Đại Học Khoa Học Tự Nhiên-Đại Học Quốc Gia Hồ Chí Minh

Khoa Điện Tử Viễn Thông

Lớp Cao Học Điện Tử - Viễn Thông - Máy Tính Khóa 21

….ooOoo….

Môn học : Phương pháp phân tích thống kê

Đề tài : Thuật toán mã hóa Viterbi

GVHD : PGS.TS Nguyễn Hữu Phương

SV : Bùi Thanh Sơn

Nội Dung

1. Giới thiệu lý thuyết mã hóa

1.1 Lý thuyết mã hóa (tiếng Anh: *coding theory*) là một ngành của toán học (*mathematics*) và khoa học điện toán (*computer science*)) nhằm giải quyết tình trạng lỗi dễ xảy ra trong quá trình truyền thông số liệu trên các kênh truyền có độ nhiễu cao (*noisy channels*)), dùng những phương pháp tinh xảo khiến phần lớn các lỗi xảy ra có thể được chỉnh sửa. Nó còn xử lý những đặc tính của mã (*codes*)), và do vậy giúp phù hợp với những ứng dụng cụ thể.

Có hai loại mã hiệu:

1. Mã hóa dùng nguồn (Mã hóa entrôpi (*Entropy encoding*))
2. Mã hóa trên kênh truyền (Sửa lỗi ở phía trước (*Forward error correction*))

Cái đầu tiên chúng ta nói đến là mã hóa dùng nguồn. Ý định của phương pháp này là nén dữ liệu từ chính nguồn của nó, trước khi truyền đi, giúp cho việc truyền thông có hiệu quả hơn. Chúng ta chứng kiến thói quen này hằng ngày trên Internet, nhất là trong cách dùng "zip" nén dữ liệu để giảm lượng dữ liệu phải truyền, giảm nhẹ gánh nặng cho mạng lưới truyền thông, đồng thời thu nhỏ cỡ tập tin. Cái thứ hai là mã hóa trên kênh truyền. Bằng việc cộng thêm những bit mới vào trong dữ liệu được truyền, còn gọi là bit chẵn lẻ (*parity bits*), kỹ thuật này giúp cho việc truyền thông tín hiệu chính xác hơn trong môi trường nhiễu loạn của kênh truyền thông. Có nhiều chương trình ứng dụng, mà người dùng trung bình không để ý đến, sử dụng mã hóa trên kênh truyền. Kỹ thuật Reed-Solomon được dùng để nhằm sửa lỗi do những vết xước và bụi trên bề mặt đĩa âm nhạc CD thông thường. Trong ứng dụng này, kênh truyền thông lại chính là bản thân cái đĩa CD. Điện thoại di động "Cell phones" cũng dùng kỹ thuật mã hóa có hiệu ứng cao (*powerful coding technique*) để sửa các lỗi trong việc truyền sóng rađiô ở tần số cao bị yếu mờ và bị nhiễu. Modem xử lý số liệu, việc truyền thông qua đường điện thoại, và đương nhiên ngay cả chính NASA, tất cả đều sử dụng kỹ thuật mã hóa trên kênh truyền hiệu ứng để truyền những bit số liệu qua đường dây.

1.2. Mã hóa kênh truyền

Mục đích của lý thuyết *Mã hóa trên kênh truyền* (*channel encoding theory*) là tìm những mã có thể truyền thông nhanh chóng, chứa đựng nhiều mã ký (*code word*) hợp lệ và có thể sửa lỗi (*error correction*) hoặc ít nhất phát hiện các lỗi xảy ra (*error detection*). Các mục đích trên không phụ thuộc vào nhau, và mỗi loại mã có công dụng tối ưu cho một ứng dụng riêng biệt. Những đặc tính mà mỗi loại mã này cần còn tuỳ thuộc nhiều vào xác suất lỗi xảy ra trong quá trình truyền thông. Đối với một đĩa CD thông thường, lỗi trong âm thanh xảy ra chủ yếu là do bụi và những vết xước trên mặt đĩa. Vì thế, các mã được lồng vào với nhau. Dữ liệu được phân bổ trên toàn bộ mặt đĩa. Tuy không được tốt cho lắm, song một mã tái diễn đơn giản có thể được dùng làm một ví dụ dễ hiểu. Chẳng hạn, chúng ta lấy một khối số liệu bit (đại diện cho âm thanh) và truyền gửi chúng ba lần liền. Bên máy thu, chúng ta kiểm tra cả ba phần lặp lại ở trên, từng bit từng bit một, rồi lấy cái nào có số bầu cao nhất. Điểm trái khoáy ở đây là, chúng ta không chỉ truyền gửi các bit theo thứ tự. Chúng ta lồng nó vào với nhau. Khối dữ liệu này, trước tiên, được chia ra làm 4 khối nhỏ. Sau đó chúng ta gửi một bit ở khối đầu tiên, tiếp theo một bit ở khối thứ hai v.v tuần tự qua các khối. Việc này được lặp đi lặp lại ba lần để phân bổ số liệu ra trên bề mặt đĩa. Trong ngữ cảnh của mã tái diễn đơn

giản ở trên, việc làm này hình như không được hiệu quả cho lắm. Song hiện nay có những mã có hiệu ứng cao, rất phù hợp với việc sửa lỗi xảy ra đột ngột do một vết xước hay một vết bụi, khi dùng kỹ thuật lồng số liệu nói trên.

Mỗi mã thường chỉ thích hợp cho một ứng dụng nhất định. Viễn thông trong vũ trụ (*deep space*) bị giới hạn bởi nhiễu nhiệt (*thermal noise*) trong thiết bị thu. Hiện trạng này không xảy ra một cách đột phát bất thường, song xảy ra theo một chu trình tiếp diễn. Tương tự như vậy, modem với dải tần hẹp bị hạn chế vì nhiễu âm tồn tại trong mạng lưới điện thoại. Những nhiễu âm này có thể được biểu hiện rõ hơn bằng một mô hình âm tạp tiếp diễn. Điện thoại di động "Cell phones" hay có vấn đề do sự suy sóng nhanh chóng xảy ra. Tần số cao được dùng có thể gây ra sự suy sóng tín hiệu một cách nhanh chóng (*rapid fading*), ngay cả khi máy nhận chỉ dời chỗ vài phân Anh (*inches*) [1]. Một lần nữa, người ta hiện đã có một loại thuộc hạng Mã hóa trên kênh truyền được thiết kế để đối đầu với tình trạng suy sóng.

Từ "Lý thuyết mã hóa đại số" ám chỉ để một chi nhánh của lý thuyết mã hóa trên kênh truyền, trong đó đặc tính của mã được biểu hiện bằng các đại số và dựa vào đó mà nghiên cứu sâu hơn.

Lý thuyết mã hóa đại số được chia ra làm 2 loại mã chính

1. Mã khối tuyến tính (*Linear block codes*)
2. Mã kết hợp (*Convolutional codes*)

Chúng phân tích ba đặc tính sau của mã -- nói chung là:

- Chiều dài của mã (*code word length*)
- Tổng số các mã ký hợp lệ (*total number of valid code words*)
- Khoảng cách Hamming tối thiểu giữa hai mã ký hợp lệ (*the minimum Hamming distance between two valid code words*)

1.3. Mã khối tuyến tính

Mã khối tuyến tính mang tính năng tuyến tính (*linearity*), chẳng hạn tổng của hai mã ký nào đấy lại chính là một mã ký; và chúng được ứng dụng vào các bit của nguồn trên từng khối một; cái tên mã khối tuyến tính là vì vậy (*linear block codes*). Có những khối mã bất tuyến tính, song khó mà chứng minh được rằng một mã nào đó là một mã tốt nếu mã ấy không có đặc tính này.

Bất cứ mã khối tuyến tính nào cũng được đại diện là (n,m,d_{min}), trong đó

1. n, là chiều dài của mã ký, trong ký hiệu (*symbols*),
2. m, là số ký hiệu nguồn (*source symbols*) được dùng để mã hóa tức thời,
3. d_{min}, là khoảng cách hamming tối thiểu của mã (*the minimum hamming distance for the code*)

Có nhiều loại mã khối tuyến tính, như

1. Mã tuần hoàn (*Cyclic codes*) (Mã Hamming là một bộ phận nhỏ (*subset*) của mã tuần hoàn)
2. Mã tái diễn (*Repetition codes*)

3. Mã chẵn lẻ (*Parity codes*)
4. Mã Reed-Solomon (*Reed Solomon codes*)
5. Mã BCH (*BCH code*)
6. Mã Reed-Muller
7. Mã hoàn hảo (*Perfect codes*)

Mã khối được gắn liền với bài toán "đóng gói đồng xu" là bài toán gây một số chú ý trong nhiều năm qua. Trên bề diện hai chiều, chúng ta có thể hình dung được vấn đề một cách dễ dàng. Lấy một nắm đồng xu, để nằm trên mặt bàn, rồi dồn chúng lại gần với nhau. Kết quả cho chúng ta một mẫu hình lục giác tương tự như hình tổ ong. Các mã khối còn dựa vào nhiều chiều khác nữa, không dễ gì mà hình dung được. Mã Golay [2] có hiệu ứng cao, dùng trong truyền thông qua khoảng không vũ trụ, sử dụng những 24 chiều. Nếu được dùng là mã nhị phân (thường thấy), các chiều ám chỉ đến chiều dài của mã ký như đã định nghĩa ở trên.

Lý thuyết về mã sử dụng mô hình hình cầu với số chiều "N". Lấy ví dụ, bao nhiêu đồng xu phải cần để phủ kín một mặt bàn, hay trong khoảng không 3 chiều, bao nhiêu hòn bi phải cần để nhồi kín một hình cầu. Những cân nhắc khác bao gồm việc chọn lựa mã. Lấy ví dụ, do nhồi nhữmg hình lục lăng vào trong một cái hộp hình chữ nhật, chúng ta để lại những khoảng trống ở các góc. Khi các chiều của hộp được tăng lên, tỷ lệ phần trăm so sánh của các khoảng trống nhỏ đi, cho đến một cỡ nào đấy, những phần nhồi chiếm hết các khoảng không và mã này được gọi mã hoàn hảo. Số mã kiểu này tương đối hiếm (Hamming [*n*,*k*,3], Golay [24,12,8],[23,12,7],[12,6,6])

Một điều thường bị bỏ qua là số lượng những hàng xóm kế cận (*neighbors*) mà một mã ký có thể có. Chúng ta có thể dùng lại ví dụ các đồng xu ở đây. Đầu tiên, chúng ta gộp các đồng xu lại theo các hàng hình chữ nhật. Mỗi một đồng xu có 4 đồng kế cận (và 4 cái ở bốn góc xa hơn). Trong bố cục của hình lục giác, mỗi đồng xu có 6 đồng kế cận. Khi chúng ta tăng số chiều lên, số lượng các đồng kế cận tăng lên một cách nhanh tróng.

Kết quả là số lượng các âm tạp, bên cạch các âm chính, mà máy thu có thể chọn, cũng tăng lên, và do đó mà gây ra lỗi. Đây chính là khuyết điểm căn bản của mã khối, và cũng là khuyết điểm của tất cả các loại mã. Có thể việc gây lỗi trở nên khó khăn hơn, nếu chỉ có một hàng xóm kế cận mà thôi, song con số các hàng xóm kế cận có thể lớn đến độ làm cho chính tổng xác suất lỗi bị ảnh hưởng (*total error probability actually suffers*).

1.4. Mã kết hợp

Mã kết hợp (*Convolutional codes*) được sử dụng trong các modem dải tần âm (*voiceband modems*) (V.32, V.17, V.34) và trong các điện thoại di động GSM, cũng như trong các thiết bị truyền thông của quân đội vũ trang và trong các thiết bị truyền thông với vệ tinh.

Ý định ở đây là làm cho tất cả các ký hiệu mã ký (*codeword symbol*) trở thành tổng trọng số (*weighted sum*) của nhiều loại ký hiệu thông điệp trong nhập liệu (*various input message symbols*). Nó tương tự như toán kết hợp được dùng trong các hệ tuyến tính bất biến (*linear time invariant systems*) để dùng tìm xuất liệu (*output*) của một hệ thống, khi chúng ta biết nhập liệu (*input*) và các đáp ứng xung (*impulse response*).

Nói chung chúng ta tìm xuất liệu của bộ mã hóa kết hợp hệ (*system convolutional encoder*), tức sự kết hợp của nhập liệu bit, đối chiếu với trạng thái của bộ mã hóa kết hợp (*convolution encoder*), hoặc trạng thái của các thanh biến (*registers*).

Về cơ bản mà nói, mã kết hợp không giúp thêm gì trong việc chống nhiễu hơn một mã khối tương ứng. Trong nhiều trường hợp, chúng nói chung cho chúng ta một phương pháp thực thi đơn giản hơn, hơn hẳn một mã khối có hiệu quả tương ứng (*a block code of equal power*). Bộ mã hóa thường là một mạch điện đơn giản, có một bộ nhớ (*state memory*), một vài biện pháp truyền thông tin phản hồi báo tình hình (*some feedback logic*), thường là các cổng loại trừ XOR (*XOR gates*). Bộ mã hóa có thể được thực thi trong phần mềm hay phần sụn (*firmware*).

Thuật toán Viterbi (*Viterbi algorithm*) là một thuật toán ngắn gọn nhất (*optimum algorithm*) được dùng để giải mã các mã kết hợp. Hiện có những phương pháp giảm ước giúp vào việc giảm khối lượng tính toán phải làm. Những phương pháp này phần lớn dựa vào việc tìm tuyến đường có khả năng xảy ra cao nhất (*most likely paths*). Tuy không ngắn gọn, song trong môi trường nhiễu thấp hơn, người ta thường thấy chúng cho những kết quả khả quan. Các bộ điều hành vi xử lý hiện đại (*Modern microprocessors*) có khả năng thực hiện những thuật toán tìm giảm ước nói trên với tỷ lệ trên 4000 mã ký trong một giây.

Trong bất kỳ một hệ thống truyền dẫn vô tuyến nào từ tương tự đến hệ thống số, các tín hiệu đều phải được điều chế lên một tần số sóng mang nhất định để truyền dẫn qua không gian. Do đó, dù là các tàu vũ trụ, các tên lửa hay hệ thống thông tin tế bào thì vấn đề tách tín hiệu ra khỏi nhiễu đóng vai trò cốt lõi thể hiện tính ưu việt của mỗi hệ thống thông tin.

Trong vũ trụ, tín hiệu luôn bị chìm sâu vào trong nhiễu do nhiễu vô tuyến từ các thiên hà hoặc quá trình phát xạ từ mặt trời. Hay như trên mặt đất, tín hiệu của một máy vô tuyến tế bào bị gây nhiễu bởi rất nhiều máy phát vô tuyến tế bào khác xung quanh nó. Vậy làm thế nào để có thể tách tín hiệu cần thiết ra khỏi nhiễu? Để thực hiện điều đó, người ta đã dùng giải pháp mã hóa kênh các tín hiệu để thêm vào thông tin gốc các bit dư để bảo vệ trước khi chúng được phát qua kênh truyền có nhiễu. Bằng cách mã hóa kênh như vậy, các bit thông tin (bit 0 hay 1) có thể được biểu diễn không chỉ biểu diễn bằng một góc dịch pha (Điều chế khóa pha *M*-PSK) mà còn có thể được biểu diễn bằng 4, 8 hoặc nhiều hơn nữa các ký hiệu. Với hệ thống dùng mã hóa kênh, ở phía thu chúng ta không tách các bit thông tin thực tế mà thay vào đó là các ký hiệu mã hóa để từ các ký hiệu đó chúng ta có thể khôi phục lại các bit thông tin gốc. Ý tưởng cơ bản của giải pháp lựa chọn mã hóa kênh là nếu như một trong số các ký hiệu phát qua kênh bị méo do nhiễu thì những ký hiệu bị ảnh hưởng bởi nhiễu ít hơn sẽ giúp giải mã các bit thông tin thực tế với độ tin cậy cao hơn.

Trong môi trường nhiễu, có một thực tế là nếu ngay khi nhận được tín hiệu thu chúng ta thực hiện giải mã các bit 0 hoặc 1 thì hiệu quả sẽ không cao. Thay vào đó, chúng ta chỉ đánh giá độ tin cậy, (rất tin cậy, tin cậy, tin cậy thấp , tin cậy rất thấp), khi giải mã bit 1 hoặc bit 0 dựa trên tín hiệu thu được. Với cách giải mã như vậy người ta gọi là quyết định mềm (soft decision). Những thông tin thể hiện độ tin cậy của quyết định mềm đó được đưa đến bộ giải mã để tìm ra bit thông tin gốc là 0 hay 1 thông qua việc so sánh kết quả với những bit thông tin lân cận. Giải thuật mềm như đã nêu ở trên sẽ rất phức tạp, độ phức tạp sẽ tăng theo hàm mũ khi độ dài các thanh ghi dịch trong bộ tạo mã tăng lên. Công trình nghiên cứu trong vòng ba tháng của Viterbi đã làm giảm độ phức tạp thông qua việc kết hợp các quyết định mềm để quyết định chuỗi bit thông tin nào là giống với chuỗi bit được phát đi nhất. Giải pháp này của Viterbi đã được chứng minh là cho kết quả giống với kết quả của phương pháp giải mã tối ưu với độ phức tạp cao ở trên. Với ưu điểm đó, JPL ngay lập tức sử dụng thuật toán giải mã

Viterbi trong hệ thống thông tin vệ tinh cũng như vũ trụ của mình. Sau này các hệ thống thông tin quân sự cũng như các mạng thông tin số như các tuyến truyền dẫn viba hay hệ thống TTDĐ tế bào GSM, CDMA đều dùng thuật toán giải mã này giải mã mã xoắn.

2. Cơ sở toán học của thuật toán Viterbi

Với bộ mã có đầu vào k bit song song, đầu ra n bit, tốc độ mã r = k/n. Chuỗi tín hiệu vào ký hiệu x, x= (x 0 (1) , x 0 (2) , ..., x 0 (k) , x 1 (1) , ..., x 1 (k) , x L+m-1 (1) , ..., x L+m-1 (k)) Từ mã c, c =(c 0 (1) , c 0 (2) , ..., c 0 (n) , c 1 (1) , ..., c 1 (n) , c L+m-1 (1) , ..., c L+m-1 (n)) Trong đó L là độ dài của chuỗi tín hiệu vào, m là độ dài lớn nhất trong số các thanh ghi.

Từ mã nhận được r r =(r 0 (1) , r 0 (2) , ..., r 0 (n) , r 1 (1) , ..., r 1 (n) , r L+m-1 (1) , ..., r L+m-1 (n)) Giải mã ra y y =(y 0 (1) , y 0 (2) , ..., y 0 (n) , y 1 (1) , ..., y 1 (n) , y L+m-1 (1) , ..., y L+m-1 (n)) y là một từ mã trong bộ mã. Thuật toán sẽ tìm y sao cho xác suất p(r|y) là lớn nhất.

Thuật toán giả sử kênh truyền là không có nhớ, tức là lỗi trên 1 bit bất kỳ không phụ thuộc vào các bit trước nó. Từ lý thuyết xác suất, ta có xác suất chung của các sự kiện độc lập bằng tích xác suất của các sự kiện riêng biệt.

$$p(\mathbf{r}|\mathbf{y}) = \prod_{i=0}^{L+m-1} [p(r_i^{(1)}|y_i^{(1)})p(r_i^{(2)}|y_i^{(2)})\cdots p(r_i^{(n)}|y_i^{(n)})]$$
$$= \prod_{i=0}^{L+m-1} \left(\prod_{j=1}^{n} p(r_i^{(j)}|y_i^{(j)}) \right)$$

Ước lượng max(p(r|y)) tương đương max(log p(r|y)) do đó có:

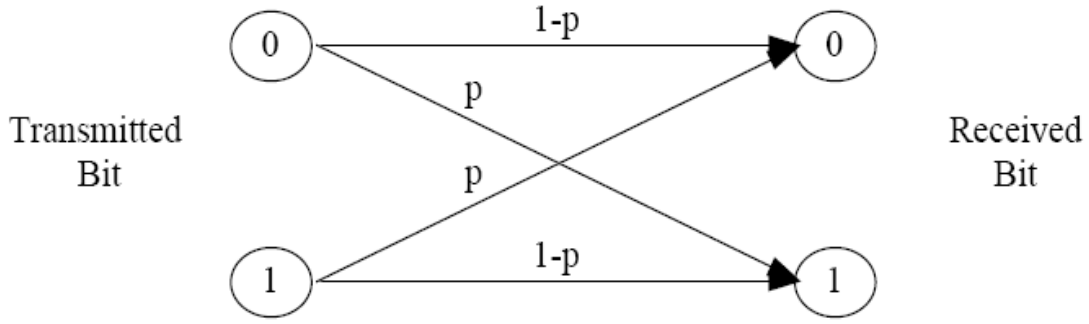
$$\log p(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=1}^{n} \log p(r_i^{(j)}|y_i^{(j)}) \right)$$

Để tính tổng này đơn giản ta định nghĩa thang đo bit như sau:

$$M(r_i^{(j)}|y_i^{(j)}) = a[\log p(r_i^{(j)}|y_i^{(j)}) + b]$$

Với a, b được xác định sao cho thang đo bit là 1 số nguyên dương nhỏ

Giá trị a và b được định nghĩa là kênh đối xứng nhị phân (Binary Symmetric Channel - BSC) hoặc giải mã cứng (hard-decision decoding) Ví dụ một kênh đối xứng nhị phân:



$$a = \frac{1}{\log p - \log(1-p)}$$

$$b = -\log(1-p)$$

$$M(r_i^{(j)} | y_i^{(j)}) = \frac{1}{[\log p - \log(1-p)]}[\log p(r_i^{(j)} | y_i^{(j)}) - \log(1-p)]$$

$M(r_i^{(j)} \mid y_i^{(j)})$	**Received Bit** $r_i^{(j)} = 0$	**Received Bit** $r_i^{(j)} = 1$
Decoded Bit $y_i^{(j)} = 0$	0	1
Decoded Bit $y_i^{(j)} = 1$	1	0

Điều này giống với khái niệm khoảng cách Hamming mà chúng ta đã học. Giải thuật Viterbi chọn từ mã y thông qua cây trellis sao cho từ mã này có khoảng cách Hamming nhỏ nhất với từ mã nhận được r.

Khoảng cách Hamming lớn nhất Kênh bất kỳ: a và b có thể tính dựa trên các lỗi và giá trị kiểm thử để đạt được bit metric chấp nhận được

$$M(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1} \left(\sum_{j=1}^{n} M(r_i^{(j)} | y_i^{(j)}) \right)$$

Biểu thức này chỉ ra tổng chi phí cho ước lượng từ chuỗi nhận được r với từ mã sau giải mã y trên cây trellis. Giá trị nhánh thứ k được định nghĩa:

$$M(\mathbf{r}_k | \mathbf{y}_k) = \sum_{j=1}^{n} M(r_k^{(j)} | y_k^{(j)})$$

Và một phần nhánh thứ k là:

$$M^k(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{k} M(\mathbf{r}_i|\mathbf{y}_i)$$
$$= \sum_{i=0}^{k}\left(\sum_{j=1}^{n} M(r_i^{(j)}|y_i^{(j)})\right)$$

Giá trị nhánh thứ k đưa ra chi phí cho việc chọn một nhánh từ cây trellis. Giá trị một phần nhánh thứ k đưa ra chi phí cho chọn một từ mã y với chỉ số thời gian là k.

Giải thuật Viterbi sử dụng cây trellis để tính toán giá trị đường dẫn - Mỗi trạng thái (node) trong cây trellis được gán một giá trị, một phần giá trị đường dẫn. - Một phần giá trị đường dẫn được xác định từ trạng thái 0 tại thời điểm 0 đến trạng tiếp k tại thời điểm t>=0. - Tại mỗi trạng thái, một phần giá trị đường dẫn tốt nhất hoặc là giá trị lớn hơn hoặc nhỏ hơn phụ thuộc vào a và b được chọn theo cách 2 hay cách 1. - Giá trị được lựa chọn biểu diễn đường dẫn còn lại và đường dẫn này được lưu trữ còn các đường khác bị loại bỏ khỏi cây trellis. Giải thuật Viterbi chọn đường dẫn cuối cùng còn lại như là đường dẫn Maximum Likelihood . Lần trở lại theo đường dẫn này trên cây trellis sẽ thu được từ mã cần tìm.

t = t+1 Tính một phần giá trị đường dẫn trong tất cả đường dẫn bắt đầu ở trạng thái Sk tại thời điểm t. Đầu tiên, tìm giá trị nhánh thứ t

$$M(\mathbf{r}_t|\mathbf{y}_t) = \sum_{j=1}^{n} M(r_t^{(j)}|y_t^{(j)})$$

Giá trị này được tính từ khoảng cách Hamming

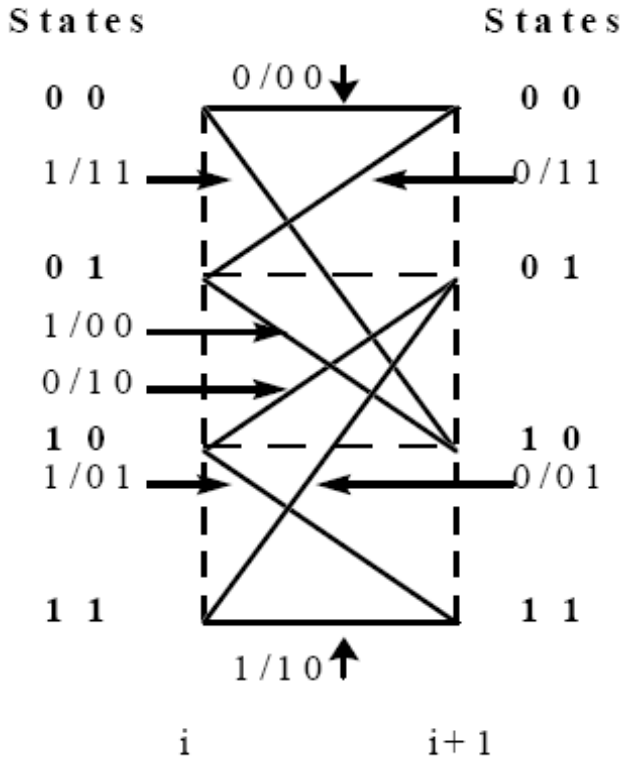
$$\sum_{j=1}^{n}\left|r_t^{(j)} - y_t^{(j)}\right|$$

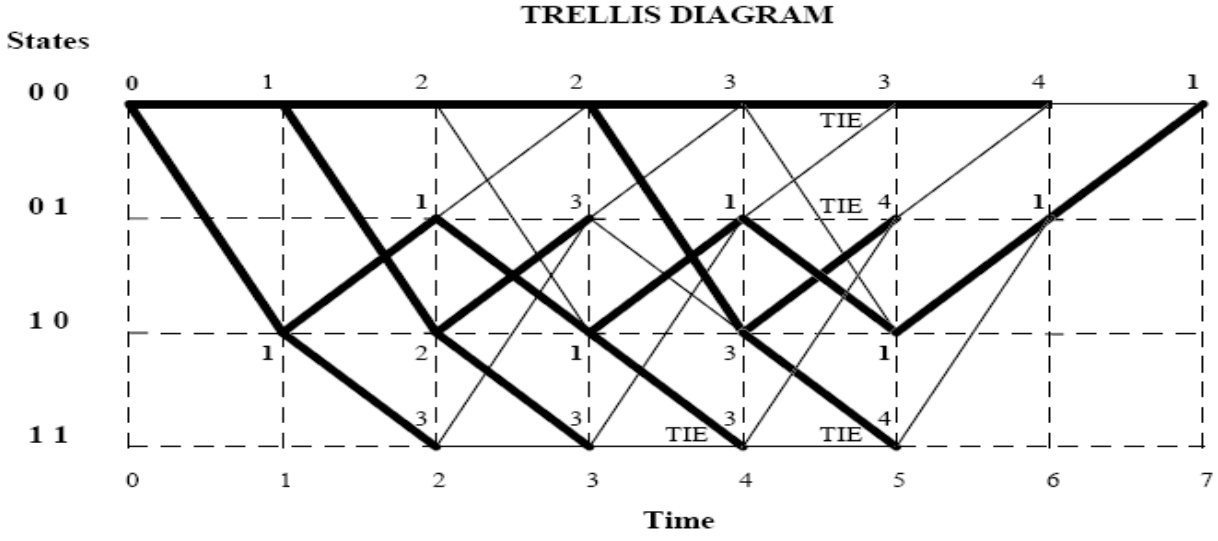
Tiếp theo, tính một phần giá trị đường dẫn thứ t

$$M^t(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{t} M(\mathbf{r}_i|\mathbf{y}_i)$$

Giá trị này được tính từ V(S k,t-1) + M(r t |y t) 3. Thiết lập V(S k,t) là giá trị đường dẫn tốt nhất từ trạng thái S k tại thời điểm t. Một phần giá trị đường dẫn tốt nhất là đường dẫn có giá trị nhỏ nhất. Nếu có nhiều đường dẫn có giá trị nhỏ nhất như nhau thì có thể chọn bất kì đường nào 4. Lưu đường này và bit còn lại và trạng thái của nó 5. Nếu t∞11. 1. t=0 V(S 0,0) = 0; V(S k,t) = + < L+m-1 thì quay lại bước 2 Kết quả của giải thuật Viterbi là đường dẫn trên cây trellis duy nhất tương ứng với từ mã ML

Ví dụ : Chuỗi input x = {1010100} với 2 bit cuối 00 là bit xóa thanh ghi Sau khi qua bộ mã hóa được c = { 11 , 10, 00, 10, 00, 10, 11}, sau khi truyền qua kênh thì chuỗi nhận được r = { 10 , 10, 00, 10, 00, 10, 11} có lỗi (gạch chân). Hình sau thể hiện sơ đồ dịch chuyển trạng thái trên cây trellis của bộ mã này





Từ cây trellis trong hình, chuỗi mã ước lượng y = {11, 10, 00, 10, 00, 10, 11}. Tận dụng đồ thị trạng thái dịch chuyển trong hình trước ta có chuỗi thông tin ước lượng x’ = {1010100}

Phương pháp 1: sử dụng khoảng cách Euclidean thay vì khoảng cách Hamming. Những bit nhận được sử dụng trong trường hợp khoảng cách Euclidean được xử lý bằng lượng tử hóa nhiều bit.

Phương pháp 2: sử dụng giá trị tương quan với bit nhận được sử dụng trong trường hợp giá trị này cũng được xử lý bởi lượng tử hóa nhiều bit.

Trong giải mã soft-decision, phía thu không gán giá trị 0 hoặc 1 cho mỗi bit nhận được mà sử dụng lượng tử hóa số bit không xác định. Chuỗi nhận được r được sử dụng trực tiếp trong giải mã Viterbi soft-decision. Giải thuật này tương đương HDVA, chỉ khác là sử dụng khoảng cách Euclidean.

Giải thuật SDVA1 như sau:

1. t=0

V(S0,0) = 0; V(Sk,t) = +∞

2. t = t+1

Tính một phần giá trị đường dẫn trong tất cả đường dẫn bắt đầu ở trạng thái Sk tại thời điểm t.

Đầu tiên , tìm giá trị nhánh thứ t

$$M(\mathbf{r}_t|\mathbf{y}_t) = \sum_{j=1}^{n} M(r_t^{(j)}|y_t^{(j)})$$

Giá trị này được tính từ khoảng cách Euclidean

$$\sum_{j=1}^{n} (r_t^{(j)} - y_t^{(j)})^2$$

Tiếp theo , tính một phần giá trị đường dẫn thứ t Giá trị này được tính từ

$$M^t(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{t} M(\mathbf{r}_i|\mathbf{y}_i)$$

V(S k,t-1) + M(r t |y t)

Thiết lập V(S k,t) là giá trị đường dẫn tốt nhất từ trạng thái S k tại thời điểm t Một phần giá trị đường dẫn tốt nhất là đường dẫn có giá trị nhỏ nhất. Nếu có nhiều đường dẫn có giá trị nhỏ nhất như nhau thì có thể chọn bất kì đường nào.

Lưu đường này và bit còn lại và trạng thái của nó .Nếu t< L+m-1 thì quay lại bước 2.

Đối với phương pháp 2, giải thuật SDVA2 được phát triển như sau. Hàm (likelihood function) được biểu diễn bởi hàm phân bố xác suất chuẩn:

$$p(r_i^{(j)}|y_i^{(j)}) = \frac{1}{\sqrt{\pi N_o}} e^{-(r_i^{(j)} - y_i^{(j)}\sqrt{E_b})^2/N_o}$$

Với E b là năng lượng nhận được trên mỗi bit của từ mã. N 0 mật độ nhiễu phổ trên một mặt. Bit nhận được là biến ngẫu nhiên chuẩn với giá trị trung bình $y_i^{(j)}\sqrt{E_b}$ và sự khác biệt N 0 /2. log 2 vế của biểu thức trên ta có:

$$\log p(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{L+m-1}\left(\sum_{j=1}^{n} \log p(r_i^{(j)}|y_i^{(j)})\right)$$

$$= \sum_{i=0}^{L+m-1}\left\{\sum_{j=1}^{n}\left[-\frac{(r_i^{(j)} - y_i^{(j)}\sqrt{E_b})^2}{N_o} - \log\sqrt{\pi N_o}\right]\right\}$$

$$= \frac{-1}{N_o}\sum_{i=0}^{L+m-1}\left[\sum_{j=1}^{n}(r_i^{(j)} - y_i^{(j)}\sqrt{E_b})^2\right] - \frac{(L+m)n}{2}\log \pi N_o$$

$$= \frac{-1}{N_o}\sum_{i=0}^{L+m-1}\left\{\sum_{j=1}^{n}[r_i^{(j)2} - 2r_i^{(j)}y_i^{(j)}\sqrt{E_b} + y_i^{(j)2}E_b]\right\} - \frac{(L+m)n}{2}\log \pi N_o$$

where $y_i^{(j)2} = 1$

$$= C_1\sum_{i=0}^{L+m-1}\left[\sum_{j=1}^{n} r_i^{(j)}y_i^{(j)}\right] + C_2$$

$= C_1 (r \bullet y) + C_2$

Với C_1 và C_2 không là hàm của y

Từ đây, bit được định nghĩa như sau

$$M(r_i^{(j)}|y_i^{(j)}) = r_i^{(j)}y_i^{(j)}$$

SDVA2 thực hiện như sau: S k,t là trạng thái trên cây Trellis tương ứng trạng thái S k tại thời điểm t. Mọi trạng thái trên cây Trellis được gán một giá trị V(S k,t)

t=0 => $V(S_{0,0}) = 0$; $V(S_{k,t}) = -\infty$

(a) t = t+1

(b) Tính một phần giá trị đường dẫn trong tất cả đường dẫn bắt đầu ở trạng thái S k tại thời điểm t.∞22. 1. t=0 V(S 0,0) = 0; V(S k,t) = -

Đầu tiên, tìm giá trị nhánh thứ t \

$$M(\mathbf{r}_t|\mathbf{y}_t) = \sum_{j=1}^{n} M(r_t^{(j)}|y_t^{(j)})$$

Giá trị này được tính từ sự tương quan giữa $r_i^{(j)}$ và $y_i^{(j)}$, $\sum_{j=1}^{n} r_i^{(j)} y_i^{(j)}$

Tiếp theo, tính một phần giá trị đường dẫn thứ t $M^t(\mathbf{r}|\mathbf{y}) = \sum_{i=0}^{t} M(\mathbf{r}_i|\mathbf{y}_i)$. Giá trị này được tính từ V(S k,t-1) + M(r t |y t)

Thiết lập V(S k,t) là giá trị đường dẫn tốt nhất từ trạng thái S k tại thời điểm t Một phần giá trị đường dẫn tốt nhất là đường dẫn có giá trị lớn nhất. Nếu có nhiều đường dẫn có giá trị lớn nhất như nhau thì có thể chọn bất kì đường nào Lưu đường này và bit còn lại và trạng thái của nó Nếu t< L+m-1 thì quay lại bước 2

3. Tài liệu tham khảo

1. http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/viterbi_algorithm/s1_pg1.html
2. http://en.wikipedia.org/wiki/Viterbi_algorithm
3. http://pw1.netcom.com/~chip.f/viterbi/tutorial.html
4. William Cary Huffman, Vera Pless. **Fundamentals of error-correcting codes**. Cambridge University Press, 2003. 646
5. Todd K. Moon. **Error correction coding: mathematical methods and algorithms**. John Wiley and Sons, 2005. 756
6. George Cyril Clark, J. Bibb Cain. **Error-correction coding for digital communications**. Springer, 1981. 422