa nguồn 2

Theorem

Cho X là một nguồn rời rạc không nhớ. Có thể mã hóa nguồn X bằng một mã hiệu nhị phân không đều, có tính prefix và có độ dài trung bình $\overline{R}$ của các từ mã thỏa mãn điều kiện

$$H(X) \leq \overline{R} < H(X) + 1$$

Chứng minh cận dưới

- Có

$$H(X) - \overline{R} = \sum_{k=1}^{L} p_k \log_2 \frac{1}{p_k} - \sum_{k=1}^{L} p_k n_k = \sum_{k=1}^{L} p_k \log_2 \frac{2^{-n_k}}{p_k}$$

- Sử dụng bất đẳng thức $\ln x \leq x - 1$ và bất đẳng thức Kraft

$$H(X) - \overline{R} \leq (\log_2 e) \sum_{k=1}^{L} p_k \left(\frac{2^{-n_k}}{p_k} - 1\right) (\log_2 e) \left(\sum_{k=1}^{L} 2^{-n_k} - 1\right) \leq 0$$

- Dấu bằng xảy ra khi $p_k = 2^{-n_k} \forall 1 \leq k \leq L$

Chứng minh cận trên

- Cần tìm một mã hiệu sao cho $\overline{R} < H(X) + 1$
- Chọn n_k sao cho $2^{-n_k} \leq p_k < 2^{-n_k+1}$. Có $n_k < 1 - \log_2 p_k$. Vậy

$$\sum_{k=1}^{L} p_k n_k \leq \sum_{k=1}^{L} p_k (1 - \log_2 p_k) = 1 + H(X)$$

Mã hóa Shannon-Fano

- Nguyên tắc: độ dài từ mã tỷ lệ nghịch với xác suất xuất hiện
- Cách thức (Fano):
  1. Chia các ký hiệu nguồn thành m nhóm (nếu mã nhị phân: 2 nhóm), xác suất xấp xỉ như nhau (làm thế nào để chia?)
  2. Gán cho mỗi nhóm một ký hiệu 0 hoặc 1
  3. Thực hiện 1 cho đến khi mỗi nhóm chỉ còn 1 ký hiệu
- Cách thức (Shanon)
  1. Sắp xếp các ký hiệu nguồn theo thứ tự giảm dần của xác suất
  2. Với mỗi ký hiệu
     1. tính tổng các xác suất của các ký hiệu đứng trước
     2. Biểu diễn tổng thu được theo hệ nhị phân, độ chính xác là xác suất của ký hiệu
     3. Từ mã tương ứng là chuỗi chữ số phần lẻ của biểu diễn trên
- Kết quả: Bộ mã thu được có tính prefix
- Có thể biểu diễn quá trình bằng một cây

Ví dụ (Shanon)

Lập mã cho nguồn có phân bố xác suất

Ký hiệu	u_1	u_2	u_3	u_4	u_5	u_6	u_7
Xác suất	0.34	0.23	0.19	0.1	0.07	0.06	0.01

Lập bảng

u_i	p_i	P_i	Nhị phân	n_i	Từ mã
u_1	0.34	0	0.0	2	00
u_2	0.23	0.34	0.01	3	010
u_3	0.19	0.57	0.1001	3	100
u_4	0.1	0.76	0.1100	4	1100
u_5	0.07	0.86	0.11011	4	1101
u_6	0.06	0.93	0.11101	5	11101
u_7	0.01	0.99	0.1111110	7	1111110

Entropy của nguồn 2.3828

Số ký hiệu nhị phân trung bình 2.99

Hiệu quả của nguồn: 0.7969

Ví dụ (Fano)

u_i	p_i	P_i	Nhị phân	n_i	Từ mã		
u_1	0.34	0	0	-	-	-	00
u_2	0.23	0	1	-	-	-	01
u_3	0.19	1	0	-	-	-	10
u_4	0.1	1	1	0	-	-	110
u_5	0.07	1	1	1	0	-	1110
u_6	0.06	1	1	1	1	0	11110
u_7	0.01	1	1	1	1	1	11111

Mã hóa Shannon-Fano

- Có thể có nhiều mã hiệu thích hợp, phụ thuộc vào cách chia nhóm và phụ thuộc vào các ký hiệu gán cho mỗi nhóm
- Nếu tồn tại cách chia nhóm ở tất cả các mức (Fano) hoặc biểu diễn nhị phân chính xác tuyệt đối, khi đó chúng ta sẽ có mã thống kê tối ưu, $\overline{R} = H(X)$
- Nếu $H(X) < 1$, các phép mã hóa sẽ không tối ưu. Giải pháp: gộp các ký hiệu nguồn.

Nguyên tắc mã hóa Huffman

- Nguyên tắc:
  - Từ mã có xác suất xuất hiện nhỏ có chiều dài lớn
  - Hai từ mã có xác suất gần giống nhau mã hóa bằng hai từ mã gần giống nhau (trọng số gần nhau)
  - Hai nhóm từ mã có chung một phần prefix có xác suất gần nhau
- Giải thuật
  1. Liệt kê các ký hiệu theo thứ tự xác suất giảm dần
  2. Chọn hai ký hiệu có xác suất nhỏ nhất, thay bằng một tin mới. Mỗi ký hiệu được gán cho một nhãn 0 hoặc 1
  3. Các tin còn lại và tin mới lại được ghi vào cột thứ 2 theo thứ tự giảm dần
  4. Bắt đầu từ bước 1 cho đến khi chỉ còn 2 ký hiệu
  5. Các từ mã thu được bằng cách khai triển các nhãn tương ứng với ký hiệu và các ký hiệu mới tạo thành từ ký hiệu đó

2.3.Mã hóa với từ mã có độ dài thay đổi

Tìm code Huffman cho nguồn tin có 8 ký hiệu, cấu trúc thống kê cho trong cột thứ 2

1	2	3	4	5	6	7
A0.25 B0.25 C0.14 D0.14 E0.055 F0.055 G0.055 (1) H0.055 (0)	A0.25 B0.25 C0.14 D0.14 GH0.11 E0.55 (1) F0.55 (0)	A0.25 B0.25 C0.14 D0.14 GH0.11(1) EF0.11 (0)	A0.25 B0.25 EFGH0.22 C0.14 (1) D0.14 (0)	CD0.28 A0.25 B0.25(1) EFGH0.22 (0)	EFGH-B0.47 CD0.28 (1) A0.25 (0)	CD-A 0.53 (1) EFGH-B 0.47 (0)

Vậy các từ mã sẽ là

A	B	C	D	E	F	G	H
10	01	111	110	0001	0000	0011	1111

- Entropy của nguồn: 2.715
- Số lượng ký hiệu trung bình: 2.72
- Hiệu quả mã hóa: 0.98

Cách thiết lập cây mã: gốc ở bên phải, mỗi lần gộp là một mức

Mã hóa nguồn có cấu trúc thống kê thay đổi

- Trong tất cả các quá trình nói trên, mã hiệu phụ thuộc vào cấu trúc thống kê của nguồn
- Có thể tăng hiệu quả mã hóa bằng cách mã hóa từng khối ký hiệu. Khi đó độ dài từ trung bình bị giới hạn bởi

$$JH(X) \leq \overline{R} < JH(X) + 1$$

  Hiệu quả của phép mã hóa sẽ gần 1 hơn.
- Khi cấu trúc thống kê của nguồn thay đổi, cần thay đổi mã hiệu theo. Bộ giải mã và bộ mã hóa cần thống nhất với nhau mã hiệu sử dụng
- Giải pháp
  - Mã hóa động: mỗi khi truyền và nhận một ký hiệu, bộ giải mã và bộ mã hóa cập nhật lại thông tin về các ký hiệu, cấu trúc lại cây mã, lập mã hiệu mới. Ví dụ: Mã Huffman động
  - Mã hóa không phụ thuộc cấu trúc thống kê. Ví dụ: mã hóa Lempel-Ziv

3. Mã hóa cho nguồn dừng rời rạc

1. Mã hóa nguồn rời rạc không nhớ
2. Mã hóa cho nguồn dừng rời rạc
   - Entropy của nguồn dừng rời rạc
   - Mã hóa Huffman cho nguồn rời rạc
   - Mã hóa độc lập thống kê nguồn Lempel-Ziv
3. Cơ sở lý thuyết mã hóa nguồn liên tục
4. Các kỹ thuật mã hóa nguồn liên tục

- Xét nguồn có nhớ (các biến ngẫu nhiên tại các thời điểm phụ thuộc thống kê)
- Entropy của một khối các biến ngẫu nhiên liên tiếp được tính theo công thức

$$H(X_1X_2\ldots X_k) = \sum_{i=1}^{k} H(X_i|X_1X_2\ldots X_{i-1})$$

- Có thể tính Entropy trung bình cho từng ký hiệu

$$H_k(X) = \frac{1}{k}H(X_1X_2\ldots X_k)$$

- Cho k tiến tới vô cùng

$$\exists? \lim_{k\to\infty} \frac{1}{k}H(X_1X_2\ldots X_k)$$

3.1.Entropy của nguồn dừng rời rạc *(Tiếp)*

- Mặt khác entropy của từng ký hiệu cũng có thể được định nghĩa theo

$$\exists? \lim_{k\to\infty} H(X_k|X_1X_2\ldots X_{k-1})$$

- Có thể chứng minh hai giới hạn này tồn tại và bằng nhau với nguồn dừng (Xem [Proakis])

- Mã hóa từng khối J ký hiệu của nguồn dừng với mã Huffman
- Số lượng ký hiệu nhị phân tối thiểu phải sử dụng thỏa mãn

$$H(X_1X_2\ldots X_J) \leq \overrightarrow{R} < H(X_1X_2\ldots X_J)$$

$$H_J(X) \leq \overline{R} > H_J(X) + \frac{1}{J}$$

- Cho J tiến tới vô cùng

$$H(X) \leq \overline{R} < H(X) + \epsilon$$

ϵ bé tùy ý

- Vậy hiệu quả của mã hóa Huffman với nguồn dừng có nhớ có thể tiệm cận 1

Tuy nhiên .....

- Cần biết hàm mật độ phân bố xác suất đồng thời của J ký hiệu nguồn liên tiếp
- Cần đánh giá các xác suất
- Cần tính lại mã hiệu
- Cần đồng bộ mã hiệu mã hóa và giải mã
- Cần ....?
- Độ phức tạp thuật toán lớn

3.5. Mã hóa bằng từ điển nguồn Lempel-Ziv

- Xét nguồn nhị phân
- Chia đầu ra nguồn nhị phân này thành các câu có tối đa n ký hiệu. Nguyên tắc chia dùng một từ điển như sau
  - Lập một bảng từ điển gồm 3 cột: vị trí, nội dung, từ mã
  - Xuất phát, từ điển rỗng, vị trí trong từ điển là 0000, cột nội dung có giá trị rỗng
  - Nhận ký hiệu đầu tiên 1, coi đó là một câu, ghi vào cột nội dung. Cột vị trí ghi giá trị 00001
  - Nhận ký hiệu 0, coi đó là một câu, ghi vào cột nội dung. Cột vị trí ghi giá trị 00000
  - Nhận các bộ 2 ký hiệu tiếp theo. Cột vị trí tăng dần các giá trị
    - Nếu là 00, từ mã bằng vị trí của 0 thêm 0 ở cuối
    - Nếu là 01, từ mã bằng vị trí của 0 thêm 1 ở cuối
    - Nếu là 10, từ mã bằng vị trí của 1 thêm 0 ở cuối
    - Nếu là 11, từ mã bằng vị trí của 1 thêm 1 ở cuối
  - Tiếp tục như vậy với các bộ 3,4 ... ký hiệu cho đến khi tràn từ điển

Ví dụ

Mã hóa dãy ký hiệu
10101101001001110101000011001110101100011011
Chia thành các câu
1,0,10,11,01,00,100,111,010, 1000, 011, 001, 110,101, 10001, 1011

Xây dựng từ điển

1,0,10,11,01,00,100,111,010, 1000, 011, 001, 110,101, 10001, 1011

Vị trí trong từ điển	Nội dung	Từ mã
0001	1	00001
0010	0	00000
0011	10	00010
0100	11	00011
0101	01	00101
0110	00	00100
0111	100	00110
1000	111	01001
1001	010	01010
1010	1000	01110
1011	011	01011
1100	001	01101
1101	110	01000
1110	101	00111
1111	10001	10101
	1011	11101

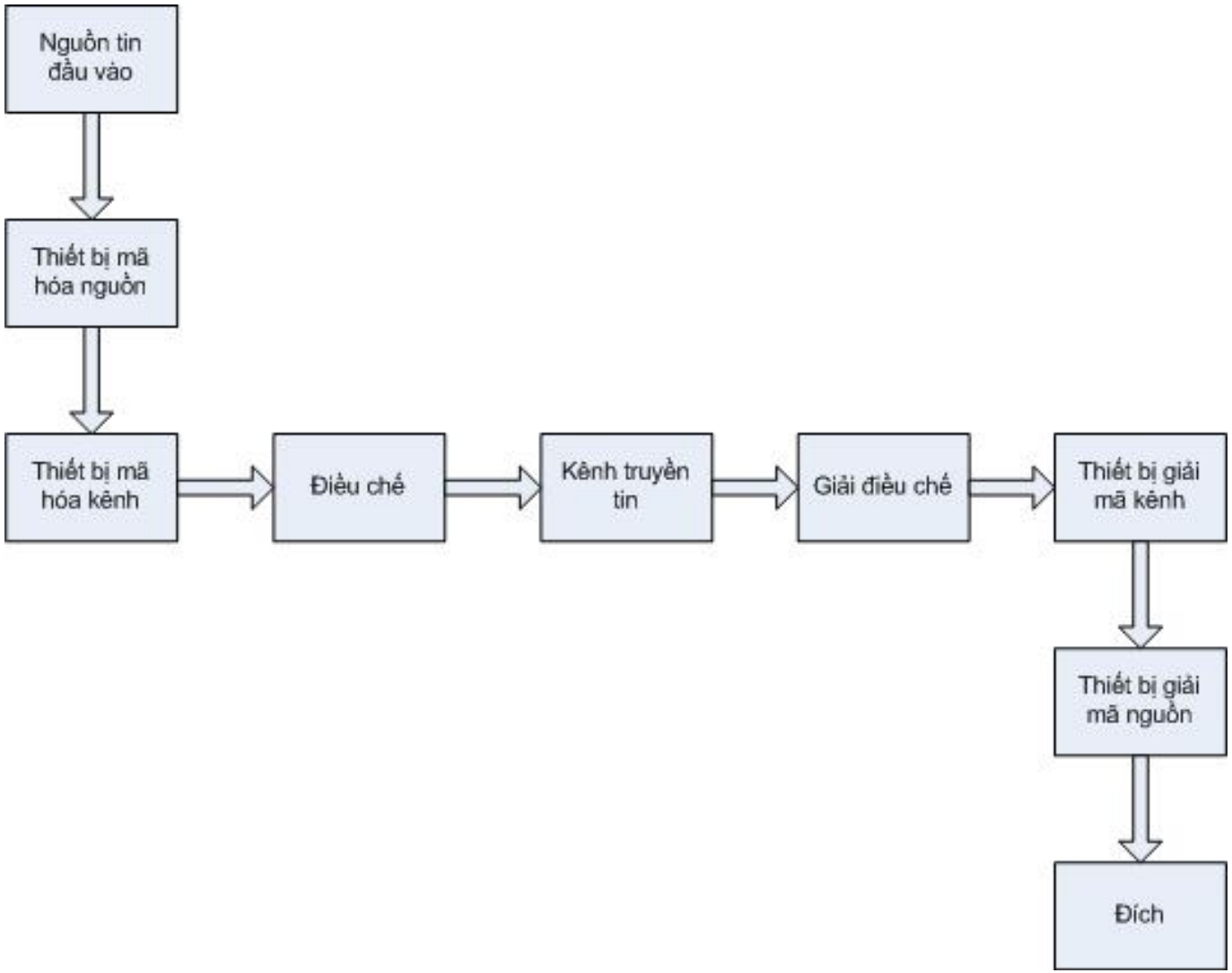
Đánh giá

- Quá trình giải mã: nhận được một từ mã, giải mã từ trái qua phải để thu được câu cần tìm. Thông thường, từ điển được xây dựng ở cả hai phía mã hóa và giải mã để làm tăng tốc độ giải mã
- Giới hạn về kích thước của từ điển
- Trong ví dụ trên, mã hóa 44 bít dùng 16 từ mã 5 bít: không hiệu quả
- Nếu có 2^n từ mã, mã hóa được 2^{n-1} câu, vậy chiều dài tối đa của câu trong trường hợp xấu nhất là n-1 bít.
- Khi nào có trường hợp xấu nhất?
- Thông thường, khi độ dài các câu đủ lớn, các chuỗi ký hiệu lặp lại nhiều, khi đó hiệu quả mã hóa sẽ lớn

4. Cơ sở lý thuyết mã hóa nguồn liên tục

1. Mã hóa nguồn rời rạc không nhớ
2. Mã hóa cho nguồn dừng rời rạc
3. Cơ sở lý thuyết mã hóa nguồn liên tục
   - Khái niệm cơ bản
   - Hàm tốc độ tạo tin sai lệch
   - Lượng tử hóa vô hướng
   - Lượng tử hóa vector
4. Các kỹ thuật mã hóa nguồn liên tục

4.1.Khái niệm cơ bản



4.1.Khái niệm cơ bản

- Nguồn tương tự: quá trình ngẫu nhiên liên tục
- Trong các hệ thống truyền thông: nguồn tương tự được biến thành nguồn tin rời rạc, xử lí rồi lại được biến đổi thành nguồn liên tục
- Rời rạc hóa nguồn liên tục
  - Lấy mẫu nguồn tương tự: biến đổi nguồn tương tự thành một chuỗi các giá trị ngẫu nhiên liên tục tại các thời điểm thời gian rời rạc
  - Lượng tử hóa nguồn tương tự: mã hóa các giá trị liên tục bằng nguồn rời rạc
- Tại đích, nguồn rời rạc được tổng hợp thành nguồn tương tự
  - Tái tạo lại giá trị liên tục của chuỗi giá trị ban đầu từ các ký hiệu của nguồn rời rạc
  - Kết nối các giá trị liên tục thành một tín hiệu ngẫu nhiên đầu ra
- Do quá trình lượng tử, đầu ra sai khác với đầu vào: Sai số lượng tử

Định luật lấy mẫu

$$X(t) = \sum_{n=-\infty}^{\infty} X\left(\frac{n}{2W}\right) \frac{\sin\left[2\pi W\left(t-\frac{n}{2W}\right)\right]}{2\pi W\left(t-\frac{n}{2W}\right)}$$

$$\phi(\tau) = \sum_{n=-\infty}^{\infty} \phi\left(\frac{n}{2W}\right) \frac{\sin\left[2\pi W\left(\tau-\frac{n}{2W}\right)\right]}{2\pi W\left(\tau-\frac{n}{2W}\right)}$$

Quá trình lượng tử hóa

- Ví dụ : Biểu diễn một biến ngẫu nhiên liên tục theo phân bố chuẩn Gaussian
  - Lượng tử hóa dùng 1 bít
  - Lượng tử hóa dùng 2 bít: cần tìm vị trí thích hợp cho bít thứ hai để có sai số nhỏ nhất
- Mục đích của một thiết bị lượng tử hóa là giảm tối thiểu sai số này với một số bít/biến ngẫu nhiên nhỏ nhất (hoặc ngược lại)

4.2.Hàm tốc độ tạo tin sai lệch

- Là tốc độ bít nhỏ nhất đảm bảo một sai lệch xác định
- Cho một nguồn tin với phân bố xác suất nguồn cho trước, các mẫu tín hiệu được lượng tử hóa với sai số $d(x, \overline{x})$.
- Sai số nhỏ đòi hỏi tốc độ truyền tin lớn và ngược lại
- Hàm tốc độ tạo tin-sai lệch biểu diễn liên hệ giữa sai số và tốc độ truyền tin

Định nghĩa

- Xác định sai số
  - Nguồn sau khi lấy mẫu gồm nhiều mẫu
  - Với mỗi mẫu, ký hiệu sai lệch là $d(x_k, \overline{x_k})$
  - Sai lệch có thể được định nghĩa theo nhiều cách: phương sai $E[(X - \overline{X})^2]$, sai lệch lớn nhất $E(max(|X - \overline{X}|))$
  - Sai số trên tập các biến ngẫu nhiên là kỳ vọng toán học cua d

$$D = E[d(X_k, \overline{X_k})] = \frac{1}{n}\sum_{k=1}^{n} E\left[d(x_k, \overline{x}_k)\right] = E\left[d(x_k, \overline{x_k})\right]$$

- Hàm tốc độ tạo tin-sai lệch

$$R_I(D) = \min_{p(\overline{x}/x):E\left[d\left(X,\overline{X}\right)\right]\leq D} I(X, \overline{X})$$

- Biểu diễn tốc độ lập tin lý thuyết nhỏ nhất để có sai số nhỏ hơn D, lượng tin tối thiểu để biểu diễn nguồn với sai số D

Định lý mã hóa nguồn với sai số cho trước

Theorem

Tồn tại một phương pháp mã hóa nguồn, mã hóa các mẫu, với tốc độ tạo tin tối thiểu là $R(D)$ bit/ký hiệu, với sai số sát tùy ý với D, với mọi D.

- Khẳng định ý nghĩa thực tiễn của khái niệm hàm tốc độ tạo tin-sai lệch
- Giới hạn lý thuyết/thực tế của quá trình lượng tử hóa
- Rất khó tính toán hàm tốc độ lập tin-sai số với các nguồn có nhớ hoặc không phải Gaussian

Ví dụ về nguồn chuẩn gaussian, không nhớ, rời rạc theo thời gian

- Tốc độ lập tin tối thiểu là

$$R_g(D) = \begin{cases} \frac{1}{2}\log_2(\sigma_x^2/D) & (0 \le D \le \sigma_x^2) \\ 0 & (D \ge \sigma_x^2) \end{cases}$$

- Như vậy nếu sai số cần thiết lớn hơn sai lệch của nguồn đã cho, không cần truyền tin nữa

Hàm sai số-tốc độ lập tin

- Biểu diễn sai số nhỏ nhất có thể có khi mã hóa một nguồn tin tương tự

$$D(R) = \min_{p(\overline{x}/x):\overline{R}\leq R} d(X, \overline{X})$$

- Có thể sử dụng một trong hai hàm để biểu diễn liên hệ giữa sai số và tốc độ lập tin
- Với nguồn Gaussian

$$D_g = 2^{-2R}\sigma_x^2$$

4.3.Lượng tử hóa vô hướng

- Xét bài toán lượng tử hóa một biến ngẫu nhiên liên tục (mẫu của một nguồn liên tục dừng không nhớ), biết hàm mật độ phân bố xác suất của biến ngẫu nhiên
- Chia miền giá trị của X thành L khoảng
  $x_0 = -\infty < x_1 < x_2 < x_3 < \ldots < x_k < \ldots < x_L = \infty$
- Mỗi một khoảng $x_{k-1} < x < x_k$ tương ứng với một mức tín hiệu $\overline{x_k}$
- Sai số tổng cộng sẽ là

$$D = \sum_{k=1}^{L} \int_{x_{k-1}}^{x_k} f\left(\overline{x}_k - x\right) p(x).dx$$

- Cần tối thiểu hóa sai số. Lấy đạo hàm theo $\overline{x_k}, x_k$

$$f\left(\overline{x}_k - x_k\right) = f\left(\overline{x}_{k+1} - x_k\right)$$

Và

$$\int_{x_{k-1}}^{x_k} f'\left(\overline{x}_k - x\right) p(x).dx = 0$$

- Để biểu diễn các mức tín hiệu, cần $\log_2 L$ bít. xác suất của mỗi mức tín hiệu sẽ là $p_k = \int_{x_{k-1}}^{x_k} p(x)dx$
- Entropy của nguồn

$$H(\overline{X}) = -\sum_{k=1}^{L} p_k \log_2 p_k$$

4.3.Lượng tử hóa vô hướng *(Tiếp)*

- Để tối ưu hóa, sau đó nguồn cần được mã hóa bằng mã hóa thống kê (Fano-Shannon-Huffman)
- Có thể chọn các mức sao cho các ký hiệu đầu ra đẳng xác suất: phân các miền giá trị đầu vào đẳng xác suất.

Ví dụ: nguồn có phân bố đều

- Biên độ đầu vào dao động trong khoảng $-A, A$, sai số $f = |x - \overline{x}|$
- Cần giải hệ

$$f\left(\overline{x}_k - x_k\right) = f\left(\overline{x}_{k+1} - x_k\right)$$

và

$$\int_{x_{k-1}}^{x_k} f'\left(\overline{x}_k - x\right) p(x).dx = 0$$

- Vậy cần chia đầu vào thành L khoảng đều nhau, trong mỗi khoảng đó lấy giá trị điểm giữa làm mức tín hiệu

Ví dụ: nguồn có phân bố đều *(Tiếp)*

- Sai số tối ưu là

$$D = \sum_{k=1}^{L} \int_{x_{k-1}}^{x_k} f\left(\overline{x}_k - x\right) p(x).dx = \frac{A}{L}$$

- Để có thể mã hóa tối ưu cần chọn L là lũy thừa của 2
- Nếu cho trước D tốc độ mã hóa tối thiểu là $log_2 \frac{A}{D}$ nếu $\frac{A}{D}$ là lũy thừa của 2 hoặc$1 + \lfloor \log_2 \frac{A}{D} \rfloor$ nếu không

4.4.Lượng tử hóa vector

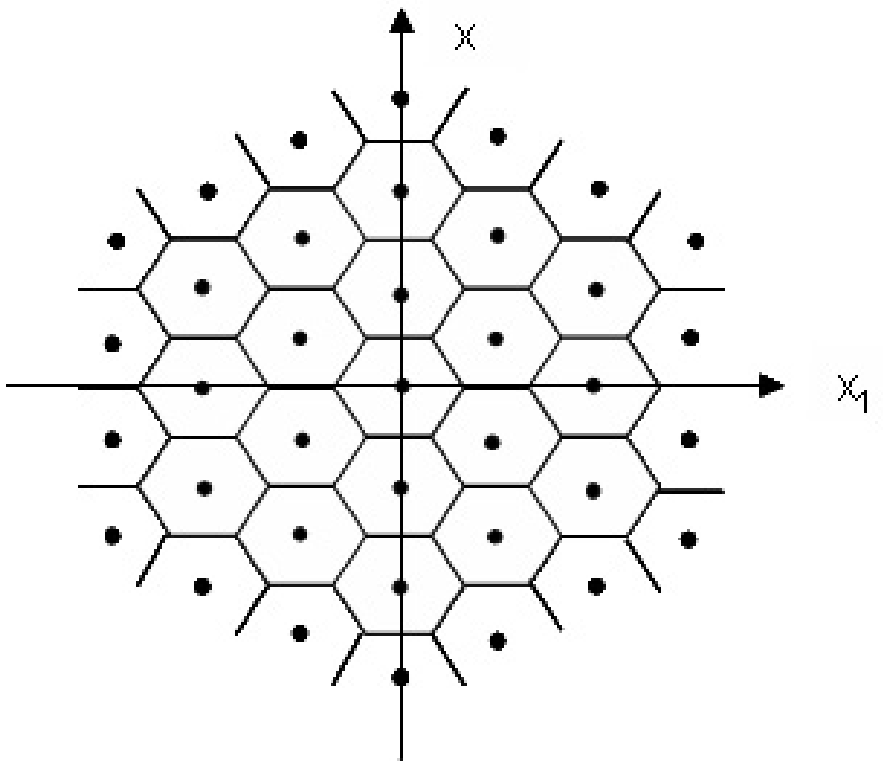
- Trong lượng tử hóa vô hướng
  - miền giá trị của biến ngẫu nhiên đầu vào được chia thành nhiều miền con
  - Tập giá trị trong miền con tương ứng với một mức tín hiệu đầu ra, đảm bảo khoảng cách ngắn nhất tới biên (trung tâm„ trọng tâm)
  - Chỉ dùng cho một biến ngẫu nhiên liên tục-> nguồn dừng, không nhớ
- Có thể tổng quát hóa khái niệm miền giá trị cho không gian n chiều
- Xét cùng lúc nhiều biến ngẫu nhiên, mỗi biến ngẫu nhiên tương ứng với một chiều
- Miền con trở thành một ô trong không gian n-chiều
- Mức tín hiệu đầu ra là một tín hiệu rời rạc ngẫu nhiên nhiều chiều, biểu diễn bằng trung tâm của ô.

4.4.Lượng tử hóa vector

- Xét n biến ngẫu nhiên nhiều chiều đặc trưng cho các mẫu của một nguồn liên tục
- Biểu diễn các biến ngẫu nhiên này trong không gian n chiều
- Chia không gian n chiều thành L ô C_k
- Các tín hiệu đầu vào được lượng tử hóa theo phép mã hóa

$$\overline{X} = Q(X)$$

- X_k là giá trị đầu ra tương ứng với tín hiệu đầu vào trong C_k



- Ví dụ: Lượn tử hóa vector 2 chiều
- Không gian hai chiều chia thành các ô có dạng hình lục giác

Sai số của phép lượng tử hóa vector

$$D = \sum_{k=1}^{L} P(X \in C_k) E\left[d(X, \overline{X}_k) \,|X \in C_k\right] =$$

$$\sum_{k=1}^{L} P(X \in C_k) \int_{X \in C_k} d(X, \overline{X}_k) p(X) dX$$

- Để tối thiểu D
  - Dạng của các ô phụ thuộc vào hàm phân bố xác suất đồng thời
  - Dạng của các ô cũng phụ thuộc vào hàm khoảng cách

$$Q(X) = \overline{X}_k \Leftrightarrow D(X, \overline{X}_k) \leq D(X, \overline{X}_j), k \neq j, 1 \leq j \leq n$$

  - Các mức tín hiệu đầu ra tương ứng là trung tâm của các ô

$$D_k = E\left[d(X, \overline{X}_k) \,|X \in C_k\right] = \int_{X \in C_k} d(X, \overline{X}_k) p(X) dX$$

Sai số-tốc độ lập tin

- Hàm sai số

$$d(X, \overline{X}) = \frac{1}{n} \sum_{k=1}^{n} (x_k - \overline{x}_k)^2$$

- Tốc độ lập tin

$$R = \frac{H(\overline{X})}{n}$$

trong đó

$$H(\overline{X}) = -\sum_{i=1}^{L} p(\overline{X}_i) \log_2 p(\overline{X}_i)$$

- Sai số-tốc độ lập tin

$$D_n(R) = \min_{Q(X)} E\left[d(X, \overline{X})\right]$$

- Hàm sai số-tốc độ lập tin

$$D(R) = \lim_{n \to \infty} D_n(R)$$

5. Các kỹ thuật mã hóa nguồn liên tục

1. Mã hóa nguồn rời rạc không nhớ
2. Mã hóa cho nguồn dừng rời rạc
3. Cơ sở lý thuyết mã hóa nguồn liên tục
4. Các kỹ thuật mã hóa nguồn liên tục
   - Mã hóa tín hiệu miền thời gian
   - Mã hóa tín hiệu miền tần số
   - Mã hóa mô hình nguồn

5.1.Mã hóa tín hiệu miền thời gian

- Biểu diễn tín hiệu theo miền thời gian
- Lấy mẫu tín hiệu theo tốc độ Nyquist, tần số f_s
- Các mẫu được lượng tử hóa

Điều chế mã xung (Pulse Code Modulation)

- Mỗi một mẫu tín hiệu được mã hóa bằng 2^R mức tín hiệu. Tốc độ thông tin của nguồn sau mã hóa là Rf_s bps.
- Giá trị tín hiệu đầu ra

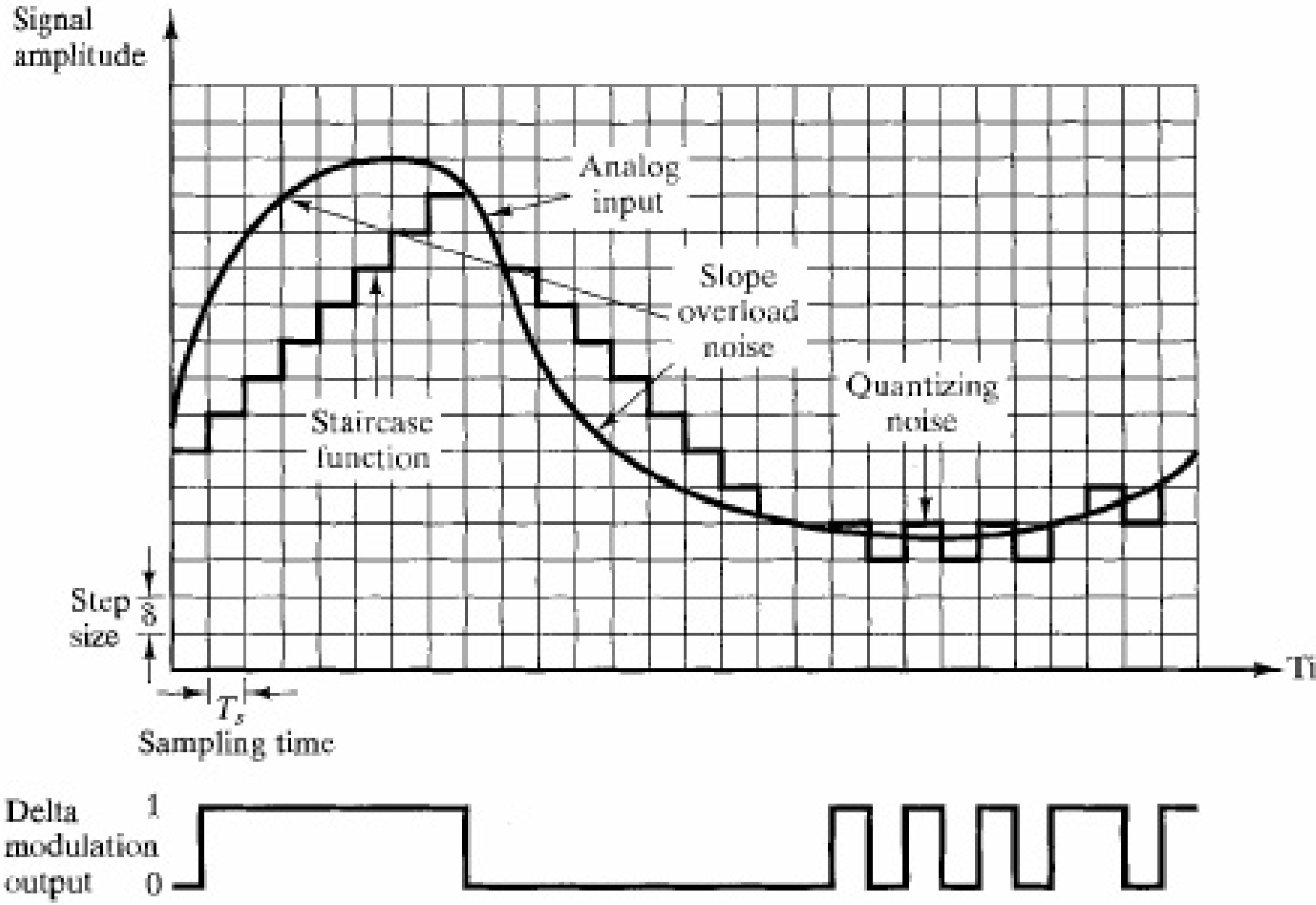
$$\overline{x_n} = x_n + q_n$$

  q_n là nhiễu lượng tử (nhiễu cộng)
- Trong trường hợp các mức đều, mật độ phân bố xác suất đều, sai số lượng tử tối thiểu (xem ví dụ trên) : Bộ lượng tử hóa đồng đều
- Trong trường hợp mật độ phân bố xác suất không đều, cần chọn các mức tương ứng : Bộ lượng tử hóa không đồng đều
- Trong thực tế, với các tín hiệu tiếng nói, thường sử dụng các mức lượng tử theo loga

Mã hóa mã xung vi sai

- Nếu tốc độ lấy mẫu cao, các mẫu có liên hệ với nhau
- Dự đoán giá trị các mẫu?
- Liên hệ thông thường: hàm số liên tục, đạo hàm hữu hạn. Giá trị mẫu sau sai khác với giá trị mẫu trước một khoảng xác định
- Không mã hóa giá trị tín hiệu, chỉ mã hóa sự sai khác so với giá trị của mẫu trước đó
- Xa hơn nữa, có thể mã hóa mẫu hiện tại dựa vào p mẫu trước đó

Ví dụ về DPCM


Signal amplitude
Analog input
Slope overload noise
Quantizing noise
Staircase function
Step size
δ
T_s
Sampling time
Delta modulation output
1
0

PCM và DPCM thích nghi

- PCM và DPCM thích hợp với các nguồn dừng (phân bố thống kê của các biến ngẫu nhiên không thay đổi theo thời gian)
- Trong thực tế các nguồn tin ít khi dừng tuyệt đối
- Phân bố thống kê của các nguồn tin thực tế thay đổi chậm (nguồn gần dừng)
- Có thể cái thiện PCM và DPCM cho phù hợp với các nguồn tin đó: thay đổi các thông số của PCM hoặc DPCM
  - PCM: thay đổi biên độ (thay đổi khoảng cách giữa các mức)
  - DPCM: Thay đổi các thông số của bộ dự đoán

5.2.Mã hóa tín hiệu miền tần số

- Mã hóa băng con
  - Mã hóa hình ảnh và tiếng nói
  - Phân tích thông tin đầu vào theo tần số
  - Chia thành nhiều dải con
  - Mã hóa độc lập từng dải
  - Ví dụ: mã hóa tiếng nói
    - Phần tần số thấp chiếm nhiều năng lượng hơn phần tần số cao
    - Phần tần số thấp được mã hóa bằng số bit ít hơn
- Mã hóa biến đổi thích nghi
  - Các mẫu được chia thành nhiều khung
  - Các khung này được biến đổi sang miền tần số và truyền đi (giống phương pháp trên)
  - Khi nhận được các khung này, biến đổi ngược lại
  - Tùy theo thông số của phổ, các phổ quan trọng được mã hóa nhiều bít hơn
  - Phép biến đổi thường là phép biến đổi Fourier.

5.3.Mã hóa mô hình nguồn

- Mô hình hóa nguồn tin: sử dụng một số tham số (là các phản ứng của nguồn tin với các tín hiệu đầu vào nhất định)
- Mã hóa các tham số
- Giải mã để thu được các tham số
- Phục hồi tín hiệu ban đầu
- Mô hình hay dùng là mô hình tuyến tính