|  |
| --- |
| **ĐẠI HỌC QUỐC GIA HÀ NỘI**  **TRƢỜNG ĐẠI HỌC CÔNG NGHỆ**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Hoàng Xuân Huấn**  ***Giáo trình***  **HỌC MÁY**      **2015** |

# MỤC LỤC

**LỜI NÓI ĐẦU ......................................................................................................................................... 7** **CHƢƠNG 1 GIỚI THIỆU ............................................................................................................. 9**

**1.1.** **HỌC MÁY LÀ GÌ? ..................................................................................................... 9**

1.1.1. Khái niệm học máy ........................................................................................................ 9 1.1.2. Tại sao cần nghiên cứu học máy? ................................................................................ 10

1.1.3. Một số lĩnh vực liên quan............................................................................................. 11

1.1.4. Các bài toán học thiết lập đúng đắn ........................................................................... 11

**1.2. MỘT SỐ LỚP BÀI TOÁN ỨNG DỤNG ĐIỂN HÌNH ............................................................ 12**

**1.3.** **KIẾN TRÚC VÀ THIẾT KẾ MỘT HỆ HỌC ..................................................................... 14**

1.3.1. Đối sánh vân tay.......................................................................................................... 14

1.3.1.1 Lược đồ đối sánh dựa trên đặc trưng chi tiết ........................................................ 14

1.3.1.2. Các thành phần của hệ nhận dạng mẫu ............................................................... 16

1.3.1.3. Thiết kế hệ nhận dạng ......................................................................................... 17

1.3.2. Tìm đường đi tối ưu cho robot ..................................................................................... 19

1.3.2.1. Thuật toán học tăng cường để tìm gần đúng đường đi tối ưu .............................. 19

1.3.2.2. Thiết kế hệ học cho bài toán tổng quát ................................................................ 20 **KẾT LUẬN ......................................................................................................................... 22**

**BÀI TẬP ............................................................................................................................ 22** **CHƢƠNG 2 HỌC CÓ GIÁM SÁT .......................................................................................... 23**

**2.1 CÁC BÀI TOÁN VÀ VÍ DỤ ĐIỂN HÌNH ............................................................................. 23**

2.1.1. Bài toán học khái niệm ................................................................................................ 23

2.1.2. Bài toán học nhiều lớp ................................................................................................ 26

2.1.3. Bài toán nội suy và hồi quy .......................................................................................... 26

**2.2. HỌC QUY NẠP ............................................................................................................. 27**

2.2.1. Phát biểu bài toán ....................................................................................................... 27

2.2.2. Một số khái niệm cơ bản ............................................................................................. 27

**2.3. HỌC KHÁI NIỆM VÀ BÀI TOÁN TÌM KIẾM ..................................................................... 28**

2.3.1. Thuật toán tìm kiếm giả thuyết chi tiết nhất .............................................................. 29

2.3.2. Các thuật toán loại trừ ứng cử ..................................................................................... 32

2.3.2.1**.** Thuật toán liệt kê loại trừ ứng cử (List-then-eliminate algorithm) ...................... 33

2.3.2.2. Một cách biểu diễn compact đối với không gian tường thuật ........................... 33

2.3.2.3. Thuật toán loại trừ ứng cử ................................................................................. 34

**2.4. HỌC HÀM NỘI SUY VÀ HỒI QUY ................................................................................... 37**

2.4.1. Phương pháp trực tiếp tìm hàm nội suy ...................................................................... 37

2.4.2. Tìm hàm hồi quy .......................................................................................................... 38

**2.5. MỘT SỐ VẤN ĐỀ LIÊN QUAN ........................................................................................ 39**

2.5.1. Khuynh hướng quy nạp ................................................................................................ 39

2.5.2. Học gần đúng theo xác suất ........................................................................................ 40

2.5.3. Chiều Vapnik-Chervonenkis ......................................................................................... 42 **KẾT LUẬN ......................................................................................................................... 43**

**BÀI TẬP ............................................................................................................................ 44**

**CHƢƠNG 3 HỌC BẰNG CÂY QUYẾT ĐỊNH ................................................................... 46**

**3.1. BIỂU DIỄN GIẢ THUYẾT BẰNG CÂY QUYÊT ĐỊ NH ......................................................... 46**

**3.2. CÁC THUẬT TOÁN HỌC ............................................................................................... 49**

3.2.1. Thuật toán học ID3 ..................................................................................................... 49

3.2.1.1. Các khái niệm ........................................................................................................ 49

3.2.1.2. Mô tả thuật toán .................................................................................................. 51

3.2.1.3. Nhận xét về ID3 ................................................................................................... 53

3.2.2. Thuật toán C4.5 ........................................................................................................... 55

**3.3. MỘT SỐ VẤN ĐỀ KHÁC TRONG HỌC BẰNG CÂY QUYẾT ĐỊ NH........................................ 57**

3.3.1. Phù hợp trội và cách khắc phục ................................................................................... 57

3.3.2. Tiêu chuẩn chọn thuộc tính ......................................................................................... 59

3.3.3. Xử lý giá trị thuộc tính bị thiếu của mẫu ................................................................... 60

3.3.4. Cây phân lớp và hồi quy .............................................................................................. 61 **KẾT LUẬN ......................................................................................................................... 62**

**BÀI TẬP ............................................................................................................................ 63**

**CHƢƠNG 4 PHÂN LỚP MẪU NHỜ HÀM PHÂN BIỆT ............................................... 65**

**4.1. HÀM PHÂN BIỆT VÀ MIỀN QUYẾT ĐỊ NH .................................................................... 65**

**4.2. CÁC MÔ HÌNH TUYẾN TÍNH ........................................................................................ 68**

4.2.1. Tách được bởi siêu phẳng ............................................................................................ 68

4.2.2. Thuật toán học perceptron ......................................................................................... 70

4.2.3. Thuật toán bình phương tối thiểu ............................................................................... 72

4.2.4. Phân lớp khoảng cách cực tiểu .................................................................................... 75

4.2.4.1. Các mêtric trong không gian đặc trưng. .............................................................. 76

4.2.4.2. Phân biệt tuyến tính Euclide ................................................................................ 77

4.2.4.3. Phân biệt tuyến tính Mahalanobis ....................................................................... 78

4.2.5. Máy véctơ tựa .............................................................................................................. 79

4.2.5.1. Các lớp tách được tuyến tính ................................................................................ 79

4.2.5.2. Các lớp không tách được tuyến tính .................................................................... 81

4.2.5.3. Hàm nhân (Kernel function) ................................................................................. 82 **4.3. BÀI TOÁN TỶ LỆ CHIỀU ............................................................................................... 83**

**KẾT LUẬN ......................................................................................................................... 85**

**BÀI TẬP ............................................................................................................................ 85**

**CHƢƠNG 5 HỌC THỐNG KÊ ................................................................................................. 87**

**5.1. LÝ THUYẾT QUYẾT ĐỊ NH BAYES ................................................................................. 87**

5.1.1. Bài toán và các quy tắc quyết đị nh ............................................................................ 87

5.1.2. Quyết đị nh Bayes trong học khái niệm ...................................................................... 89

5.1.3. Giả thuyết có khả năng nhất và sai số bình phương tối thiểu ..................................... 91

**5.2. PHÂN LỚP BAYES ........................................................................................................ 92**

5.2.1. Các quy tắc phân lớp MAP và ML ................................................................................ 93

5.2.2. Phân lớp cực tiểu rủi ro ................................................................................................ 95

5.2.3. Bộ phân lớp tối ưu bayes ............................................................................................. 98

5.2.4. Phân lớp Bayes ngây thơ (Naïve Bayes) ....................................................................... 99

5.2.5. Phân lớp Bayes khi mỗi lớp có phân bố chuẩn .......................................................... 101

**5.3. QUY TẮC QUYẾT ĐỊ NH K LÁNG GIỀNG GẦN NHẤT (K-NN) .......................................... 103**

**5.4. LỰA CHỌN ĐẶC TRƯNG VÀ PHÂN TÍCH THÀNH PHẦN CHÍNH ....................................... 104**

5.4.1. Lựa chọn đặc trưng .................................................................................................... 104

5.4.2. Phân tích thành phần chính (PCA) ............................................................................. 105

**5.5. ĐÁNH GIÁ CÁC BỘ PHÂN LỚP .................................................................................... 106**

5.5.1. Ước lượng lỗi của bộ phân lớp .................................................................................... 106

5.5.2. Phương pháp k-tập (k-folds) đánh giá phương pháp học ......................................... 107

5.5.3. So sánh các bộ phân lớp ............................................................................................ 108

5.5.4. Một số đại lượng và thông tin khác ........................................................................... 109 **KẾT LUẬN ....................................................................................................................... 110**

**BÀI TẬP .......................................................................................................................... 111** **CHƢƠNG 6 HỌC KHÔNG GIÁM SÁT............................................................................. 112**

**6.1. ƯỚC LƯỢNG HÀM MẬT ĐỘ ......................................................................................... 112**

6.1.1. Kỹ thuật có tham số ................................................................................................... 112

6.1.2. Kỹ thuật phi tham số ................................................................................................. 114

**6.2. PHÂN CỤM DỮ LIỆU .................................................................................................. 117**

6.2.1. Bài toán phân cụm dữ liệu ......................................................................................... 117 6.2.2 . Vấn đề chuẩn hóa dữ liệu ......................................................................................... 119 6.3.3. Phương pháp phân cấp .............................................................................................. 120

6.3.4. Phương pháp phân hoạch .......................................................................................... 122

6.3.5. Phân cụm bán giám sát .............................................................................................. 127 **KẾT LUẬN ....................................................................................................................... 128**

**BÀI TẬP .......................................................................................................................... 129**

**CHƢƠNG 7 MẠNG NƠRON NHÂN TẠO ...................................................................... 130**

**7.1. GIỚI THIỆU .............................................................................................................. 130** c ......................................................... 130

7.1.2. Mô hình và kiến trúc mạng nơ ron ........................................................................... 131

7.1.2.1. Cấu tạo của nơron .............................................................................................. 132

7.1.2.2. Các kiểu kiến trúc mạng nơron ........................................................................... 133 ng nơron ......................................................................................... 135

**7.2. PERCEPTRON ........................................................................................................... 137**

7.2.1. Perceptron của Roseblatt .......................................................................................... 137

7.2.2. Mạng ADALINE .......................................................................................................... 138

**7.3. PERCEPTRON NHIỀU TẦNG (MẠNG MLP) .................................................................. 143**

7.3.1. Kiến trúc mạng .......................................................................................................... 143

7.3.2. Thuật toán huấn luyện lan truyền ngược (BP).......................................................... 144

**7.4. MẠNG HÀM CƠ SỞ BÁN KÍNH (MẠNG RBF) ............................................................... 151**

7.4.1. Kiến trúc mạng RBF .................................................................................................. 151

ng RBF ................................................................................................ 152

7.4.2.1. Các thuật toán dựa trên tìm cực tiểu SSE ........................................................... 152

7.4.2.2. Phương pháp huấn luyện lặp mạng nội suy ....................................................... 153 **KẾT LUẬN ....................................................................................................................... 158**

**BÀI TẬP .......................................................................................................................... 158**

**CHƢƠNG 8 CÁC MÔ HÌNH HỌC ĐỊA PHƢƠNG ........................................................ 160**

**8.1. HỌC K-LÁNG GIỀNG GẦN NHẤT (K-NN) ..................................................................... 160**

8.1.1. Lược đồ tổng quát ..................................................................................................... 160

8.1.2. Thuật toán nghị ch đảo khoảng cách ........................................................................ 161

**8.2. HỒI QUY TRỌNG SỐ ĐỊ A PHƯƠNG .............................................................................. 162**

8.2.1. Lược đồ tổng quát ...................................................................................................... 162

8.2.2. Hồi quy tuyến tính đị a phương ................................................................................. 163

**8.3. MẠNG NƠRON RBF HỒI QUY ..................................................................................... 165**

8.3.1. Đặt vấn đề ................................................................................................................. 165

8.3.2. Xây dựng mạng nơron RBF hồi quy ............................................................................ 165

**8.4. MẠNG RBF ĐỊ A PHƯƠNG .......................................................................................... 166**

8.4.1. Kiến trúc và thủ tục xây dựng mạng .......................................................................... 166

8.4.2. Thuật toán phân cụm nhờ cây k-d ............................................................................. 168 **8.5. LẬP LUẬN DỰA TRÊN TÌNH HUỐNG (CBR) ................................................................... 170**

**KẾT LUẬN ....................................................................................................................... 171**

**BÀI TẬP .......................................................................................................................... 173** **CHƢƠNG 9 HỌC TĂNG ......................................................................................................... 174**

**9.1. TÁC TỬ VÀ CÁC BÀI TOÁN HỌC .................................................................................. 174**

9.1.1. Một số ví dụ ............................................................................................................... 174

9.1.2. Tác tử ......................................................................................................................... 176

9.1.3. Các bài toán học ........................................................................................................ 178

**9.2. HỌC Q (Q learning) ................................................................................................... 180**

9.2.1. Học Q trong các bài toán đơn đị nh ........................................................................... 180

9.2.1.1. Hàm Q ............................................................................................................... 181

9.2.1.2. Một thuật toán học Q ........................................................................................ 181

9.2.2. Học Q trong các bài toán ngẫu nhiên ........................................................................ 187

**9.3. PHUƠNG PHÁP TỐI ƯU ĐÀN KIẾN (ACO) ................................................................... 189**

9.3.1. Phát biểu bài toán tối ưu tổ hợp tổng quát ............................................................... 189

9.3.2. Thuật toán tổng quát ................................................................................................. 190 **KẾT LUẬN ....................................................................................................................... 193**

**BÀI TẬP .......................................................................................................................... 194** **CHƢƠNG 10 KẾT HỢP CÁC BỘ HỌC .............................................................................. 195**

**10.1. LÝ DO NÊN HỌC TẬP THỂ ......................................................................................... 195**

**10.2. PHƯƠNG PHÁP BỎ PHIẾU ........................................................................................ 196**

**10.3. KỸ THUẬT TẠO VÀ KẾT HỢP BỘ NHẬN DẠNG CƠ SỞ .................................................. 198**

10.3.1. Nhặt theo gói ........................................................................................................... 198

10.3.2. Nhặt đị nh hướng ..................................................................................................... 199

10.3.3. Rừng ngẫu nhiên ...................................................................................................... 202

**10.4. KIẾN TRÚC BẬC THANG ........................................................................................... 203 KẾT LUẬN ....................................................................................................................... 204**

**BÀI TẬP .......................................................................................................................... 205**

**TÀI LIỆU THAM KHẢO ........................................................................................................... 206 BẢNG CHỮ VIẾT TẮT .............................................................................................................. 207**

# LỜI NÓI ĐẦU

Học m{y l| một phần của lĩnh lực trí tuệ nh}n tạo, mục đích của nó l| ph{t triển c{c phương ph{p x}y dựng c{c chương trình m{y tính tự động cải tiến chất lượng thực hiện nhờ sử dụng dữ liệu hoặc kinh nghiệm đã có. Những nghiên cứu trong lĩnh vực n|y nhằm đ{p ứng c{c b|i to{n đa dạng trong thực tiễn, bắt đầu từ c{c b|i to{n nhận dạng ngôn ngữ, tiếng nói , ph}n tích dữ liệu thống kê đến c{c b|i to{n điều khiển tự động<

Cùng với sự ph{t triển nhanh chóng của công nghệ thông tin, đã có nhiều ứng dụng th|nh công của học m{y trong nhiều lính vực kh{c nhau như: c{c hệ nhận dạng tiếng nói, chữ viết tay; tiếp thị (marketing) nhờ ph}n tích dữ liệu b{n lẻ, c{c hệ thống điều khiển tự động <. Trong đó, ứng dụng trung t}m của học m{y l| kh{m ph{ tự động tri thức từ dữ liệu nhằm trợ giúp quyết định.

Gi{o trình n|y giới thiệu kiến thức nhập môn trong thời lượng ba tín chỉ cho sinh viên của c{c ng|nh trong khoa công nghệ thông tin. Vì học m{y l| lĩnh vực quan trọng, hướng ứng dụng, ph{t triển nhanh v| đa dạng nên với thời lượng như vậy không thể chuyển tải nhiều. Hơn nữa, để hiểu thấu đ{o c{c kỹ thuật học m{y thì sinh viên phải có nền tảng tốt về to{n, đặc biệt l| giải tích ngẫu nhiên nên chúng tôi chú trọng v|o c{c phương ph{p cơ bản v| hướng dẫn sử dụng m| không thể đi vào các kỹ thuật ph{t triển quá sâu.

C{c phương ph{p v| thuật to{n học l| trung t}m của lĩnh vực học m{y, ở đ}y được trình bày theo từng nhóm hướng tiếp cận v| b|i to{n giải quyết thể hiện ở tiêu đề chương để người đọc dễ theo dõi v| thấy được to|n cảnh. Tuy nhiên, việc ph}n loại c{c tiếp cận của phương ph{p học có tính nhập nhằng, chẳng hạn, c{c thuật to{n ở c{c chướng 4 v| chướng 6 có thể đưa ra c{ch nhìn để xếp v|o tiếp cận học thống kê. Vì vậy, chúng tôi chọn c{ch diễn đạt để nội dung phù hợp với tiếp cận của mỗi chương. Ngoại trừ hai chương đầu, nội dung cả c{c chương được viết tương đối độc l}p v| người dùng có thể đọc riêng rẽ. Tuy vậy, nếu đủ thời gian thì nên theo đúng trình tự được trình b|y thì dễ tiếp thu hơn.

Chương đầu của gi{o trình giới thiệu một định nghĩa cho học m{y, ph{c họa bức tranh chung về b|i học v| quy trình thiết kế hệ học. Chương 2 c{c b|i to{n học có gi{m s{t, bắt đầu từ kh{i niệm chung về học quy nạp v| hai thuật to{n đơn giản để tìm tập luật trong học kh{i niệm nhằm cho người đọc thấy được đặc điểm chính của việc tìm kiếm giả thuyết trong không gian giả thuyết, sau đó l| b|i to{n hồi quy v| c{c vấn đề liên quan đến học quy nạp. C{c phương ph{p học c}y quyết định cơ bản để tìm tập luật trong học có gi{m s{t được trình b|y trong chương 3. Chương 4 giới thiệu c{c mô hình tuyến tính để ph}n biệt mẫu. C{c phương ph{p dựa trên mô hình x{c suất như lý thuyết quyết định Bayes v| ứng dụng, phương ph{p ph}n lớp k-l{ng giềng gần nhất, một số vấn đề liên quan tới b|i to{n học có gi{m s{t như chọn đặc trưng v| đ{nh gi{ bộ ph}n lớp được giới thiệu trong chương 5. Chương 6 giới thiệu c{c phương ph{p học không gi{m s{t thường gặp trong ước lượng h|m mật độ v| ph}n cụm dữ liệu. Chương 7 d|nh cho giới thiệu c{c kiến thức chung về mạng nơron v| một số mạng truyền tới thường gặp nhất. Một số phương ph{p học địa phương cho b|i to{n hồi quy v| phương ph{p lập luận dựa trên tình huống được trình b|y trong chương 8. Chương 9 d|nh cho giới thiệu b|i to{n học tăng cường v| ứng dụng. Một số phương ph{p kết hợp c{c bộ học để tăng độ chính x{c v| một số tiếp cận để x}y dựng c{c bộ nhận dạng cơ sở được giới thiệu trong chương cuối.

Trong trình b|y, chúng tôi chú trọng giới thiệu c{c thuật to{n dưới dạng dễ hiểu v| dễ vận dụng. Về hình thức, tùy theo ngữ cảnh m| c{c thuật to{n có thể được đặc tả dưới dạng giả mã hoặc mô tả rõ từng bước thực hiện để người đọc l|m quen với cả hai dạng thông dụng này.

Dựa trên gi{o trình n|y, Các độc giả muốn mở rộng kiến thức chung có thể tham khảo các t|i liệu \*1-3+, khi cần tìm hiểu s}u hơn về học thống kê có thể tham khảo c{c t|i liệu [4-7], để có nội dung đầy đủ hơn về nhận dạng mẫu có thể tham khảo c{c t|i liệu [8-12], còn để có kiến thức hệ thống v| đầy đủ hơn về mạng nơron thì có thể tham khảo c{c t|i liệu [13-15].

Mặc dù đã được dùng l|m b|i giảng từ nhiều năm nay nhưng l| lần đầu xuất bản nên chắc chắn gi{o trình còn nhiều thiếu sót, chúng tôi mong nhận được c{c ý kiến góp ý để gi{o trình được ho|n thiện hơn.

**Tác giả**

## Chƣơng 1 GIỚI THIỆU

**1.1. HỌC MÁY LÀ GÌ?**

### 1.1.1. Khái niệm học máy

Kh{i niệm học có nghĩa rộng giống như sự thông minh, bao gồm cả qúa trình và khó có một định nghĩa chính xác. Theo nghĩa tự điển, học là quá trình thu nhận kiến thức, kỹ năng do người khác truyền lại hoặc đọc đi, đọc lại, nghiền ngẫm ghi nhớ (học thuộc lòng). Rộng hơn, học bao gồm cả quá trình đúc rút tri thức từ các quan sát, trải nghiệm thực tiễn.

*Học máy* (machine learning) mang hai nghĩa thông dụng: 1) sử dụng m{y tính để khám phá tri thức từ dữ liệu, 2) sự học trong máy (t{c tử: agent). Về phương diện công nghệ, học m{y l| một lĩnh vực của [trí tuệ nh}n tạo,](http://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) trong đó nghiên cứu c{c kỹ thuật x}y dựng v| ph{t triển các chương trình m{y tính có thể thích nghi và "học" từ c{c dữ liệu mẫu hoặc kinh nghiệm. Đến nay, đã có nhiều định nghĩa cho kh{i niệm n|y, tuy nhiên khó có một định nghĩa thỏa đ{ng được mọi người thừa nhận. Định nghĩa sau ph{t triển từ định nghĩa của T. Mitchell cho ta c{ch nhìn to{n học của một chương trình học khi nghiên cứu, thiết kế.

***Định nghĩa1.1***. Một chương trình m{y tính được gọi l| học từ dữ liệu/kinh nghiệm **E** đối với lớp nhiệm vụ **T** v| độ đo mức thực hiện **P** nếu việc thực hiện c{c nhiệm vụ **T** của nó khi đo bằng **P** được cải tiến nhờ dữ liệu hoặc kinh nghiệm **E**.

Theo định nghĩa n|y, người ta cần tối ưu hóa độ đo thực hiện P dựa trên ph}n tích dữ liệu/ kinh nghiệm E để tìm c{ch thực hiện nhiệm vụ T tốt nhất. Để hiểu rõ hơn, ta hãy l|m quen với một số ví dụ v| một số vấn đề liên quan.

### Một số ví dụ

***Ví dụ 1****.* Phân tích dữ liệu b{n lẻ của siêu thị

Hằng ng|y c{c siêu thị b{n ra một lượng lớn những mặt h|ng phong phú v| lưu lại c{c hóa đơn thanh to{n (bản sao giỏ h|ng). Từ c{c dữ liệu b{n lẻ có được, ta có thể ph}n tích c{c giỏ h|ng để tiên đoán được một kh{ch hàng mua mặt h|ng A thì sẽ mua mặt h|ng B với x{c suất bao nhiêu? Nếu x{c suất n|y l| lớn thì ta nên xếp c{c mặt h|ng n|y gần nhau, như thế tiện cho kh{ch h|ng v| lượng h|ng b{n được cũng tăng lên so với việc để kh{ch h|ng phải tìm kiếm khắp nơi.

Rộng hơn, nếu có mô hình ph}n tích tốt, ta cũng có thể dự đo{n được lượng h|ng cần đ{p ứng trong thời gian tới, xu thế sở thích của kh{ch h|ng, trên cơ sở đó có được quyết s{ch thích ứng. Trong ví dụ n|y T l| dự b{o, E l| dữ liệu b{n lẻ lưu trữ v| P l| độ chính x{c của kết quả dự b{o.

***Ví dụ 2***. Đối s{nh v}n tay

B|i to{n đối s{nh v}n tay bắt nguồn từ hai b|i to{n truy nguyên v| x{c thực v}n tay. Trong bài toán truy nguyên, người ta phải đối s{nh một ảnh v}n tay thu được khi điều tra với c{c ảnh v}n tay trong kho lưu trữ để x{c định xem có v}n tay n|o trong kho lưu trữ l| do cùng một ngón lăn ra với ảnh điều tra không. Trong b|i to{n x{c thực, người ta cần x{c minh ảnh v}n tay đăng nhập (ta cũng sẽ gọi l| điều tra) có đúng l| cùng ngón sinh ra với ảnh đã đăng ký hay không? Cả hai bài toán n|y được đưa về b|i to{n đối s{nh cặp ảnh v}n tay điều tra *Iq* với ảnh lưu trữ *It* để trả lời xem chúng cùng hay khác ngón sinh ra. Để x}y dựng chương trình đối s{nh v}n tay, người ta cần một tập dữ liệu bao gồm c{c cặp ảnh do cùng ngón v| kh{c ngón sinh ra. Dựa trên tập dữ liệu n|y, một thuật to{n được {p dụng để x}y dựng chương trình. Người đọc dễ d|ng x{c định được c{c th|nh phần T, E, P cho b|i to{n n|y. Việc giải b|i to{n sẽ được đề cập ở mục 1.3

***Ví dụ 3***. Tìm đường đi ngắn nhất cho robot

Một mạng lưới gồm *n* trạm hoạt động tự động, khoảng c{ch giữa chúng kh{c nhau. Định kỳ, một robot cần đi kiểm tra c{c trạm n|y một lần. Giả sử Robot ghi nhớ được c{c trạm đã qua v| độ d|i đường đi giữa chúng, biết được c{c trạm cần kiểm tra tiếp. Khi đó qua từng trạm nó sẽ tìm đường đi tới trạm tiếp theo. Nếu có chiến lược học tốt, c|ng ng|y robot sẽ tìm được đường đi ngắn hơn, thậm chí l| tối ưu. Trong trường hợp n|y, P v| E tương ứng l| độ d|i v| c{c đường đi kiểm tra đã tìm được, T l| tìm đường đi kiểm tra.

B|i to{n n|y được tổng qu{t hóa như sau. Cho một hệ thống gồm một tập hữu hạn trạng th{i S, với mỗi trạng th{i trong S có một tập t{c động có thể *A*(*si*) để khi hệ ở trạng th{i n|y v| chọn t{c động *a A*( ) thì hệ chuyển sang trạng th{i mới v| mất chi phí nhất định. Ban đầu hệ ở trạng th{i v| kết thúc ở trạng th{i

thuộc tập F, ta cần tìm chuỗi t{c động sao cho chi phí tích lũy nhỏ nhất.

**1.1.2. Tại sao cần nghiên cứu học máy?**

Sự th}m nhập mạnh mẽ của công nghệ thông tin kinh tế, xã hội công nghệ tri thức ph{t triển v| tạo nên nhu cầu ứng dụng rộng rãi. Sau đ}y l| một số phạm vi nghiên cứu, ứng dụng điển hình:

* X}y dựng c{c hệ nhận dạng mẫu dùng cho c{c thiết bị nghe nhìn cho robot v| trong lĩnh vực tự động hóa, nhận dạng chữ viết tay, chuyển đổi c{c b|i nói th|nh văn bản, ph}n tích ảnh tự đông<
* Tạo ra c{c chương trình m{y tính có thể hoạt động thích nghi với môi trường thay đổi hay thực hiện c{c nhiệm vụ mà ban đầu chưa x{c định rõ, chẳng hạn, hệ l{i tự động (máy bay, ôtô, t|y thủy<), trò chơi hay các điều khiển robôt đa năng.
* Khám ph{ tri thức từ dữ liệu, đặc biệt l| c{c cơ sở dữ liệu lớn, để trợ giúp ra quyết định. Chẳng hạn, ph}n tích thị trường, chẩn đo{n bệnh của bệnh nh}n v| x{c định phương {n điều trị nhờ ph}n tích c{c bệnh {n lưu trữ<

#### 1.1.3. Một số lĩnh vực liên quan

Trong mấy chục năm qua, c{c nghiên cứu khoa học v| ứng dụng của học m{y ph{t triển nhanh, kết hợp các tiến bộ của nhiều lĩnh vực khác. Sau đ}y l| c{c lĩnh vực góp phần quan trọng cho nghiên cứu học m{y:

* Lý thuyết x{c suất v| thống kê: L| tiền th}n của lĩnh vực học m{y, trong đó, cho phép suy luận (inference) từ quan s{t cụ thể để có kết luận kh{i qu{t nhờ th|nh tựu của giải tích ngẫu nhiên.
* Mô hình thần kinh sinh học. Việc nghiên cứu cơ chế hoạt động, xử lý phi tuyến v| cấu tạo hệ thần kinh sinh học nói chung cho phép tạo nên c{c mô hình v| thuật to{n phỏng sinh học, đặc biệt l| c{c mạng nơron.
* Lý thuyết độ phức tạp tính to{n. Cho phép ước lượng độ phức tạp của c{c nhiệm vụ học đo qua c{c ví dụ đ|o tạo, số lỗi v| c{c thủ tục tính to{n...
* Lý thuyết điều khiển thích nghi. C{c thủ tục học để điều khiển qu{ trình nhằm tối ưu ho{ mục đích định trước hay học c{ch đo{n c{c trạng th{i tiếp theo của qu{ trình điều khiển<
* T}m lý học: Cho phép mô phỏng c{c đ{p ứng thực tế của con người, x}y dựng c{c mô hình xử lý hiệu quả, chẳng hạn, học tăng cường.
* C{c mô hình tiến hóa. Việc nghiên cứu c{c mô hình tiến hóa cho phép chúng ta đưa ra c{c thuật to{n học mô phỏng tự nhiên như: thuật to{n di truyền (GA), tối ưu đ|n kiến (ACO), tối ưu bầy đ|n (PSO), hệ miễn dịch nh}n tao (AIS)<.

#### 1.1.4. Các bài toán học thiết lập đúng đắn

B|i to{n học được cho l| thiết lập đúng khi thực sự có thể cải tiến được P qua kinh nghiệm E. Thông thường mô hình to{n học để x}y dựng thuật to{n cho một b|i to{n học đòi hỏi phải đúng đắn theo Hadamard. Trong c{c b|i to{n thực tế, [Hadamard](http://en.wikipedia.org/wiki/Jacques_Hadamard) cho rằng một mô hình to{n học ứng dụng được xem l| thiết lập đúng đắn (well-posed problem) nếu nó có c{c tính chất:

1. Luôn tồn tại lời giải
2. Chỉ có duy nhất một lời giải
3. Khi c{c điều kiện ban đầu thay đổi ít thì *lời giải* cũng thay đổi ít.

Tuy nhiên, trong nhiều b|i to{n, điều kiện duy nhất một lời giải nhiều khi khó đ{p ứng. Trong trường hợp đó người ta hay dùng phương ph{p chính quy hóa (hiệu chỉnh h|m mục tiêu) để b|i to{n trở nên thiết lập đúng đắn.

B|i to{n học phải được x{c đính đúng đắn dựa trên việc x{c định rõ nhiệm vụ cụ thể, độ đo việc thực hiện v| nguồn dữ liệu/ kinh nghiệm.

Phương ph{p thông dụng nhất để đưa ra thuật to{n cho c{c b|i to{n học l| x}y dựng một mô hình to{n học phụ thuộc c{c tham số v| dùng dữ liệu hoặc kinh nghiệm đã có để x{c định gi{ trị thích hợp cho c{c tham số n|y.

### 1.2. MỘT SỐ LỚP BÀI TOÁN ỨNG DỤNG ĐIỂN HÌNH

Trong mục trên, ta đã l|m quen với 3 ví dụ về b|i to{n học m{y cụ thể. C{c ứng dụng của học m{y rất đa dạng, sau đ}y điểm qua một số lớp b|i to{n ứng dụng thường gặp.

*Học các kết hợp*

Trong nghiên cứu thị trường, người ta thường quan t}m tới các sự kiện X v| Y cùng xảy ra v| ước lượng x{c suất có điều kiện P(Y/X) để Y xảy ra với điều kiện X xảy ra. Công việc n|y gọi l| *học các kết hợp*. Chẳng hạn, trong ví dụ 1 mục trước, nhà cung cấp cần ph}n tích giỏ h|ng của kh{ch hàng qua c{c hóa đơn để tìm x{c suất P(Y/X) để nếu khách mua sản phẩm X thì cũng mua sản phẩm Y, nhờ đó người ta có thể dự đo{n được khả năng một kh{ch hàng khi mua sản phẩm X thì sẽ mua sản phẩm Y.

*Phân loại mẫu*.

C{c đối tượng thuộc tập ***X*** được ph}n th|nh *k* lớp dựa trên một tập con ***D*** đã biết nhãn. Chẳng hạn, c{c chữ số viết tay có 10 lớp, còn b|i to{n đối s{nh v}n tay thuộc loại hai lớp: trùng với ảnh lưu trữ hay không. Bài toán ph}n loại thuộc về học có giám sát v| l| b|i to{n thường gặp nhất trong ứng dụng.

Nhiều khi, người ta dùng từ *phân lớp* (classify) để thay cho *phân loại* (categorize), mặc dù thuật ngữ ph}n lớp có nghĩa rộng hơn, bao gồm cả *phân cụm* (cluster). Về sau, khi không g}y nên nhầm lẫn, hai từ n|y có thể dùng thay cho nhau. Một ứng dụng quan trọng của b|i to{n n|y l| ph}n tích hồ sơ người vay để đ{nh gi{ rủi ro trong hoạt động tín dụng, trong đó dựa trên c{c yếu tố đặc trưng về khả năng tài chính của người vay, ng}n h|ng cần đo{n nhận xem kh{ch hang có khả năng trả nợ đúng hạn không để cho vay.

*Hồi quy hàm số*

Trong thực tiễn, ta thường phải x{c định gi{ trị h|m số tại những điểm chưa biết dựa trên gi{ trị h|m đã biết tại một số điểm. Bài toán này phát biểu như sau. Có một h|m chưa biết , nhưng biết được tập trong gồm N đối tượng quan s{t được:

*yj*= , (1.1)

trong đó l| nhiễu trắng (c{c biến ngẫu nhiên độc lập, cùng ph}n bố v| có kỳ

vọng bằng không)

,

. Ta cần tìm hàm gần đúng của

cho c{c đối tượng kh{c của ***X.*** Hàm *g* sẽ được gọi l| h|m hồi quy của *f*. Nếu không quan t}m tới ph}n bố nhiễu thì ta gọi l| b|i to{n xấp xỉ hàm. Khi các phân bố rộng trên tập X v| đòi hỏi:

*g*( (1.2)

)

=

thì b|i to{n xấp xỉ n|y gọi l| b|i to{n nội suy v| h|m *g* sẽ được gọi l| h|m nội suy của hàm *f*.

*Học không giám sát*

C{c b|i to{n trên thuộc loại học có giám sát, trong đó ta biết được nhãn của tập dữ liệu quan s{t được. Trong học không gi{m s{t, ta chỉ đơn thuần ph}n tích đặc điểm của tập dữ liệu đế có thông tin. Ba b|i to{n học không có gi{m s{t thường gặp là: ước lượng h|m mật độ, ph}n cụm dữ liệu và dóng h|ng (align) dựa trên cấu trúc..

Trong bài to{n ước lượng h|m mật độ, có một tập mẫu dữ liệu lấy ngẫu nhiên cùng ph}n bố, ta cần dựa trên đó để ước lượng h|m mật độ của ph}n bố n|y.

Trong bài toán phân cụm dữ liệu, người ta chia tập dữ liệu th|nh c{c tập con (cụm) sao cho c{c phần từ trong cùng một cụm thì *giống nhau* hơn c{c phần tử kh{c cụm. Đặc tính *giống nhau* n|y thường được x{c định bởi *khoảng cách*, đối tượng A giống đối tượng B hơn đối tượng C nếu khoảng c{ch từ A đến B nhỏ hơn khoảng c{ch từ A đến C. Khi tập dữ liệu cần xử lý lớn thì việc ph}n cụm cho phép ta giảm thời gian chạy của c{c ứng dụng. Tuy nhiên b|i to{n n|y l| b|i to{n thiết lập không đúng đắn (ill-posed) v| thường không duy nhất nghiệm.

Ph}n tích c{c dữ liệu có cấu trúc x}u/ trình tự (string/sequence) hoặc mạng dẫn đến c{c b|i to{n dóng h|ng trong xử lý ngôn ngữ tự nhiên v| tin sinh học. Việc dóng h|ng c{c trình tự DNA, RNA, Protein v| c{c mạng tương t{c protein cho phép hiểu được c{c tính tương đồng v| kh{c biệt về nhiều đặc điểm sinh học của c{c c{ thể sinh vật v| lo|i.

*Học tăng cường*

Trong nhiều trường hợp, đầu ra của hệ thống l| một chuỗi t{c động. Khi đó mỗi t{c động riêng lẻ không quan trọng m| điều quan trọng l| chuỗi t{c động n|y cần đạt được mục đích định trước. Chẳng hạn, trong c{c trò chơi, một nước đi không thực sự quan trọng m| quan trọng l| chuỗi nước đi đưa đến kết quả thắng, ví dụ 3 nêu ở trên l| trường hợp riêng của loại n|y.

Tương tự như phương thức học nhờ trải nghiệm cuộc sống, người ta có thể tạo ngẫu nhiên nhiều lời giải chấp nhận được v| sau mỗi lần lặp điều chỉnh trọng số định hướng lựa chọn t{c động để c|ng về sau chuỗi t{c động có trọng số cao giúp ta đạt được mục đích cần có.

B|i to{n học tăng cường sẽ khó hơn với c{c b|i to{n chỉ quan s{t được từng phần hoặc cần hợp t{c của nhiều t{c tử (agent) để đạt được đích.

### 1.3. KIẾN TRÚC VÀ THIẾT KẾ MỘT HỆ HỌC

Kiến trúc v| việc thiết kế hệ học kh{ đa dạng tùy theo từng b|i to{n m| c{c vấn đề then chốt phải giải quyết kh{c nhau.

Trong mục n|y ta xét b|i to{n nhận dạng v}n tay v| b|i to{n tìm đường đi tối ưu nhờ học tăng cường để minh hoạ.

#### 1.3.1. Đối sánh vân tay

Trở lại với b|i to{n đối s{nh v}n tay trong ví dụ 2: Dựa trên tập mẫu c{c cặp v}n tay cùng hoặc kh{c ngón sinh ra, ta cần kết luận hai ảnh v}n tay điều tra *Iq* v| lưu trữ *It* có cùng ngón sinh ra hay không? Kiến trúc v| quy trình thiết kế cho hệ n|y mang đặc điểm chung của một hệ nhận dạng mẫu (pattern recognition). Trước hết, ta cần l|m quen với phương ph{p đối s{nh v}n tay dựa trên đặc trưng chi tiết*.*

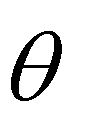
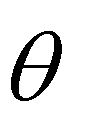
*1.3.1.1 Lược đồ đối sánh dựa trên đặc trưng chi tiết*

C{c điểm kỳ dị trên v}n tay như c{c điểm cụt, rẽ nh{nh, t}m< được gọi l| c{c điểm đặc trưng chi tiết (ĐTCT)*.* Người ta có thể dựa trên tính tương đồng/ kh{c biệt của c{c tập ĐTCT trên hai ảnh v}n tay để kết luận chúng có cùng ngón sinh ra không. Lược đồ của phương ph{p n|y như sau.

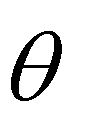
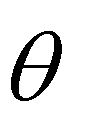
Từ hai ảnh v}n tay, ta x{c định hai tập điểm ĐTCT *Mq*và *M*t tương ứng của *Iq* và

*It*:

*Mq*= {*m1, m2, …, mM*}, *mi* = (*xi,yi, i*) *i = 1,..,M;* (1.3)



*Mt*= {*m1’, m2’, …, mN'*}, *mi*'= (*xi’,yi’, i’*) *i = 1-N*, (1.4) trong đó *mi* l| vectơ điểm ĐTCT trích chọn từ ảnh tương ứng, (*xi,yi*) /(*xi’ yi’*) là tọa độ của *mi* trên mặt phẳng chứa nó còn *i* / *i’* l| hướng v}n tại mi’.

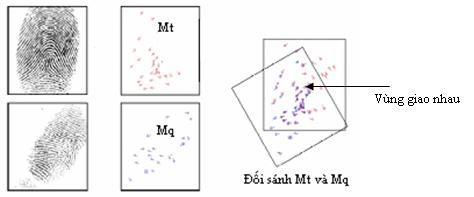


Vì miền sinh ra ảnh *Iq* và *It* không trùng nhau nên nói chung M N. Để đối s{nh, người ta x}y dựng phép biến đổi *T* ít thay đổi tôpô của một ảnh để chồng ảnh *Iq*lên ảnh *It* sao cho c{c điểm ĐTCT tương ứng của chúng cũng *“trùng khớp”* với nhau từng đôi một nếu cùng một ngón sinh ra (xem hình 1.1). Một điểm ĐTCT trên *It* gọi l| *trùng khớp* với một điểm ĐTCT *mk* trên *Iq* nếu sau biến đổi ảnh của rơi v|o một l}n cận đủ bé của *mk* như được mô tả trong hình 1.1. Tuy nhiên, do chất lượng ảnh hoặc hiện tượng biến dạng ngón tay khi lấy dấu tay nên chỉ có *n* cặp điểm *trùng khớp* và chúng được gọi l| cặp điểm tương ứng. Sau khi x}y dựng v| dùng c{c phép biến đổi, độ giống nhau của hai ảnh v}n tay được đặc trưng bằng độ đo tương tự S(*It,Iq*) cho bởi công thức

S(*It,Iq*) = n2/*Mt.Mq*. (1.5)

Nếu độ tương tự n|y nhỏ hơn ngưỡng cho trước thì nói rằng chúng không cùng một ngón sinh ra, ngược lại, ta kết luận cùng ngón. Ngưỡng n|y được x{c định dựa trên tập mẫu đã có.

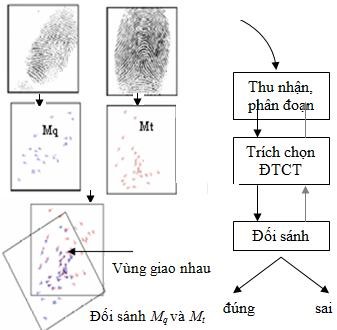
*Mq*



**Hình 1.1**. Xác định các cặp điểm ĐTCT tương ứng giữa hai tập Mt và Mq

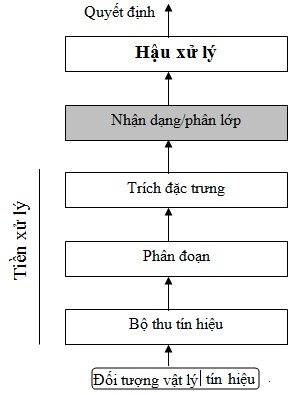
*1.3.1.2. Các thành phần của hệ nhận dạng mẫu*

Kiến trúc của hệ đối s{nh vân tay được mô tả trong hình 1.2, trong đó bao gồm c{c th|nh phần: thu nhận, ph}n đoạn ảnh; trích chọn ĐTCT v| đối s{nh. Sau khi thu nhận ảnh từ chỉ bản hoặc sensor ta cần ph}n đoạn để t{ch phần ảnh rõ để trích chọn c{c điểm ĐTCT, sau đó sử dụng th|nh phần đối s{nh để kết luận chúng cùng ngón sinh ra. Kết luận n|y sẽ được xem xét sử dụng.



**Hình 1.2.** Sơ đồ hệ đối sánh vân tay

Kiến trúc trên l| trường hợp riêng của một hệ nhận dạng mẫu, được kh{i qu{t trong hình 1.3. bao gốm c{c th|nh phần: Tiền xử lý, nhận dạng/ph}n lớp v| hậu xử lý.



**Hình 1.3.** Kiến trúc của một hệ nhận dạng

Trong th|nh phần tiền xử lý có c{c modun: Thu tín hiệu, ph}n đoạn v| trích chọn đặc trưng. Tín hiệu vật lý được xử lý ở bộ thu để có tín hiệu rõ v| t{ch tín hiệu quan t}m ở modun ph}n đoạn trước khi trích chọn đặc trưng để đưa v|o bộ nhận dạng/ph}n lớp.

Bộ nhận dạng/ph}n lớp (về sau gọi gọn l| *nhận dạng*) l| trung t}m của hệ thống. Dựa trên ph}n tích đặc điểm của c{c gi{ trị thuộc tính được trích chọn, bộ n|y đưa ra nhãn cho đối tượng.

Đầu ra của bộ nhận dạng sẽ đưa v|o bộ hậu xử lý xem kết luận đã tốt chưa, cần ph}n tích bổ sung nữa hay không v| đưa ra quyết định cuối cùng.

Như vậy một hệ nhận dạng có kh}u chính: tiền xử lí dữ liệu, ph}n lớp v| hậu xử lí. Trong đó ph}n lớp l| trung t}m của cả hệ thống.

*1.3.1.3. Thiết kế hệ nhận dạng*

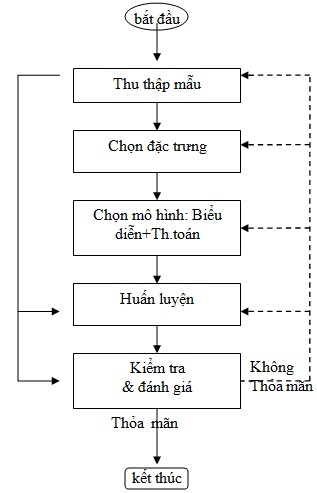
Khi có dự {n x}y dựng bộ nhận dạng, việc đầu tiên l| thu thập tập mẫu hay dữ liệu. Đối với nhiều trường hợp học, cơ sở lý thuyết về lấy mẫu thống kê có vai trò quan trọng để đảm bảo chất lượng của hệ thống. C{c dữ liệu thu được cần có giải ph{p l|m sạch v| l|m gi|u để sử dụng.

Dựa trên đặc điểm b|i to{n v| mô hình dự kiến {p dụng, cần quyết định chọn c{c đặc trưng: bao nhiêu đặc trưng l| đủ, đặc trưng n|o sẽ được chọn.

Kh}u trung t}m l| x}y dựng mô hình bao gồm biễu diễn b|i to{n v| quan trọng nhất, đưa ra thuật to{n giải hiệu quả. Thông thường, một b|i to{n có thể có nhiều mô hình, thuật to{n để giải, cần c}n nhắc để chọn mô hình, thuật to{n thích hợp.

Sau đó, người ta sử dụng dữ liệu v| mô hình để huấn luyện bộ nhận dạng. Vì có nhiều mô hình, thuật to{n giải, hơn nữa, c{c kỹ thuật c|i đặt cũng ảnh hưởng tới hiệu quả của hệ nên cần phải đ{nh gi{, kiểm tra qua thực nghiệm để đ{nh gi{ hiệu quả để có quyết định cuối cùng. Bởi vì c{c bước trên liên quan mật thiết với nhau nên việc thiết kế luôn có mối liên hệ ngược (Feedback) khi thực hiện v| điều chỉnh mỗi bước. Qu{ trình n|y chính l| qu{ trình thiết kế hệ, trong đó trung t}m l| x}y dựng v| c|i đặt thuật to{n học.

Hình 1.4 biểu diễn qu{ trình x}y dựng một bộ nhận dạng trong đó c{c bước có thể lặp lại cho tới khi đạt yêu cầu.



**Hình 1.4.** Quá trình thiết kế bộ nhận dạng

#### 1.3.2. Tìm đƣờng đi tối ƣu cho robot

Ta quay lại với b|i to{n tìm đường đi cho robot trong ví dụ 3, khi biểu diện mạng lưới c{c trạm bởi một đồ thị đầy đủ với mỗi nút l| một trạm, c{c cạnh nối c{c đỉnh (*i,j*) có trọng số l| độ d|i đường đi giữa c{c đỉnh n|y (ban đầu robot chưa biết) thì bài to{n có thể ph{t biểu như sau: Tìm chu trình Hamilton có độ d|i ngắn nhất cho robot đi kiểm tra c{c trạm. Ngay cả khi robot biết trước độ d|i c{c cạnh thì b|i to{n n|y cũng thuộc loại NP-khó nên không giải đúng được khi số trạm lớn.

Giả sử robot thực hiện nhiều lần tìm kiếm nhanh đường đi (không thực hiện kiểm tra) bằng việc lựa chọn chọn ngẫu nhiên c{c đỉnh tiếp theo, khi đó thuật to{n đơn giản sau có thể {p dụng.

*1.3.2.1. Thuật toán học tăng cường để tìm gần đúng đường đi tối ưu*

Trong thuật to{n n|y, robot sẽ tìm c{ch đ{nh dấu c{c cạnh thuộc đường đi tốt nhất bằng chỉ số nghịch đảo độ d|i đường đi n|y, c{c cạnh không thuộc đường đi ngắn nhất sẽ đ{nh dấu bằng nghịch đảo độ d|i đường đi d|i nhất. Điểm thú vị ở đ}y l| độ d|i mỗi cạnh chỉ góp phần v|o độ d|i đường đi m| nó thuộc v|o chứ không có tính quyết định v| robot chỉ tìm lời giải gần đúng, như vậy c{c chỉ số được g{n cũng chỉ l| “*xấp xỉ*”. Thuật to{n thực hiện như sau.

Với số tự nhiên *k* v| số *L0* cho trước, khởi tạo chỉ số cạnh bằng *L0*. Robot thực hiện lặp việc x}y dựng đường đi ngẫu nhiên v| điều chỉnh trọng số cạnh nhờ thủ tục bước ngẫu nhiên sau.

Khởi tạo từ đỉnh xuất ph{t *u0*, x}y dựng tuần tự c{c đỉnh tiếp theo: nếu đang ở đỉnh *i* robot chọn đỉnh *j* tiếp sau một c{ch ngẫu nhiên với x{c suất l|

, (1.6)

trong đó l| chỉ số của cạnh (*i,j* ) tương ứng còn *J(i)* l| nhưng trạm chưa đến.

Qu{ trình tiếp tục cho đến khi đi hết c{c đỉnh thì x{c định độ d|i đường đi v| c{c cạnh đã qua. Cứ sau một chu kỳ *k* lần thì robot thực hiện so s{nh để tìm đường đi ngắn nhất. Ký hiệu *Lbest*và *Lworst*tương ứng l| nghịch đảo đường đi ngắn nhất v| d|i nhất tìm được, robot sẽ điều chỉnh chỉ số cho c{c cạnh theo công thức:

(1.7)

l| hệ số cho trước, l| đại lượng cho bởi công thức:

trong đó

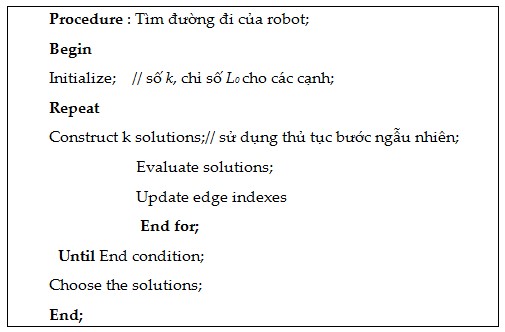
(1.8)

Qu{ trình học thường dừng sau số bước lặp chọn trước. Sau khi huấn luyện, robot sẽ chọn cạnh có chỉ số cao nhất cho đường đi tới đỉnh tiếp theo khi thực hiện kiểm tra. Thuật to{n n|y được đặc tả trong hình 1.5.

Người ta chứng minh được sau một số lần lặp đủ nhiều, lời giải tìm được có độ d|i gần với lời giải tối ưu v| chỉ số tương ứng trên cạnh của nó gần với chỉ số mong muốn. Hơn nữa, người ta có thể cố định trước gi{ trị chỉ số *Lbest*và *Lworst* m| không cần phải thay đổi theo độ d|i đường đi trong từng bước lặp, khi chọn thích hợp thì có thể cho kết quả tốt hơn.

*1.3.2.2. Thiết kế hệ học cho bài toán tổng quát*

B|i to{n tìm đường đi tối ưu minh họa cho c{c vấn đề khi thiết kế hệ học cho lớp b|i to{n tìm chuỗi t{c động tối ưu trong hệ thống hữu hạn trạng th{i nêu ở trên. Sau đ}y l| một số vấn đề cần lưu ý khi thiết kế hệ học:



**Hình 1.5**. Đặc tả thuật toán tìm đường đi robot

1) Chọn phương thức tạo sinh kinh nghiệm đ|o tạo

Kinh nghiệm đ|o tạo l| tập c{c mẫu (lời giải) ta có, phương thức tạo ra /thu nhận mẫu n|y l| vấn đề đầu tiên gặp phải khi thiết kế hệ. Ba vấn đề sau, chúng ta phải đối diện.

*Thứ nhất.* Cần chọn kiểu tạo sinh kinh nghiệm đ|o tạo để máy có thể sinh ***trực tiếp*** hay ***gián tiếp***vàcác mẫu tạo sinh sau có thể có liên hệ với lời giải trước hoặc không. C{c mẫu trong thuật to{n nêu trên được sinh trực tiếp, còn trong sinh gián tiếp thì mẫu có thể lấy từ c{c biên bản thực hiện trước đó hoặc có thêm thông tin hướng dẫn bổ sung.

*Thứ hai.* Chọn mức độ điều khiển của bộ học đối với chuỗi mẫu đ|o tạo. Trong trường hợp trên, chuỗi mẫu được sinh tự động theo thuật to{n m| không có tương tác nào thêm của người thiết kế. Nhiều trường hợp, bộ học có thể trao đổi với người thiết kế (thầy) để định hướng tạo mẫu tiếp theo.

*Thứ ba.*  L|m thế n|o để ph}n bố thực hiện thí nghiệm dựa trên đ{nh gi{ độ đo đích thực P qua c{c độ đo trên mẫu đ|o tạo. Ở trên, ta tìm kiếm c{c mẫu mới v| đ{nh gi{ theo định hướng quanh lời giải tốt nhất tìm được. Chúng ta cúng có thể kh{m ph{ v| đ{nh gi{ theo hướng kh{c.

1. Chọn h|m đích

Trong trường hợp trên, h|m đích l| độ d|i đường đi, để dễ c|i đặt thuật to{n, ta xét nghịch đảo độ d|i để đưa về b|i to{n tìm cực tiểu. Nhiều trường hợp, độ đo thực hiện không tường minh, chẳng hạn, khi thiết kế trò chơi với độ đo l| tỷ lệ thắng thì mục tiêu n|y còn tùy thuộc trình độ đối thủ. Ngo|i ra ta có mục tiêu thứ cấp (như l| tham số) có thể chọn bằng nhiều c{ch như đã nêu trên.

1. Chọn biểu diễn v| xấp xỉ tham số

H|m đích/tham số đạt được trong qu{ trình học thường l| gi{ trị gần đúng của h|m đích/tham số đặt ra. Ở trên cho thấy ta có nhiều c{ch chọn chỉ số *Lbest*và *Lworst* và cách chọn tự nhiên dựa trên độ dài cạnh chưa hẳn tốt, việc cập nhật chỉ số n|y cũng có nhiều cách khác nhau, chẳng hạn, đơn giản nhất là chọn tham số khác nhau. Khi đó, sau mỗi bước lặp ta có gi{ trị chỉ số kh{c nhau.

Qu{ trình trên thực chất l| ta thiết lập v| chọn thuật to{n giải b|i to{n tối ưu

1. X}y dựng hệ học

Chất lượng của hệ học được đo bằng độ đo thực hiện, nó được quyết định bởi nhiều yếu tố, trong đó quyết định nhất l| thuật to{n học. Ngo|i độ đo thực hiện, thời gian huấn luyện cũng l| yếu tố được quan t}m khi đ{nh gi{ thuật to{n. Thời gian chạy n|y tùy thuộc v|o kỹ thuật c|i đặt khi x}y dựng bộ học. Chẳng hạn, nếu ta chọn kiểu dữ liệu kh{c nhau thì thời gian xử lý sẽ kh{c nhau v| do đó thời gian huấn luyện cũng kh{c nhau.

Việc đ{nh gi{ hiệu năng của hệ chỉ thực hiện được qua thực nghiệm. Vì vậy phải đảm bảo tính kh{ch quan v| tin cậy được dựa v|o c{c kết luận thống kê

#### KẾT LUẬN

Học m{y nghiên cứu l|m sao x}y dựng c{c chương trình m| chúng có thể cải tiến chất lượng thực hiện nhiệm vụ dựa trên dữ liệu hoặc kinh nghiệm đã có. Sau đ}y l| một số điểm cần lưu ý.

* C{c thuật to{n đã chứng tỏ rất có ý nghĩa trong thực tế ở lĩnh vực ứng dụng kh{c nhau như: Kh{m ph{ tri thức từ dữ liệu (Data mining), c{c công cụ hoạt động tự động có thể thích nghi với môi trường thay đổi, nghiên cứu y-sinh học<
* Học m{y kết hợp ý tưởng v| kỹ thuật từ nhiều lĩnh vực kh{c nhau như x{c suất v| thống kê, trí tuệ nh}n tạo, t}m lý v| thần kinh học, lý thuyết điều khiển, lý thuyết tính to{n<.
* B|i to{n học phải được x{c định đúng đắn dựa trên việc x{c định rõ nhiệm vụ cụ thể, độ đo việc thực hiện v| nguồn dữ liệu/ kinh nghiệm.
* Việc thiết kế hệ học bao gồm việc thu thập xử lý dữ liệu/ tạo sinh kinh nghiệm, x{c định h|m đích phản {nh độ đo thực hiện, quan trọng nhất l| một thuật to{n học để học h|m đích dựa trên dữ liệu/kinh nghiệm đ|o tạo.

#### BÀI TẬP

1. X{c định T, E, P v| đặc điểm thiết kế cho các bài toán sau:
   1. Rô bot học l{i trên đường 4 l|n đường học l{i xe robot
   2. Nhận dạng chữ viết tay
   3. Dựa trên c{c bệnh {n đã có, chẩn đo{n bệnh v| đề xuất ph{c đồ điều trị cho bệnh nh}n
2. Giả sử anh hay chị sẽ x}y dựng hệ nhận dạng chữ viết tay/ mặt người. Hãy mô tả hệ nhận dạng v| ph{c họa thiết kế.
3. So s{nh c{c kh}u thiết kế hệ học chơi cờ v| hệ đối s{nh v}n tay, tìm đường đi tối ưu của robot.

**Chƣơng 2**

## HỌC CÓ GIÁM SÁT

Chương n|y giới thiệu phương ph{p học có giám sát từ tập mẫu quan s{t được, bắt đầu từ các khái niệm chung về học quy nạp một bài toán phân lớp và các thuật to{n đơn giản để học một khái niệm, sau đó xét b|i to{n hồi quy với thuộc tính và đầu ra nhận giá trị liên tục, cuối cùng thảo luận một số vấn đề liên quan như khuynh hướng, lỗi của giả thuyết và chiều Vapnik-Chevonekis.

### 2.1 CÁC BÀI TOÁN VÀ VÍ DỤ ĐIỂN HÌNH

Hiểu theo nghĩa rộng, học có gi{m s{t (supervised learning) dùng để chỉ c{c b|i to{n học m| trong dữ liệu/kinh nghiệm hay qu{ trình học có c{c thông tin định hướng để cải thiện độ đo thực hiên.

Hai b|i to{n học có gi{m s{t thường gặp nhất l| ph}n lớp v| hồi quy h|m số dựa trên một tập mẫu đã biết nhãn. Tùy theo ngữ cảnh của b|i to{n cụ thể, chúng có tên gọi kh{c nhau: học kh{i niệm/ ph}n lớp/ph}n loại, hồi quy/ xấp xỉ/nội suy h|m số. Mô tả tổng qu{t cho c{c b|i to{n n|y như sau.

Cho một tập mẫu quan s{t được D= (*xk* , *yk* ), trong đó D*x*= l| tập con của tập đối tượng X, còn l| nhãn trong tập Y của c{c đối tượng tương ứng, ta cần tìm nhãn cho c{c đối tượng ***x*** mới trong X. Trước khi đi v|o c{c phương ph{p học, ta l|m quen với c{c b|i to{n điển hình.



*N*

*k*

1



#### 2.1.1. Bài toán học khái niệm

Nhiều trường hợp ta cần học một kh{t niệm tổng qu{t như: tr}u, bò, mèo, chó, ôtô, xe m{y<từ c{c trường hợp quan s{t cụ thể v| ta gọi l| học kh{i niệm. Trong c{c trường hợp quan s{t được, ta biết c{c đối tượng/ sự kiện (ví dụ) có phù hợp (dương tính) với kh{i niệm cần học hay không (}m tính), từ đó m| x{c định xem c{c đối tượng/sự kiện kh{c l| phù hợp hoặc không với kh{i niệm cần học.

Để giải b|i to{n n|y ta cần một *mô hình biểu diễn* c{c đối tượng nhờ đặc điểm của chúng. Thông thường, mỗi đối tượng được đặc trưng bởi tập gi{ trị của c{c thuộc tính tương ứng. Nhãn của mỗi đối tượng có gi{ trị boolean: bằng 1 khi đối tượng dương tính v| bằng 0 nếu }m tính. Như vậy nếu ký hiệu tập đối tượng l| ***x*** v| được biễu diễn bởi *n* thuộc tính thì mỗi ***x*** có biễu diễn là trong đó nhận gi{ trị trong tập gi{ trị tương ứng của thuộc tính *A*i. Từ các mẫu đã biết nhãn, ta cần x{c định hàm logic sao cho khi đã biết nhãn thì c( ) trùng với nhãn n|y.

***Ví dụ 1*** *(giá trị thuộc tính rời rạc).* Giả sử ta học kh{i niệm: *thời tiết của ngày bạn A thích chơi môn thể thao dưới nước.* Trước hết, ta cần có mô hình biểu diễn cho đặc điểm thời tiết ảnh hưởng tới quyết định chơi thể thao của bạn A.

*Mô hình biễu diễn.* Ta chọn biễu diễn thời tiết với c{c thuộc tính cho trong bảng 2.1: tiết trời, nhiệt độ, độ ẩm, gió, nước (nhiệt độ) với c{c gi{ trị có thể nhận được cho tương ứng trong ngoặc nhọn.

**Bảng 2.1.** Các thuộc tính biểu thị thời tiết ngày ưa thể thao

-Tập mẫu X: C{c ng|y với gi{ trị thuộc tính có thể nhận.

A1: Tiết trời: ,nắng, nhiều m}y, mưa}

A2: Nhiệt độ: ,ấm , lạnh}

A3: Độ ẩm: ,trung bình, cao}

A4: Gió: ,mạnh, yếu}

A5: Nước: ,ấm, lạnh}

A6: Dự b{o: ,không đổi, đổi}

Dựa trên tập dữ liệu thu được từ c{c ng|y bạn A chơi hoặc không (chẳng hạn, được cho trong bảng 2.2.) ta sẽ dự đo{n bạn A có đi chơi hoặc không khi biết gi{ trị c{c thuộc tính liên quan của một ng|y mới. Ta sẽ tìm hàm dự báo cho dưới dạng tập luật: {nếu là ,…, là thì c( )= *y*}.

**Bảng 2.2.** Các ví dụ về những ngày chơi hoặc không của A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tiết trời | nhiệt độ | độ ẩm gió | nước | dự b{o thích chơi? |
| 1  2  3  4 | nắng nắng mưa nắng | ấm ấm  lạnh ấm | trung bình mạnh cao mạnh cao mạnh cao mạnh | ấm ấm ấm  lạnh | không đổi có không đổi có  đổi không đổi có |

Trong ví dụ trên, c{c thuộc tính nhận gi{ trị rời rạc, ví dụ sau minh họa b|i to{n có gi{ trị thuộc tính liên tục.

***Ví dụ 2*** *(giá trị thuộc tính liên tục).*

B}y giờ giả sử ta cần học kh{i niệm *người có sức khỏe tốt* dựa trên xét nghiệm hai chỉ số sinh hóa *x1* và *x2* nhận gi{ trị thực với tập mẫu quan s{t được cho trong hình }m tính (sức khỏe không tốt).

2.1

, trong đó đối tượng

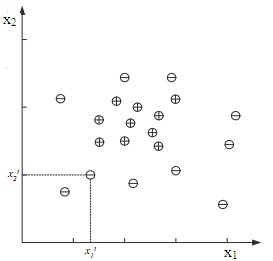
l| dương tính (biểu thị sức khỏe tốt) còn đối tượng

là

**Hình 2.1.**

Biễu thị sức khỏe qua

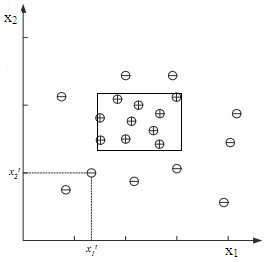
hai thuộc tính sinh hóa



Một lời giải khả dĩ cho trường hợp n|y l|: nếu thì ngược lại thì

, tức l| những đối tượng có đặc trưng thuộc hình chữ

nhật được minh họa trong hình 2.2 sẽ có sức khỏe tốt.



và

**Hình 2.2.** Một miền xác định người sức khỏe tốt dựa trên hai chỉ số sinh hóa

#### 2.1.2. Bài toán học nhiều lớp

B|i to{n học kh{i niệm có thể xem l| b|i to{n học 2 lớp, lớp dương tính v| lớp }m tính. B|i to{n ph}n lớp tổng qu{t từ tập mẫu quan s{t được ph{t biểu như sau.

Có k lớp *C1,…, Ck* và tập quan sát , ;

, trong đó là tập đối tượng, ta cần phân lớp cho c{c đối tượng

mới ***x*** trong . Chẳng hạn, ***D*** l| tập c{c chữ viết tay mẫu còn l| tập c{c chữ c{i, ta cần đo{n nhận c{c chữ c{i mới. B|i to{n n|y cũng có thể xem gồm *k* bài toán học kh{i niệm, mỗi kh{i niệm l| một lớp. Về sau, nếu không có gì đặc biệt, ta chỉ xét trường hợp hai lớp v| người đọc có thể tổng qu{t hóa cho trường hợp nhiều lớp như l| b|i tập thực h|nh.

#### 2.1.3. Bài toán nội suy và hồi quy

Ta trở lại với *bài toán nội suy* đã được nói đến trong mục 1.2: Có một h|m chưa biết trong :

, nhưng bi

ế

t đư

ợ

c t

ậ

p quan sát

trong đó

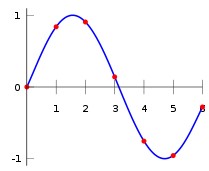
, (2.1)

Ta cần x{c định giá trị hàm *f (****x****)* tại c{c điểm chưa quan s{t

được *x=(x1,…,xn)*. Các giá trị *f (x)* này sẽ được x{c định nhờ tìm một hàm sao cho:

(2.2)

và *g(****x****)* được dùng để tính gần đúng *f (****x****)*. Hàm *g* sẽ gọi l| h|m nội suy của *f*. Nếu ***x*** *thuộc miền* có c{c điiểm {*xj*} thì b|i to{n x{c định *f(****x****)* là *bài toán nội suy* (interpolation) còn *x* ra ngo|i miền thì gọi l| *ngoại suy*(extrapolation). Hình 2.3 minh họa đồ thị một h|m nội suy 2 biến dựa trên 7 mẫu quan s{t được.



**Hình 2.3.** Một hàm nội suy từ 7 ví dụ quan sát

Khi dữ liệu quan s{t được có nhiễu thì đòi hỏi tìm h|m hồi quy thỏa mãn chặt biểu thức (2.2) không cần thết m| người ta chỉ đòi hỏi gần đúng v| tốt theo một tiêu chuẩn n|o đó.

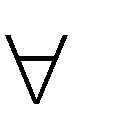
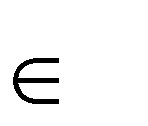
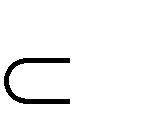
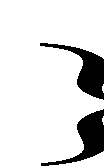
B}y giờ ta ph{t biểu b|i to{n học có gi{m s{t cho b|i to{n học quy nạp từ c{c mẫu quan s{t được.

### 2.2. HỌC QUY NẠP

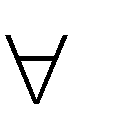
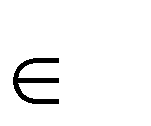
Học h|m tổng qu{t hóa từ c{c mẫu quan s{t được l| trung t}m của học m{y v| được gọi l| học quy nạp. Trước hết ta cần ph{t biểu to{n học cho b|i to{n n|y.

#### 2.2.1. Phát biểu bài toán

Cho ***X*** l| tập đối tượng, ***Y*** l| tập nhãn v| một {nh xạ *c*: ***X→Y*** chưa biết nhưng có tập mẫu đã biết ***D*** = (*x*,*c*(*x*)) / *x DX X* , tìm cần tìm {nh xạ *h*: ***X→Y*** sao cho:



*h(****x****) = c(****x****) x D X .* (2.3)



Ta xem {nh xạ *h* tìm được l| {nh xạ *c* cần tìm. Tập ***D*** còn được gọi l| tập huấn luyện hay tập đ|o tạo.

Dễ d|ng thấy rằng c{c b|i to{n học kh{i niệm, ph}n lớp v| nội suy nêu trên l| trường hợp riêng của b|i to{n n|y, còn khi xét biểu thức (2.3) theo nghĩa gần đúng thì nó bao gồm cả b|i to{n hồi quy.

#### 2.2.2. Một số khái niệm cơ bản

*Biễu diễn đối tượng*

Thông thường, c{c đối tượng trong ***X*** được biễu diễn qua c{c gi{ trị trong tập c{c thuộc tính được chọn ,*A1,…,An*} hay còn gọi l| c{c đặc trưng của chúng. Mỗi thuộc tính *Ai*của đối tượng nhận gi{ trị trong tập gi{ trị *Vi*tương ứng. Với mỗi tập đối tượng ***X*** có thể có nhiều c{ch biễu diễn với c{c tập đặc trưng phù hợp. Việc trích v| chọn đặc trưng vượt ra khỏi khuôn khổ của gi{o trình n|y, v| ta chỉ để cập tới một v|i điểm then chốt m| thôi. Mỗi c{ch biễu diễn với c{c đặc trưng được chọn ta gọi l| một ***mô hình biễu diễn***.

Cho mô hình biễu diễn với tập đặc trưng ,*A1,…,An*}, khi đó mỗi đối tượng ***x*** trong ***X*** ứng với một một vectơ v| ký hiệu l| ***x*** = <*x1,….,xn*>, trong đó c{c *xk* có thể nhận gi{ trị thực hoặc gi{ trị định danh tùy theo đặc trưng của nó.

*Không gian giả thuyết*

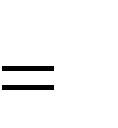
Mỗi {nh xạ *h:* ***X****→****Y*** có thể xem xét l|m lời giải (không nhất thiết thỏa mãn (2.3)) cho b|i to{n. Tức l| nó x{c định một c{ch g{n nhãn cho c{c đối tượng trong ***X*** và được gọi l| một giả thuyết. Vì số {nh xạ từ X v|o Y có lực lượng l| , trong đó chỉ số phần tử của tập *S*, nên sẽ có rất nhiều giả thuyết khi ***X*** và ***Y*** l| tập lớn. Người ta thường tìm gần đúng {nh xạ *c* trong tập con **H** n|o đó của tập tất cả c{c {nh xạ mà ta có thể dự đo{n h|m đích thuộc v|o. Tập ***H*** n|y sẽ được gọi l| không gian giả thuyết ((hypothesis), trong trường hợp n|y, mỗi *h* ***H*** sẽ gọi l| một giả thuyết nếu không xảy ra nhầm lẫn. Thuật to{n tìm *h* sẽ gọi l| *thuật toán học*, mỗi đối tượng ***x*** trong ***X*** cũng được gọi l| một *mẫu* (instance).

*Giả thuyết học quy nạp*

Học quy nạp chỉ {p dụng được cho c{c b|i to{n thỏa mãn giả thiết rằng: bất cứ giả thuyết n|o xấp xỉ tốt h|m đích trên tập mẫu đủ lớn thì cũng xấp xỉ tốt h|m đích trên c{c mẫu chưa biết.

Các bài to{n học có tập giả thuyết thoả mãn giả thiết trên gọi l| b|i to{n thỏa mãn ***giả thuyết học quy nạp***.

*Giả thuyết phù hợp và không gian tường thuật*

***Định nghĩa 2.1***. (*Giả thuyết phù hợp*) Một giả thuyết *h* gọi l| ***phù hợp*** với tập dữ liệu đ|o tạo ***D*** nếu v| chỉ nếu *h(x )* = *c(x)* với mỗi <*x,c(x)*> trong ***D***. consistent(*h,****D***) ( *x*,*c*(*x*) *D*) *h*(*x*)*c*(*x*).



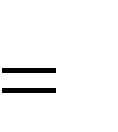
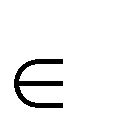
Tập tất cả c{c giả thuyết phù hợp n|y gọi l| không gian tường thuật đối với không gian giả thuyết ***H*** v| tập mẫu đ|o tạo ***D***.

***Định nghĩa 2.2***.(*không gian tường thuật*) Không gian tường thuật (được ký hiệu l| *VSH,D*) đối với không gian giả thuyết ***H*** v| tập dữ liệu huấn luyện ***D*** l| tập con của ***H*** phù hợp với tập mẫu ***D***:

### *VSH*,*D H* consistent(*h*,*D*). (2.4)



*h*



Khi c{c tập ***X*** và ***Y*** đều l| hữu hạn thì *VSHD* có thể x{c định nhờ tìm kiếm trên

to|n không gian giả thuyết. Tuy nhiên, như đã nói ở trên, khi ***X*** và ***Y*** lớn thì việc tìm

kiếm như vậy không khả thi. Để minh họa xét b|i to{n tìm kiếm trong học kh{i niệm.

#### 2.3. HỌC KHÁI NIỆM VÀ BÀI TOÁN TÌM KIẾM

2.3 CONCEPT LEARNING AS SEARCH (page 23. Tom Mitchell)

Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation. The goal of this search is to find the hypothesis that best fits the training examples. It is important to note that by selecting a hypothesis representation, the designer of the learning algorithm implicitly defines the space of all hypotheses that the program can ever represent and therefore can ever learn. Consider, for example, the instances X and hypotheses H in the EnjoySport learning task.

Example Sky AirTemp Humidity Wind Water Forecast EnjoySport 1 Sunny Warm Normal Strong Warm Same Yes 2 Sunny Warm High Strong Warm Same Yes 3 Rainy Cold High Strong Warm Change No 4 Sunny Warm High Strong Cool Change Yes TABLE 2.1 Positive and negative training examples for the target concept EnjoySport

Xét bài toán học kh{i niệm với thuộc tính nhận gi{ trị trên tập hữu hạn “*thời tiết của ngày bạn A thích chơi môn thể thao dưới nước”* nêu ở ví dụ 1 mục trước với tập mẫu ***X*** cho trong bảng 2.2.

Dễ d|ng thấy có 3.2.2.2.2.2 = 96 mẫu kh{c biệt trong ***X*** và có {nh xạ kh{c nhau từ ***X*** v|o tập Như vậy, với mỗi tập dữ liệu huấn luyện ***D*** đã cho, số lượng giả thuyết có thể xem xét l| qu{ nhiều nên việc tìm kiếm vét cạn không dễ.

Ta sẽ xét một thuật to{n giải b|i to{n n|y theo tiếp cận hạn chế không gian giả thuyết.

##### **2.3.1. Thuật toán tìm kiếm giả thuyết chi tiết nhất**

Trước khi trình b|y thuật to{n, ta quy ước c{c thuộc tính có thể nhận những gi{ trị sau:

* Ký hiệu bởi ? nếu bất cứ gi{ trị n|o của thuộc tính cũng được chấp nhận
* Ký hiệu bởi  nếu không gi{ trị n|o được chấp nhận  Một gi{ trị cụ thể của thuộc tính.

Chẳng hạn, giả thuyết A chỉ chơi khi trời lạnh v| độ ẩm cao được biểu diễn bởi:

<?, lạnh, cao,?,?,?>.

Khi đó giả thuyết tổng qu{t nhất : mọi ng|y đều chơi l|

<?, ?, ?,?,?,?>,

còn giả thuyết chi tiết nhất: không ng|y n|o chơi l|

<>.



,



,



,



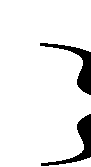
,



,



B}y giờ ta sẽ tìm lời giải cho b|i to{n n|y trong không gian giả thuyết H bao gồm tất cả c{c giả thuyết có dạng một liên kết c{c gi{ trị thuộc tính, chúng có thể l| ?;  hoặc gi{ trị cụ thể, gi{ trị kh{i niệm đích *c* (ưa thể thao) thuộc tập 0,1 .Dễ d|ng



tính được có 5.4.4.4.4.4 = 5120 c{c giả thuyết kh{c nhau trong H. Tuy nhiên, c{c giả thuyết có ký hiệu đều như nhau nên chỉ còn lại 1+ 4.3.3.3.3.3 = 973 giả thuyết kh{c nhau về ngữ nghĩa. Với số lượng giả thuyết như vậy, ta có thể tìm được thuật to{n tìm kiếm có hiệu quả không gian tường thuật *VSH,D* nếu nó kh{c rỗng trong không gian giả thuyết n|y. Để giới thiệu thuật to{n, ta cần đến cấu trúc thứ tự của tập c{c giả thuyết H vừa nêu ở trên.

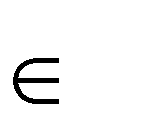
***Cấu trúc thứ tự “tổng quát đến chi tiết” của các giả thuyết***.

Nhiều thuật to{n học kh{i niệm được giải bằng c{ch tìm kiếm trong không gian giả thuyết nhờ dùng cấu trúc thứ tự *tổng quát đến chi tiết của không gian giả thuyết.* Nhờ cấu trúc n|y, ta có thể tìm kiếm vét cạn trong không gian giả thuyết (thậm chí có thể vô hạn) m| không cần đ{nh số giả thuyết. Để minh hoạ thứ tự n|y ta xét hai giả thuyết:

*h1*= <nắng,?,?, mạnh, ? , ? > *h2*= <nắng,?,?, ?, ? , ? >

v| xét tập mẫu được *h1, h2* ph}n lớp dương. Bởi vì *h2* ít r|ng buộc hơn nên nó có nhiều mẫu dương hơn *h1*. Khi đó ta nói *h2* tổng qu{t hơn *h1*.

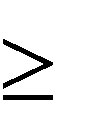
Trước khi định nghĩa chính x{c quan hệ thứ tự tổng qu{t ta cần kh{i niệm mẫu ***x*** thoả mãn giả thuyết *h*.

***Định nghĩa 2.3*:** (*đối tượng thỏa mãn giả thuyết*) Với mọi đối tượng ***x******X*** v| giả thuyết *h* ***H*** , ta nói ***x*** ***thoả mãn*** *h* nếu *h*(***x***) = 1.

Chú ý rằng kh{i niệm *thỏa mãn* n|y kh{c với kh{i niệm phù hợp trong định nghĩa 2.1, ở đ}y chỉ xét với mẫu dương. B}y giờ ta định nghĩa thứ tự tổng qu{t (chi tiết) của c{c giả thuyết.

***Định nghĩa 2.4***. (*Quan hệ tổng quát-chi tiết*) Giả sử *hj* và *hk* l| hai h|m gi{ trị boolean x{c định trên ***X***. Ta nói *hj* tổng qu{t hơn *hk* ( v| viết l| *h j  hk* ) nếu v| chỉ nếu bất cứ mẫu n|o thỏa mãn *hk* thì cũng thỏa mãn *hj*:

*g*



(***X****) [hk(x)=1* *hj(x)=1].* (2.5)

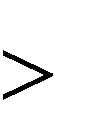


***x***



Ta cũng nói *hj*  tổng qu{t chặt hơn *hk* ( v| viết l| *h j  hk* ) nếu v| chỉ nếu:

*g*

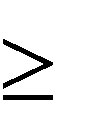


(*h j h j* ). (2.6)

*k*

*g*

*h*



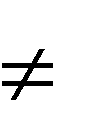
)



(

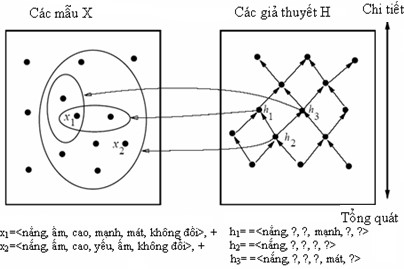
*k*

*h*



Trong c{c trường hợp trên ta cũng nói *hk* chi tiết (chặt) hơn *hj*.

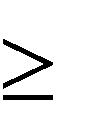
Để minh hoạ định nghĩa n|y ta xét ba giả thuyết *h1; h2; h3* trong hình 2.4 ở b|i to{n của ví dụ đang xét. Trong hình n|y, bên tr{i biễu thị tập mẫu ***X***, bên phải biểu thị tập H tất cả c{c giả thuyết, mỗi giả thuyết ứng với một tập con c{c mẫu m| nó ph}n lớp dương. Trong hình, *h2*tổng qu{t hơn *h1* và *h3,* còn *h1* và *h3* không so sánh được với nhau.



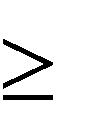
**Hình 2.4**. Cấu trúc thứ tự của các giả thuyết: bên trái là tập mẫu X, bên phải là tập giả thuyết H, mũi tên chỉ quan hệ “tổng quát hơn”

Chú ý rằng quan hệ và >g l| độc lập với kh{i niệm đích ; chúng chỉ phụ thuộc v|o c{c mẫu thoả mãn hai giả thuyết m| không phụ thuộc v|o sự ph}n lớp c{c mẫu phù hợp với kh{i niệm đích. Quan hệ rất quan trọng vì nó cho một cấu trúc có ích trên không gian giả thuyết H đối với mọi b|i to{n học kh{i niệm. Thuật to{n sau cho phép tìm giả thuyết chi tiết nhất sao cho mọi đối tượng dương trong tập mẫu đều thỏa mãn nó.

*g*



*g*



##### **Thuật toán Find-S (tìm một giả thuyết chi tiết nhất)**

Giả sử *h* l| một giả thuyết phù hợp với tập mẫu *D* v| không gian giả thuyết nêu trên chứa giả thiết phù hợp với ***D*** thì dễ d|ng kiểm tra được giả thiết chi tiết nhất m| mọi mẫu dương đều thỏa mãn sẽ l| giả thuyết phù hợp. Để tìm một giả thuyết n|y, ta bắt đầu từ giả thuyết chi tiết nhất *h* = v| nhờ cấu trúc thứ tự đã biết, tổng qu{t ho{ giả thuyết n|y mỗi khi có một mẫu dương không thoả mãn nó. Thuật to{n Find-S được trình b|y trong bảng 2.3.



,

,

,

,

,

Chẳng hạn nếu đưa ví dụ 1 trong bảng 2.2:

***x***1= < nắng, ấm, trung bình, mạnh, ấm, không đổi> thì h < nắng, ấm, trung bình, mạnh, ấm, không đổi>.

Nếu đưa thêm ví dụ 2: ***x***2 = < nắng, ấm, cao, mạnh, ấm, không đổi> thì h< nắng, ấm, ?, mạnh, ấm, không đổi>.

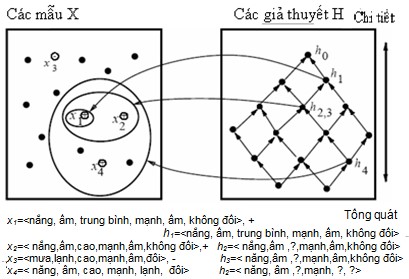
**Bảng 2.3**: Mô tả thuật toán Find-S

|  |
| --- |
| 1. Khởi tạo *h* l| giả thuyết chi tiết nhất; // *h*   ,  ,  ,  ,  ,   1. Với mỗi mẫu đ|o tạo dương ***x*** thực hiện:   Với mỗi thuộc tính *Ai* trong *h,* nếu r|ng buộc *ai*thoả mãn thì giữ nguyên, ngược lại thay *ai* trong *h* bởi r|ng buộc tổng qu{t hơn thoả mãn x. |

Thuật to{n n|y bỏ qua c{c thí dụ sai (thứ 3) v| tới ví dụ thứ tư:

***x***4 = < nắng, ấm, cao, mạnh, lạnh, đổi> thì *h*< nắng, ấm, ?, mạnh, ?, ?>.

Thuật to{n {p dụng cho ví dụ được minh hoạ trong hình 2.5.



**Hình 2.5**. Thuật toán Find-S cho ví dụ ở bảng 2.2

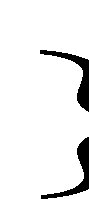
Thuật to{n n|y hạn chế không gian giả thuyết trên tập c{c *giả thuyết cho bởi liên kết các ràng buộc* *của thuộc tính*. Khi đó ta tìm được giả thuyết chi tiết nhất phù hợp với c{c mẫu dương v| nó cũng sẽ đúng với mẫu }m khi giả thuyết đúng v| tập mẫu cho đúng. Tuy vậy còn một số c}u hỏi chưa trả lời đối với thuật to{n n|y l|:

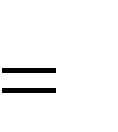
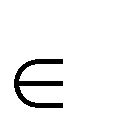
* Thuật to{n có đảm bảo hội tụ tới giả thuyết phù hợp với ***D*** không? Nếu có thì ngo|i giả thiết tìm được, còn có bao giả thuyết phù hợp v| liệu ta có thể tìm ra giả thuyết đúng hay không? Nếu không thì ta cũng cần ước lượng được tính không chắc chắn của nó.
* Tại sao ta ưa giả thuyết chi tiết nhất? c{c giả thuyết kh{c thì sao? (tổng qu{t nhất hoặc trung gian)
* Nếu tập mẫu ***D*** có ví dụ sai thì sao?

Sau đ}y | một tiếp cận kh{c, trong đó ta tìm *VSHD*m| không hạn chế không gian giả thuyết.

###### **2.3.2. Các thuật toán loại trừ ứng cử**

Trong mục n|y đưa ra thuật to{n vét cạn để tìm tất cả c{c giả thuyết phù hợp với tập mẫu đ|o tạo, tức l| to|n bộ không gian tường thuật:

 *VSH*,*D h H* consistent(*h*,*D*).



Sau đó giới thiệu một biến thể của nó dựa trên biễu diễn compact của không gian tường thuật.

*2.3.2.1****.*** *Thuật toán liệt kê loại trừ ứng cử (List-then-eliminate algorithm)*

Một c{ch đơn giản để biểu diễn không gian tường thuật l| liệt kê mọi phần tử của nó. Ý tưởng n|y dẫn tới thuật to{n liệt kê-loại trừ ứng cử được mô tả trong bảng 2.4. Thuật to{n n|y khởi tạo to|n bộ không gian giả thuyết rồi sau đó loại trừ dần c{c giả thuyết không phù hợp khi đối s{nh tuần tự với c{c mẫu quan s{t được.

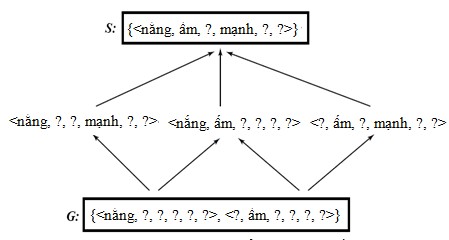
**Bảng 2.4.** Thuật toán liệt kê loại trừ ứng cử

|  |
| --- |
| 1. *VSH,D*  danh s{ch c{c giả thuyết trong H 2. Với mỗi <x,c(x)> trong D  loại trừ c{c giả thuyết h*VSH,D* mà h(x)c(x)  3. Đầu ra *VSH,D* |

Về nguyên tắc thì thuật to{n n|y có thể {p dụng khi không gian giả thuyết ***H*** hữu hạn nhưng trong thực h|nh thì thường không thể liệt kê v| tìm kiếm vét cạn ***H***. Một biến thể của thuật to{n n|y được thực hiên dựa trên biểu diễn sau của không gian tường thuật

*2.3.2.2.Một cách biểu diễn compact đối với không gian tường thuật*

Người ta có thể biểu diễn không gian tường thuật nhờ thứ tự tổng qu{t đến chi tiết bằng c{ch chỉ ra cận dưới chi tiết nhất v| cận trên tổng qu{t nhất của *VSH,D* mà không phải liệt kê chúng. Để minh hoạ cho c{ch biểu diễn n|y, ta trở lại với b|i to{n học kh{i niệm thích chơi thể thao với c{c ví dụ trong bảng 2.2. Thuật to{n Find-S cho ta giả thuyết chi tiết nhất l|: < nắng, ấm, ?, mạnh, ?, ?> thực ra có s{u giả thuyết trong H phù hợp với tập mẫu trong bảng 2.2, c{c giả thuyết n|y được mô tả trong hình 2.6.

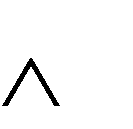
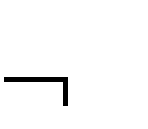
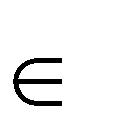
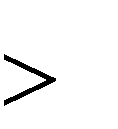
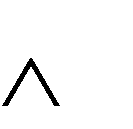


**Hình 2.6.** VSHD với biên tổng quát và chi tiết của khái niệm học từ bảng 2.2

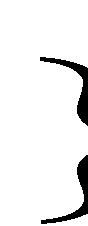
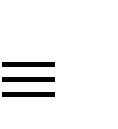
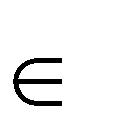
Ý tưởng n|y dẫn đến thuật to{n loại trừ ứng cử. Để trình b|y thuật to{n ta cần đến định nghĩa c{c tập biên tổng qu{t ***G*** v| chi tiết ***S*** đối với không gian giả thuyết ***H*** và tập dữ liệu ***D***.

***Định nghĩa 2.5***. (*Biên tổng quát và biên chi tiết)*

1. *Biên tổng quát* ***G*** đối với không gian giả thuyết ***H*** v| tập dữ liệu đ|o tạo ***D*** l| tập c{c giả thuyết tổng qu{t nhất của ***H*** phù hợp với ***D***:
   1. *H consistent*(*g*, *D*) ( *g*' *H*) (*g*' *g g*) *consistent*(*g*', *D*)



*g*

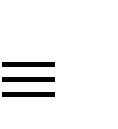
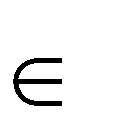


1. *Biên chi tiết* ***S*** đối với không gian giả thuyết ***H*** v| tập dữ liệu đ|o tạo ***D*** l| tập c{c giả thuyết chi tiết nhất của ***H*** phù hợp với ***D***:

***S*** *H consistent*(*s*,*D*) *consistent*(*s*',*D*)



*s*



)

'

(

)

'

(

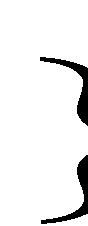
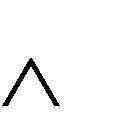
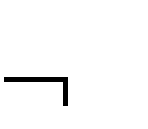
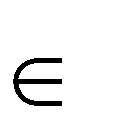
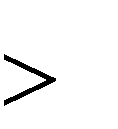
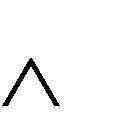
*s*

*s*

*H*

*s*

*g*



Tập ***G*** và ***S*** n|y ho|n to|n có thể đặc tả *VH,D* nhờ định lý sau.

**Định lý 2.1.** *(Biễu diễn không gian tường thuật)*. Giả sử ***X*** l| tập mẫu tuỳ ý v| ***H*** l| tập giả thuyết gi{ trị boolean x{c định trên ***X***. Cho ***c***: ***X***,0,1} l| một kh{i niệm đích x{c định trên ***X*** và ***D*** l| tập mẫu quan s{t được tuỳ ý của nó : ***D***= {***x***, c(***x***)}, khi đó với mọi mọi ***X***, ***H***, ***c*** và ***D*** sao cho ***G*** và ***S*** được x{c định thì

*VH,D* = *h*  (2.7)



)

)(

)(

(

*s*

*h*

*g*

*G*

*g*

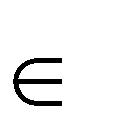
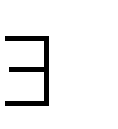
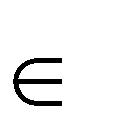
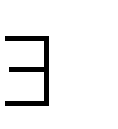
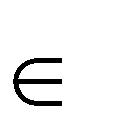
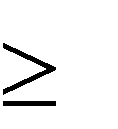
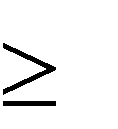
*S*

*s*

*H*

*g*

*g*



.

*2.3.2.3.**Thuật toán loại trừ ứng cử*

Trong thuật to{n n|y ta khởi tạo ***G*** và ***S*** bởi giả thuyết ***G0*** và ***S0*** l| c{c giả thuyết tổng qu{t v| chi tiết nhất:

* 1. ***G0*** =<?,?,?,?,?,?>

***S S0*** = < , , , , , >.



Khi mỗi mẫu đ|o tạo được xem xét, ***G*** và ***S*** được chi tiết ho{ v| tổng qu{t ho{ để loại trừ khỏi *VSH,D* c{c giả thuyết không phù hợp với dữ liệu tương ứng. Sau khi tập mẫu ***D*** đã được xét hết thì ***VSH,D*** chỉ chứa c{c giả thuyết phù hợp với ***D*** trong ***H***. Thuật to{n được mô tả trong bảng 2.5.

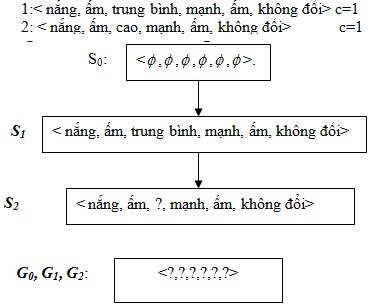
###### ***Một ví dụ minh hoạ***

Hình 2.7. biểu diễn thuật to{n khi {p dụng hai mẫu đầu trong bảng 2.2. Tập biên khởi tạo bởi G0 và S0 . Khi xem xét mẫu thứ nhất (ở đ}y l| mẫu dương), thuật to{n kiểm tra biên ***S*** v| thấy nó không phù hợp với mẫu n|y v| đưa v|o ***S1*** như trong hình, còn ***G1***=***G0*** vì ***G0***vẫn phù hợp với mẫu n|y. Khi xét mẫu 2, cũng l| mẫu dương, tương tự trước ta tổng qu{t hóa ***S1***thành ***S2*** còn ***G2***= G0 không đổi. Ta thấy xử lý hai mẫu dương n|y tương tự như trong thuật to{n Find S.

**Bảng 2.5.** Thuật toán loại trừ ứng cử

|  |
| --- |
| Bước 1. Khởi tạo G l| tập giả thuyết tổng qu{t nhất trong H,  Khởi tạo S l| tập giả thuyết chi tiết nhất trong H; Bước 2.( Lặp)Với mỗi ví dụ đ|o tạo *d*={***x***,c(***x***)}, thực hiện:  2.1. Nếu ***d*** l| ví dụ dương tính (*c*(*x*)=1):  2.1.2. Lấy khỏi ***G*** c{c giả thuyết không phù hợp với *d*;  2.1.2. Với mỗi *s* trong ***S*** không phù hợp với *d*:  Lấy *s* khỏi ***S***;  Thêm vào ***S*** c{c tổng qu{t ho{ chi tiết nhất ***h*** của *s* mà *h* phù hợp với *d* v| có một phần tử trong ***G*** tổng qu{t hơn ***h***;  Lấy khỏi ***S*** c{c giả thuyết tổng qu{t hơn c{c giả thuyết kh{c trong ***S***  2.2. Nếu *d* l| ví dụ }m tính (*c*(*x*)=0)  2.2.1. Lấy khỏi ***S*** c{c giả thuyết không phù hợp với *d*  2.2.2. Với mỗi *g* trong ***G*** không phù hợp với *d*:  Lấy *g* khỏi ***G;***  Thêm vào ***G*** c{c chi tiết ho{ nhỏ nhất *h* của *g* mà *h* phù hợp với *d* và có một phần tử trong ***S*** chi tiết hơn *h*;  Lấy khỏi ***G*** c{c giả thuyết ít tổng qu{t hơn c{c giả thuyết trong ***G*** |

*Mẫu huấn luyện*:

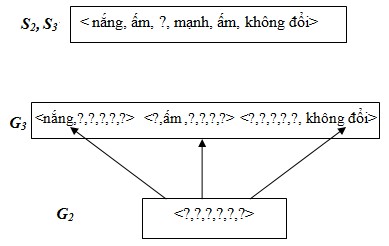


**Hình 2.7.** Kết quả xử lý các ví dụ 1 và 2

Trong hai bước n|y, c{c mẫu dương có thể l|m cho biên chi tiết của không gian tường thuật tăng tính tổng qu{t hơn. Còn ví dụ }m sẽ tăng tính chi tiết của biên tổng qu{t như khi xét mẫu tiếp theo.

B}y giờ ta xét mẫu 3: < mưa, lạnh, cao, mạnh, ấm, đổi> *c* = *0* như minh hoạ trong hình 2.8. Trong trường hợp n|y ***S3*** vẫn như ***S2*** còn ***G3***chứa ba giả thuyết chi tiết hóa nhỏ nhất của ***G2***.

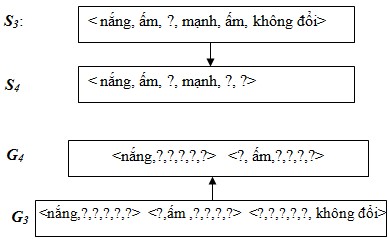
*Mẫu*: < mưa, lạnh, cao, mạnh, ấm, đổi> *c* = *0*



**Hình 2.8.** Kết quả xử lý mẫu 3

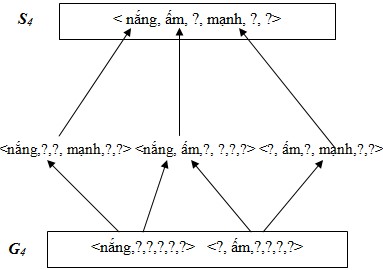
Tiếp theo, ta xét mẫu thứ tư: <nắng, ấm, cao, mạnh, lạnh, đổi> c =1, kết quả được minh họa trong hình 2.9. Trong biên tổng qu{t ***G3*** , mẫu n|y không thỏa mãn giả thuyết <?,?,?,?,?, không đổi> nên bị loại v| ***G4***chỉ còn hai giả thuyết. Giả thuyết trong ***S3*** không phù hợp với mẫu mới nên được thay bằng giả thuyết tổng qu{t hơn nhỏ nhất v| chi tiết hơn giả thuyết trong ***G4***, ta có ***S4***.

*Mẫu*: < mưa, ấm, cao, mạnh, ấm, đổi> *c* = *1*



**Hình 2.9.** Kết quả xử lý mẫu 4

Cuối cùng, biễu diễn compact của ***VH,D***được mô tả trong hình 2.10.



**Hình 2.10.** Biểu diễn compact của *VSH,D*

**Nhận xét.** Trong c{c thuật to{n trên, ta ngầm hiểu c{c gi{ trị thuộc tính được lấy trên tập hữu hạn v| c{c mẫu quan s{t được không có nhiễu (đều l| mẫu đúng). Khi c{c thuộc tính nhận gi{ trị thực, ta có thể rời rạc hóa miền gi{ trị của mỗi thuộc tính bởi các khoảng. Tuy nhiên với b|i to{n nội suy thì việc giải đúng hệ phương trình 2.2 thường rất khó thực hiện m| thường phải giải gần đúng như b|i to{n hồi quy. B}y giờ ta chuyển sang việc giải c{c b|i to{n n|y

#### 2.4. HỌC HÀM NỘI SUY VÀ HỒI QUY

##### **2.4.1. Phƣơng pháp trực tiếp tìm hàm nội suy**

Xét bài toán nội suy đã nêu trong mục 2.1.3: dựa trên tập quan s{t được gi{ trị quan s{t của h|m chưa biết *f* tại . Cần tìm h|m *g* có dạng đã chọn thỏa mãn hệ

trong

, trong đó l|

điểm

phương trình ( 2.2):

để nội suy h|m *f* tại c{c điểm x mới.

Việc xây dựng hàm *g* thường tu}n theo lược đồ tổng qu{t như sau: chọn trước một hàm dạng tổng quát phụ thuộc *N* tham số v| h|m nội suy g có dạng:

*g*(***x***) = , (2.8a) trong đó c{c tham số được x{c định dựa trên hệ phương trình (2.2).

Khi chọn trước hệ N h|m độc lập tuyến tính và hàm l| tổ hợp tuyến tính của chúng:

(2.8b)

thì h|m nội suy *g* có thể x{c định nhờ giải hệ phương trình tuyến tính để tìm c{c hệ số :

*j = 1,…,N.* (2.8c)

Với n = 1 v| , ta tìm được h|m nội suy Lagrange quen biết. Khi *N*

lớn*,* việc giải hệ (2.8c) có thể có sai số lớn do tích lũy sai số tính to{n, ta có thể tìm gần đúng c{c hệ số nhờ cực tiểu tổng bình phương sai số như trong mục sau.

##### **2.4.2. Tìm hàm hồi quy**

Phương ph{p thông dụng giải b|i to{n hồi quy l| tìm h|m hồi quy *g* có biểu diễn tham số tương tự như trong biểu thức (2.8a):

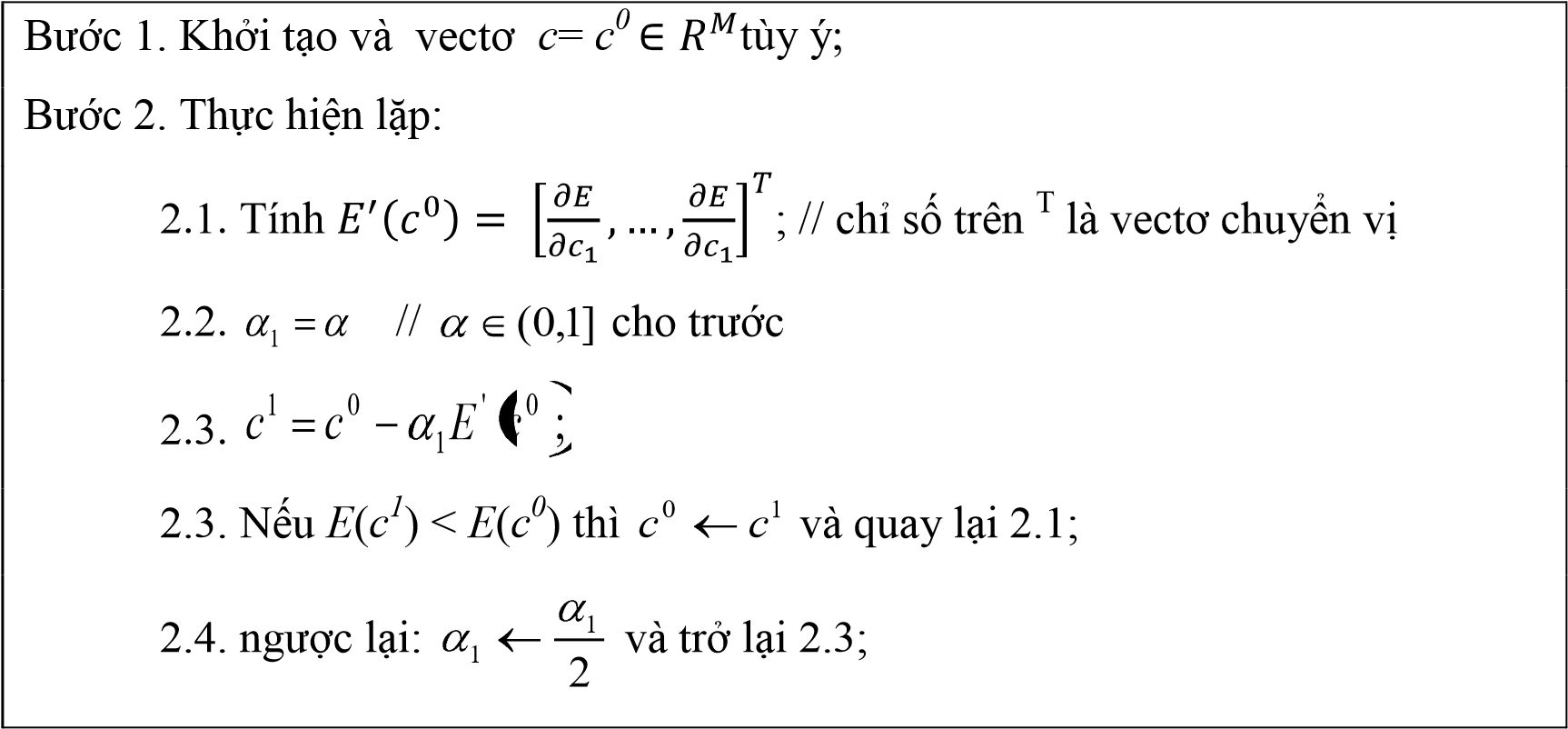
*g*(***x***) = , (2.9a) trong đó *M* có thể nhỏ hơn *N*.

Việc x{c định c{c hệ số được thực hiện nhờ tìm cực tiểu tổng bình phương sai số (sum of squared errors viết tắt l| SSE) *E*:

. (2.9b)

Với b|i to{n nội suy thì cực tiểu của *E* tìm được sẽ bằng không. Trường hợp tổng qu{t, vectơ ***c*** có thể tìm bằng thuật to{n gradient**[[1]](#footnote-1)** (hoặc một biến thể của nó). Một biến thể đơn giản của thuật to{n gradient được đặc tả trong bảng 2.6.

**Bảng 2.5.** Thuật toán gradient tìm cực tiểu hàm E



Thuật to{n dừng khi *E* ' *c*1 đủ bé, trong đó là chuẩn Euclide của vectơ u trong *RM*.

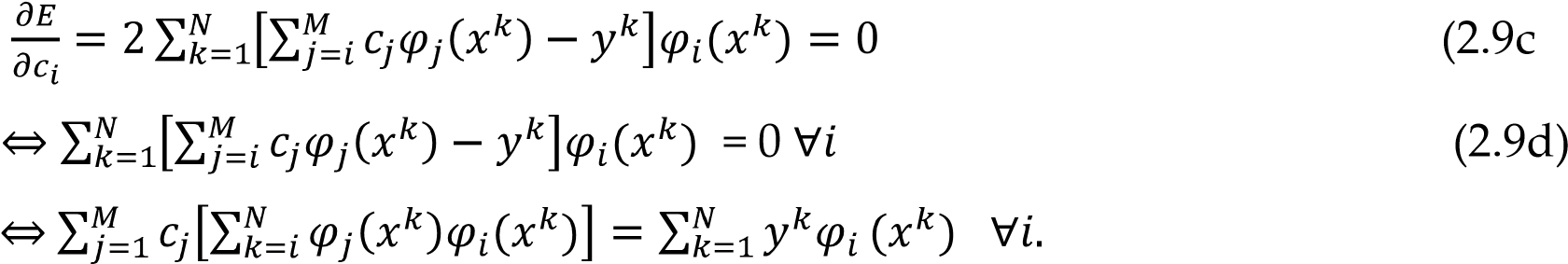


*Trường hợp tuyến tính*

Khi có dạng tuyến tính của c{c h|m cơ sở :

(2.9b)

thì ta dễ d|ng x{c định được hệ phương trình để tìm điểm dừng của E: cho bởi =0 với mọi *i=1,…,M*

 (2.9e)

Biểu thức (2.9c) cũng cho ta x{c định được vectơ gradient *E’(c)*. Nếu M không lớn, ta có thể giải hệ phương trình tuyến tính (2.9e) để x{c định h|m hồi quy.

#### 2.5. MỘT SỐ VẤN ĐỀ LIÊN QUAN

##### **2.5.1. Khuynh hƣớng quy nạp**

Như đã biết trong mục 2.3, nếu không gian giả thuyết H l| tất cả c{c h|m có thể có từ ***X*** lên ***Y*** thì nói chung có nhiều lời giải phù hợp với tập dữ liệu huấn luyện ***D*** đã cho. Như vậy, b|i to{n học thiết lập không đúng đắn. Hơn nữa việc tìm kiếm trên không gian n|y thường không khả thi trong thực tế. Vì vậy người ta thường đưa ra c{c giả thuyết cho tập dữ liệu để thuật to{n học duy nhất nghiệm v| hạn chế không gian tìm kiếm. Tập c{c giả thiết n|y gọi l| khuynh hướng quy nạp(bias) của thuật to{n học.

Chẳng hạn, trong thuật to{n Find-S ta giả thiết không gian H chỉ chứa c{c liên kết r|ng buộc v| c{c ví dụ }m luôn đúng khi giả thuyết thỏa mãn c{c ví dụ dương. Đối với trường hợp cho trong bảng 2.6, giả thuyết n|y sẽ không thỏa mãn.

**Bảng 2.6.** Một tập mẫu mới về những ngày chơi hoặc không của A

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ví dụ | Bầu trời | nhiệt độ độ ẩm | gió | nước | dự b{o | thích chơi |
| 1  2  3 | nắng mây mưa | ấm trung bình ấm trung bình ấm trung bình | mạnh mạnh mạnh | mát mát mát | đổi đổi đổi | có  có  không |

Thực vậy, khi đó giả thuyết chi tiết nhất phù hợp với hai mẫu đầu của tập l|:

***S2*** = < ?, ấm, trung bình, mạnh, m{t, đổi>.

Tuy nhiên, giả thuyết n|y không phù hợp với mẫu thứ ba. Sở dĩ như vậy vì khuynh hướng học: H chỉ l| c{c liên kết r|ng buộc v| c{c ví dụ }m luôn đúng khi giả thuyết thỏa mãn c{c ví dụ dương không thỏa mãn dẫn đến không có lời giải.

##### **2.5.2. Học gần đúng theo xác suất**

Ta trở lại với b|i to{n học kh{i niệm theo c{ch đ{nh gi{ khả năng học gần đúng theo x{c suất (ký hiệu l| PAC: Probably Approximately Correct) của bộ ph}n lớp đối với một lớp kh{i niệm. Cho tập đối tựơng ***X*** (chẳng hạn, tập người) được mô tả bởi c{c đặc trưng (chẳng hạn, c{c chỉ số xét nghiệm sinh hóa, v| số đo vật lý) v| một lớp kh{i niệm cần học **C** (chẳng hạn, c{c loại bệnh lý có thể gặp). Mỗi kh{i niệm c **C** ứng với một h|m gi{ trị boolean *c*: ***X***{*0,1*} sao cho nếu *x* l| đúng với kh{i niệm thì *c*(*x*) = *1*, ngược lại *c*(*x*) = 0 nếu x l| }m tính.

Giả sử để học khái niệm, các mẫu trong ***X*** được lấy ngẫu nhiên với phân bố xác suất n|o đó. C{c mẫu huấn luyện được tạo sinh nhờ lấy ngẫu nhiên c{c mẫu *x* cũng theo phân bố , v| quan s{t h|m đích *c*(*x*) để có tập dữ liệu quan s{t cho bộ học ***L***. Bộ học n|y sẽ tìm kh{i niệm đích trên không gian giả thuyết ***H*** n|o đó. Sau khi sử dụng tập dữ liệu quan s{t tìm được, bộ học sẽ cho một giả thuyết *h* trong không gian giả thuyết ***H*** để dùng l|m ước lượng hàm *c*. Ta sẽ đ{nh gi{ sự thành công của bộ học***L*** thông qua khả năng thực hiện đo{n nhận của *h* trên các mẫu mới được lấy ra một cách ngẫu nhiên trong ***X*** theo cùng phân bố này.

Ở đ}y ta sẽ chú ý tới khả năng thực hiện của c{c bộ học kh{c nhau khi sử dụng c{c không gian giả thuyết kh{c nhau v| với c{c kh{i niệm đích lấy ra từ c{c lớp **C** kh{c nhau. Bởi vì ta cần bộ học phải học được đủ tốt bất cứ một kh{i niệm đích n|o trong C mà không để ý đến phân bố các mẫu đ|o tạo, nên ta thường chú ý phân tích trong c{c trường hợp xấu nhất trên tập khái niệm đích có thể có trong **C** và mọi phân bố mẫu có thể gặp.

##### **Lỗi của một giả thuyết**

Ta hãy bắt đầu với kh{i niệm *lỗi đúng* (true error) của một giả thuyết *h* đối với khái niệm đích *c* và một phân bố mẫu .

***Định nghĩa 2*.6**. (*Lỗi đúng*) Lỗi đúng của một giả thuyết *h* đối với khái niệm đích *c* và

phân bố ) l| x{c suất *h* ph}n lớp sai một mẫu lấy ngẫu

, (2.10)

được kí hiệu l|

(

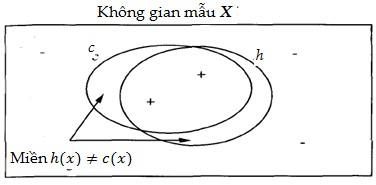
nhiên theo phâ

n b

ố

:

trong đó là kí hiệu xác suất thu được trên phân bố mẫu . Hình 2.11 ph{c họa miền lỗi, trên đó hai h|m *h* và *c* nhận gi{ trị kh{c nhau, x{c suất ở (2.11) l| độ đo hai miền n|y theo ph}n bố mẫu .



**Hình 2.11.** Xác suất lối của giả thuyết

Khi là phân bố đều v| X l| c{c điểm trên hình chữ nhật thì lỗi đúng l| tỷ lệ diện tích miền và diện tích toàn hình chữ nhật. Khi phân bố khác nhau thì nói chung x{c suất lỗi kh{c nhau. Tuy nhiên, ta không ước lượng trực tiếp được lỗi n|y m| phải ước lượng qua lỗi đ|o tạo (t*raining errorD(h*)) trên tập quan s{t được ***D***.

***Định nghĩa 2***.7***.*** (*Lỗi đào tạo*) Lỗi đ|o tạo trên tập ***D*** l| x{c suất sai của giả thuyết trên tập huấn luyện n|y:

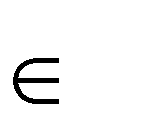
*Training errorD*(*h*) = *Prx Є D*(*c*(*x*)≠*h*(*x*)). (2.11)

Lỗi đ|o tạo dùng để ước lượng khoảng cho lỗi đúng, ta sẽ quay lại chủ đề n|y trong chương 5.

##### **Khả năng học gần đúng theo xác suất**

Theo giả thuyết quy nạp, số lượng mẫu quan s{t được c|ng nhiều thì lỗi đúng c|ng nhỏ v| ta luôn mong muốn tìm được giả thuyết *h* có lỗi đúng =0. Tuy nhiên điều n|y l| không thể trong thực tế, trừ khi tập mẫu quan s{t được l| to|n bộ tập ***X***.

Người ta khắc phục khó khăn n|y bằng cách chỉ đòi hỏi khi tập mẫu đ|o tạo đủ lớn thì với xác suất lớn hơn trong đó là hằng số đủ nhỏ tùy ý, bộ học tìm được giả thuyết *h* có lỗi đúng nhỏ hơn hằng số bé tùy ý cho trước. Khi đó ta nói bộ học ***L*** có khả năng học gần đúng x{c suất (PAC) lớp kh{i niệm *C.* Mô hình PAC được Valiant đề xuất (1984), Mitchell ph{t triển th|nh định nghĩa sau.

***Định nghĩa2.8.*** (*Học PAC*) Xét một lớp c{c kh{i niệm ***C*** trên một tập ***X*** c{c mẫu có độ dài *n* v| một bộ học ***L*** sử dụng không gian giả thuyết H. Ta nói ***C*** có thể học PAC bằng ***L*** nhờ dùng ***H*** nếu với mọi kh{i niệm *c*  ***C***, phân bố trong X và các hằng số ε, δ trong khoảng (0,1/2) thì với xác suất ít nhất là (1- δ), bộ học *L* đưa ra một được giả thuyết *h****H*** sao cho ≤ ε, với thời gian l| đa thức theo 1/ε, 1/δ, n v| size(*c*). (Ở đ}y size(c) l| độ d|i đã được mã ho{ của c trong C)

Trong định nghĩa trên, ta có hai yêu cầu với L. Thứ nhất: với x{c suất cao tùy ý

(lớn hơn 1- δ), ***L*** phải đưa ra đươc một giả thuyết *h* với một x{c suất lỗi bé tùy ý (nhiều nhất l| ε). Thứ hai, ***L*** phải l|m việc rất hiệu quả, theo thời gian nó chỉ tăng theo đa thức với c{c tỉ số 1/ε, 1/δ, *n* và *size*(*c*) về độ phức tạp.

###### **2.5.3. Chiều Vapnik-Chervonenkis**

Chiều *Vapnik-Chervonenkis*(VC) dùng để đo độ phức tạp của không gian giả thuyết ***H*** biểu thị qua khả năng ph}n biệt mẫu của nó. Xét tập mẫu ***S*** trong ***X****.* Mỗi giả thuyết *h* trong***H***sẽ t{ch ***S*** th|nh hai phần: tập mẫu dương tính v| tập mẫu }m tính. Tập ***S*** có tập con có thể l| tậpdương tính. Một b|i to{n đặt ra l| không gian giả thuyết ***H*** đã cho có ph}n t{ch (shatter) được một tập con bất kỳ của *S* như l| tập dương tính hay không? Trước khi định nghĩa chiều VC, ta cần định nghĩa sau về khả năng t{ch được của ***H*** đối với tập ***S***

***Định nghĩa 2.9***. (*Tính tách được*) Một tập mẫu ***S*** được gọi l| t{ch được bởi không gian giả thuyết ***H*** nếu v| chỉ nếu với mọi tập con của *S* luôn tồn tại một giả thuyết *h* trong ***H*** sao cho tập con n|y dương tính v| phần bù của nó trong S }m tính. Khi đó ta cũng nói ***H*** t{ch được ***S****.*

Không gian giả thuyết ***H*** được xem l| lý tưởng nếu nó t{ch được to|n bộ không gian mẫu ***X***. Khi ***H***  không thể t{ch ***X***, nhưng lại có thể ph}n t{ch một tập con ***S*** của ***X*** có cỡ (size) lớn thì sao? Bằng trực gi{c, có thể nói, tập con của ***X*** có khả năng t{ch được tập con c|ng lớn thì ***H*** c|ng có ý nghĩa. Chiều VC của ***H*** biểu thị độ đo n|y

***Định nghĩa 2.10***.(*Chiều Vapnik-Chervonenkis)* Chiều VC của không gian giả thuyết ***H*** (kí hiệu *VC(H*)) trên không gian mẫu ***X*** l| số cực đại c{c đối tượng trong ***X*** tách được bởi ***H***. Nếu tập con hữu hạn tuỳ ý của ***X*** có thể t{ch được bởi ***H*** thì *VC*(*H*)



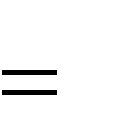
Dễ d|ng kiểm tra được với mọi không gian giả thuyết hữu hạn ta đều có ước lượng:

*VC*(*H*) ≤ . (2.12)

Chứng minh điều n|y d|nh cho độc giả như b|i tập

**Ví dụ 1.**

Giả sử không gian mẫu *X* l| tập c{c số thực *X* (ví dụ, mô tả chiều cao của người) v| *H* l| tập c{c khoảng số thực dạng : *a x b*với *a* và *b* l| c{c hằng số thực. Khi đó *VC(H)* là gì?

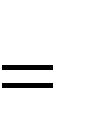
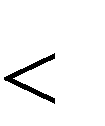
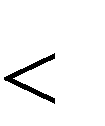
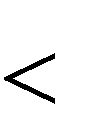
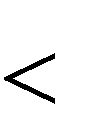
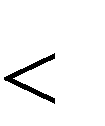
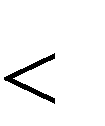
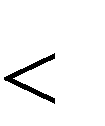
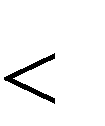


Để trả lời c}u hỏi n|y, chúng ta cần tìm tập lớn nhất của *X* có thể t{ch được bởi

*H*. Xét một tập con S gồm 2 mẫu kh{c nhau trong tập 3,1, 5,7 , ta thấy 4 giả thuyết

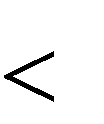
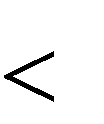


1 *x* 2 , 1 *x* 4 , 4 *x* 7 , 1 *x* 7 l|m được việc n|y ( tương ứng thể hiện 4 phân tách trên *S*: không có mẫu n|o, một trong 2 mẫu v| cả 2 mẫu). Bởi vì chúng ta đã tìm ra một tập có cỡ 2 t{ch được bởi *H*, do đó *VC(H)* tối thiểu l| 2. B}y giờ ta cần trả lời c}u hỏi: có tập n|o cỡ bằng 3 t{ch được bởi H không? Giả sử tập *S x*0 ,*x*1,*x*2 chứa 3 mẫu, không mất tính tổng qu{t ta cho *x*0 *x*2 . Rõ r|ng tập n|y không t{ch được bởi H vì mọi khoảng chứa tập con ,*x0*, *x2*} đều chứa *x1*. Vậy thì *VC(H)=2*. Chú ý rằng *H* vô hạn nhưng *VC(H)* l| hữu hạn.



1

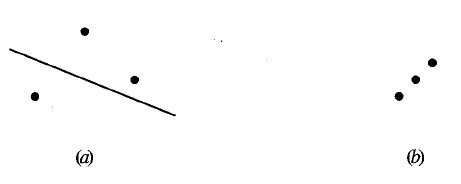
*x*



###### ***Ví dụ 2***

Tiếp theo, ta xét tập ***X*** có c{c mẫu tương ứng với c{c điểm trong mặt phẳng hai chiều *x,y* (xem hình 2.12)*.* ***H*** l| tập tất cả quyết định tuyến tích trong mặt phẳng (chia mặt phẳng bằng đừng thẳng). Ta có thể dễ d|ng thấy bất cứ 2 điểm n|o trong mặt phẳng đều có thể t{ch được bởi ***H***. Vậy *VC*(*H*) ≥2.

Với 3 điểm không thẳng h|ng, ta có thể tìm được 2 3 mặt tuyến tính t{ch chúng. Dĩ nhiên 3 điểm thẳng h|ng không thể ph}n t{ch được (lý do giống như 3 điểm trên trục số thực không thể ph}n t{ch được ở ví dụ trên). Vậy *VC(H)* tối thiểu l| 3. Định nghĩa về chiều *VC* chỉ ra rằng nếu chúng ta tìm ra bất kì tập mẫu n|o kích thước *d* mà ***H*** t{ch được thì *VC*(*H*)*d* . Để chứng minh *VC*(*H*)*d* , chúng ta phải chỉ ra không có tập n|o kích thước *d* có thể ph}n t{ch được. Trong ví dụ n|y, không có tập n|o kích thước 4 t{ch đuợc bởi H (b|i tập 9) thì *VC(H)=3 (hình 2.12).* Tổng qu{t hơn, kích thước *D* của c{c mặt tuyến tính trong một không gian *r* chiều l| *r+1*..



(a) Tập 3 điểm có thể được phân tách bởi quyết định tuyến tính.

(b)Tập 3 điểm không thể phân tách

**Hình 2.12.** Chiều VC của các đường thẳng trong mặt phẳng là 3

###### **KẾT LUẬN**

Học quy nạp từ tập mẫu quan s{t được có vị trí quan trọng trong học m{y. Trong đó, học kh{i niệm có thể thực hiện nhờ tìm kiếm trong một không gian giả thuyết lớn cho trước. Dựa trên cấu trúc thứ tự tổng qu{t đến chi tiết, thuật toán FindS thực hiện tìm kiếm giả thuyết chi tiết nhất phù hợp với tập mẫu đ|o tạo, trong khi thuật to{n loại trừ ứng cử x{c định không gian tường thuật nhờ c{c biên chi tiết v| tổng qu{t của nó.

B|i to{n hồi quy có thể thực hiện nhờ dùng thuật to{n gradient hoặc một biến thể của nó để tìm h|m hồi quy trong không gian giả thuyết biểu diễn bằng một h|m phụ thuộc tham số. Nói riêng khi h|m nội suy v| hồi quy l| tuyến tính đối với họ h|m cơ sở đã cho thì chúng có thể tìm bằng giải trực tiếp hệ phương trình tuyến tính đối với c{c hệ số.

B|i to{n học quy nạp thường không duy nhất nghiệm v| phải tìm kiếm trên không giả thuyết lớn, dùng khuynh hướng quy nạp l| một tiếp cận để khắc phục hiện tượng n|y.

Mô hình gần đúng x{c suất xét c{c thuật to{n học trong lớp kh{i niệm C với tập mẫu được lấy ngẫu nhiên với phấn bố x{c suất n|o đó. Mô hình đòi hỏi với mỗi kh{i niệm *c*, bộ học có khả năng tìm được giả thuyết gần đúng cho kh{i niệm có lỗi đúng nhỏ hơn ε bé tùy ý với x{c suất ít nhất l| (1- δ) bằng một thuật to{n thời gian đa thức theo 1/ε, 1/δ, n v| cỡ (size)của X, ***c***.

###### **BÀI TẬP**

1. Giải thích tại sao trong b|i to{n ở mục 2.2.3 lại có 96 mẫu v| 973 giả thuyết kh{c nhau, số mẫu v| giả thuyết sẽ l| bao nhiêu nếu thêm thuộc tính *dòng nước* với c{c gi{ trị *nhẹ*, *trung bình* và *xiết*
2. Chứng minh rằng lời giả của thuật to{n Find-S l| duy nhất
3. X}y dựng thứ tự tổng qu{t-chi tiết v| thuật to{n tương tự Find-S cho b|i to{n học kh{i niệm với thuộc tính liên tục
4. Cho biết thứ tự lấy ví dụ trong thuật to{n loại trừ ứng cử có ảnh hưởng tới kết quả không, tại sao?
5. Cho tập mẫu gồm c{c đối tượng có hai thuộc tính đặc trưng, đặc trưng thứ nhất nhận hai gi{ trị 0 hoặc 1, đặc trưng thứ hai nhận gi{ trị trong 5 chữ c{i. Dữ liệu quan s{t được gồm hai lớp cho bởi bảng sau:

|  |
| --- |
| : (0,A) (0,B) (1,B) (0,E) (0,D)  1 |
| : (1,A) (0,C) (1,C) (1,E) (1,D)  2 |

1. Tính số đối tượng kh{c nhau của b|i to{n
2. Áp dụng thuật to{n Find-S cho tập mẫu n|y v| tính số giả thuyết kh{c nhau trong không gian giả thuyết
3. Áp dụng thuật to{n loại trừ ứng cử cho tập mẫu n|y
4. Cho h|m hai biến *z =* *f*(*x,y*) với tập quan s{t cho bởi bảng:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | -2 | -1 | 0 | 1 | 2 | 1 | 0 | -1 |
| *y* | -1 | 0 | 2 | -1 | 1 | -2 | 1 | 1 |
| *z* | 2 | 5 | 8 | 3 | 21 | 4 | 2 | -4 |

* 1. Tìm hàm hồi quy bậc hai: cho hàm *z*
  2. Tìm h|m nội suy của *z* dạng:

1. Ph{t triển lược đồ {p dụng c{c thuật to{n Find-S v| loại trừ ứng cử cho b|i to{n học nhiều lớp

8-Chứng minh công thức (2.12)

9-Chứng minh rằng một tập gồm 4 điểm tùy ý trong mặt phẳng không thể t{ch được bởi không gian giả thuyết H trong ví dụ 2 ở mục 2.5.3

**Chƣơng 3**

## HỌC BẰNG CÂY QUYẾT ĐỊNH

Chương này giới thiệu phương pháp học có giám sát nhờ cây quyết định, bắt đầu từ bài toán phân lớp các đối tượng có thuộc tính nhận giá trị rời rạc, sau đó phát triển xử lý các trường hợp có thuộc tính liên tục và một số vấn đề liên quan.

### 3.1. BIỂU DIỄN GIẢ THUYẾT BẰNG CÂY QUYÊT ĐỊNH

Chương trước, giới thiệu tiếp cận giải các bài toán *phân lớp*/*hồi quy hàm* *số* nhờ tìm kiếm trong không gian giả thuyết. Học bằng cây quyết định để xây dựng trực tiếp giả thuyết cần tìm là một phương pháp thông dụng trong ứng dụng.

Để đơn giản, ta bắt đầu từ các bài toán mà các thuộc tính của đối tượng nhận giá trị trong tập hữu hạn. Bài toán được phát biểu như sau: Có hàm đích *c*: *X→ Y* trong đó ***x***=<*x1,…,xn*>, *xi* là giá trị trong miền giá trị của của thuộc tính *Ai* tương ứng, tập đích *Y* là các tập hữu hạn. Từ tập mẫu quan sát được ***D*** = *x* , *y* ta cần xác định một giả thuyết gần đúng *h* cho hàm *c* dưới dạng một tập luật để xác định hàm đích dựa trên giá trị thuộc tính dạng:



*N*

*k*

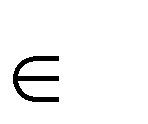
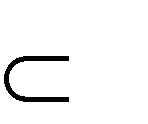
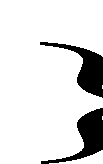
*k*

*k*

1



{ *Nếu x i vi* /*i I* 1,2,..., *n thì h(****x****)=yj* }*.* (3.1)



Các giả thuyết này có thể biễu diễn bằng một cây quyết định, trong đó mỗi đường đi xuôi từ gốc đến lá cho ta một luật có biểu diễn dạng (3.1) theo các ràng buộc giá trị-thuộc tính trên đường đi n|y.

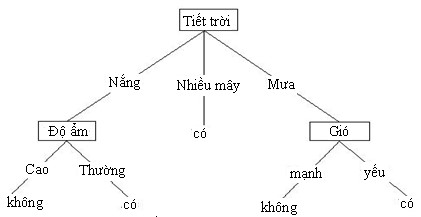
Nút gốc và mỗi nút trong trên cây sẽ định rõ một *thuộc tính/điều kiện* kiểm tra của mẫu và mỗi nh{nh đi xuống từ mỗi nút tương ứng với một trong những *giá trị/giá trị chân lý* có thể của *thuộc tính/điều kiện* này. Nhãn của các mẫu phù hợp với đường đi n|y được gắn ở nút lá. Với mỗi mẫu mới, nhãn của nó được x{c định bằng cách khởi đầu từ nút gốc của cây, kiểm tra *thuộc tính/điều kiện* ở nút n|y v| sau đó di chuyển xuống nh{nh tương ứng với *giá trị/giá trị chân lý* x{c định ở lá.

Ta xét tập mẫu quan sát về thời tiết thích hợp để bạn *A* chơi cầu lông dựa trên 4 thuộc tính { tiết trời, nhiệt độ, độ ẩm, gió} cho trong bảng 3.1.

**Bảng 3.1.**Tập mẫu về thời tiết phù hợp để chơi cầu lông

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ngày | Tiết trời | Nhiệt độ | Độ ẩm | Gió | Chơi cầu  lông |
| D1  D2  D3  D4  D5  D6  D7  D8  D9  D10  D11  D12  D13  D14 | Nắng  Nắng  Nhiều mây  Mưa  Mưa  Mưa  Nhiều mây  Nắng  Nắng  Mưa  Nắng  Nhiều mây  Nhiều mây  Mưa | Nóng  Nóng  Nóng  Mát lạnh lạnh lạnh Mát lạnh Mát  Mát  Mát  Nóng  Mát | Cao  Cao  Cao  Cao  Thường  Thường  Thường  Cao  Thường  Thường  Thường  Cao  Thường  Cao | Yếu  Mạnh  Yếu  Yếu  Yếu  Mạnh  Mạnh  Yếu  Yếu  Yếu  Mạnh  Mạnh  Yếu  Mạnh | Không  Không  Có  Có  Có  Không  Có  Không  Có  Có  Có  Có  Có  Không |

Hình 3.1 biễu thị một cây quyết định cho bài toán học khái niệm *thời tiết phù hợp để bạn A chơi cầu lông* dựa trên dữ liệu đ|o tạo cho trong bảng 3.1 v| được xây dựng nhờ thuật toán ID3 trong mục 3.2.



**Hình 3.1.** Cây quyết định học được từ mẫu cho trong bảng 3.1

Trong ví dụ này, các nút biểu thị thuộc tính kiểm tra, các nhánh từ mỗi nút ứng với giá trị kiểm tra của mẫu. C}y n|y tương ứng với tập gồm 5 luật:

<

{

*N*

*ế*

*u Ti*

*ế*

*t tr*

*ờ*

*i = n*

*ắ*

*ng*

*đ*

*ộ*

*ẩ*

*m = cao thì h*

(

*x*

)

=

*không*

>

;

<

*N*

*ế*

*u ti*

*ế*

*t tr*

*ờ*

*i = n*

*ắ*

*ng*

*đ*

*ộ*

*ẩ*

*m=*

*thư*

*ờ*

*ng thì h*

(

*x*

=

)

*có*

>

;

<

*N*

*ế*

*u ti*

*ế*

*t tr*

*ờ*

*i = nhi*

*ề*

*u mây thì h*

(

*x*

=

)

*có*

;

>

<

*N*

*ế*

*u ti*

*ế*

*t tr*

*ờ*

*i =mưa*

*gió = m*

*ạ*

*nh thì h*

(

*x*

=

)

*Không*

;

>

<

*N*

*ế*

*u ti*

*ế*

*t tr*

*ờ*

*i =mưa*

*gió= y*

*ế*

*u thì h*

(

*x*

=

)

*có*

>}

M

ộ

t cây h

ồ

i quy c

ủ

a m

ộ

t hàm ba bi

ế

n trong đó có hai bi

ế

n th

ự

c, bi

ế

n

nh

ậ

n

m

ộ

t trong t

ậ

p

đư

ợ

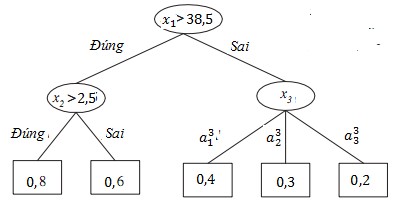
c bi

ể

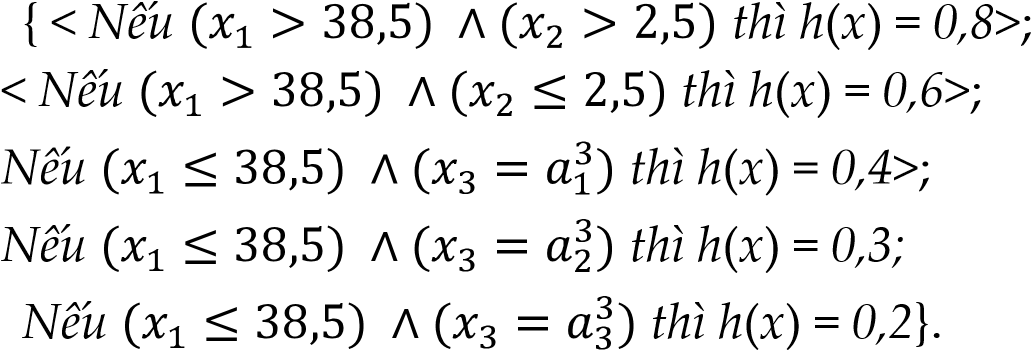
u th

ị

trong hình 3.2.



**Hình 3.2.** Một cây hồi quy của hàm 3 biến với 2 biến thựcTập luật tương ứng của cây này là:



Quá trình xây dựng cây quyết định dựa trên tập mẫu đ|o tạo sẽ được gọi là học cây quyết định. Mặc dù phương ph{p học bằng cây quyết có miền ứng dụng khá rộng, nhưng thích hợp nhất với các bài toán thuộc tính nhận giá trị rời rạc, dữ liệu quan s{t được có thể có lỗi hoặc mất/thiếu giá trị thuộc tính. Chẳng hạn, đối tượng dữ liệu thường có nhiễu hoặc bị thiếu giá trị của một vài thuộc tính trong các bài toán:

* Chẩn đo{n v| điều trị bệnh: dựa trên theo dõi triệu chứng lâm sàng và các kết quả xét nghiệm để chẩn đo{n bệnh v| đề xuất phương {n điều trị.
* Phân tích rủi ro ngân hàng: dựa v|o c{c đặc điểm kh{ch h|ng để đo{n nhận khách hàng có khả năng thanh to{n hay không, trên cơ sở đó quyết định cho vay hoặc không.

### 3.2. CÁC THUẬT TOÁN HỌC

#### 3.2.1. Thuật toán học ID3

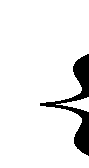
*3.2.1.1. Các khái niệm*

Thuật toán ID3 (Quinlan 1986) học cây quyết định cho bài toán học khái niệm bằng cách xây dựng bắt đầu từ gốc và phát triển dần đến các nút lá. Mỗi nút gốc hoặc nút trong biểu thị một thuộc tính kiểm tra, các cạnh biểu thị giá trị kiểm tra của thuộc tính tương ứng. Thuộc tính tốt nhất của tập dữ liệu đ|o tạo được chọn làm nút gốc theo tiêu chuẩn cực đại lượng *thu hoạch thông tin* (Information gain )*.* Để định nghĩa thu hoạch thông tin, ta cần đến khái niệm Entropy của một tập.

*Entropy*

Cho một tập **S**, gồm các mẫu nhận giá trị đích dạng boolean x{c định bởi hàm *c*, ta dùng các ký hiệu:

(3.2b)



;

0

)

(

:

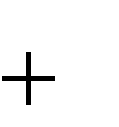
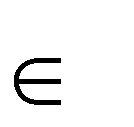
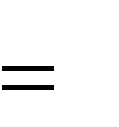
,

1

)

(

:



*s*

*c*

*S*

*s*

*S*

*s*

*c*

*S*

*s*

*S*

(3.2a)

*S*

*S*

*p*

*S*

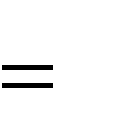
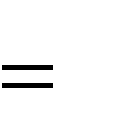
*S*

*p*

/

,

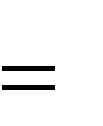
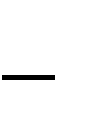
/



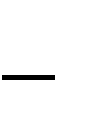
.

Khi đó Entropy của S l| đại lượng x{c định bởi:

*Ent ropy* ( *S* ) log 2 *p* log 2 *p*  (3.3a)



*p*



*p*

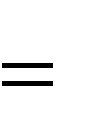
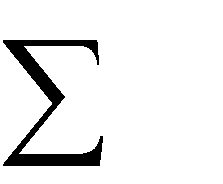
**Ví dụ .** Để minh hoạ, ta giả sử có một người có chơi cầu lông hay không dựa vào đặc điểm thời tiết: Tiết trời, Nhiệt độ, Độ ẩm, Gió. Tập mẫu quan s{t được ***S*** cho trong bảng 3.1 gồm 14 mẫu với 9 mẫu dương tính v| 5 mẫu âm tính (chúng ta dùng ký hiệu [9+,5-] ), thì entropy của ***S*** là:

*Entropy*([9+,5–]) = –(9/14)log2(9/14) – (5/14)log2(5/14) = 0.940

Lưu ý rằng *Entropy(S)* lớn nhấtkhi các mẫu dương tính v| }m tính của***S*** có số lượng bằng nhau và *Entropy(S) = 0* nếu tất cả các phần tử của S đều thuộc cùng một lớp.

Tổng qu{t hơn, trong trường hợp có *k* lớp và ký hiệu *pk* là tỷ lệ mẫu có nhãn *ck* trong ***S***  thì khi đó Entropy(S) ứng với tập này là:

Entropy(S) - pi log 2 *pi* . (3.3b)



k

1

i

*Thu hoạch thông tin*

Thu hoạch thông tin của một thuộc tính *A* ứng với một tập mẫu *S* được ký hiệu là *Gain*(*S,A*), đo sự giảm kỳ vọng entropy theo thuộc tính n|y, được x{c định như sau:

*Gain*(*S,A*)  *Entropy*(***S***) -  *Entropy(Sv*), (3.4)



)

(

*A*

*Values*

*v*

*S*

*S*

*v*

trong đó, *Values*(*A*) là một tập các giá trị mà thuộc tính *A* có thể nhận, và *Sv* là tập hợp con của *S* nhận giá trị *v* ở thuộc tính này: *Sv* = {s *A(s)=v}*). Số hạng đầu trong vế phải của (3.4) là entropy của toàn bộ tập ***S*** còn số hạng thứ hai là giá trị kỳ vọng của entropy sau khi tách ***S*** ra bằng thuộc tính *A*. Chính vì vậy *Gain*(*S,A*) là chiết giảm entropy mong đợi dựa trên các giá trị biết được của thuộc tính A.

S



Trở lại tập mẫu về c{c ng|y chơi cầu lông phù hợp với thời tiết cho trong bảng 3.1, thuộc tính gió có thể nhận giá trị yếu hoặc mạnh. ***S*** có phân bố [9+,5-+, trong đó tập mẫu giá trị thuộc tính *Tiết trời* là *nắng* có phân bố [2+,3-+ v| l| mưa có ph}n bố [3+,2-], các mẫu còn lại có thuộc tính này là nhiều mây với phân bố [4+,0-]. Thu hoạch thông tin thu được của S theo thuộc tính *Tiết trời* được tính như sau:

Values(tiết trời) = Nắng, Mưa,N\_m}y

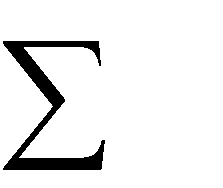
***S***  [9+,5–];

***S****nắng* [2+,3-], *Entropy*(*Snắng*) = -2/5. log(2/5) – 3/5.log(3/5)=0,971;

***S****mưa* [3+,2–], *Entropy*(*Smưa*) = -3/5.log(3/5)– 2/5. log(2/5)=0,971;

**S***N\_*mây [4+,0–], *Entropy*(*SN-mây*) = -4/4. Log(4/4)- 0.log(0/4)=0;

*Gain*(***S****,Tiết trời*) = *Entropy*(***S***) –  *Entropy*(***S*v**)



)

(

*Gio*

*Values*

*v*

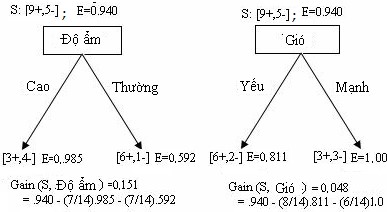
*v*

*S*

*S*

= *Entropy*(***S***) – (5*/14*)*Entropy*(***S****năng*) – (5/14)*Entropy*(***S****Mưa*) - (4/14)**S***N\_*mây = *0.94 – (5/14).0,971 – (4/14). 0,*00 - *(5/14).0,971* = *0.246.*

Tương tự, lượng thu hoạch thông tin của ***S*** theo các thuộc tính độ ẩm v| gió được tính và biểu thị tóm tắt trong hình 3.3, trong đó E để chỉ entropy.



**Hình 3.3.** Thu hoạch thông tin của S theo độ ẩm và gió

Bây giờ ta trở lại mô tả thuật toán với qui ước: *tập mẫu* là tập quan s{t được đang xét***,*** *tập nhãn* các nhãn lớp cần dự đo{n cho mẫu, còn *tập thuộc tính* là các thuộc tính không phải thuộc tính đích của mẫu. Để đơn giản, ta chỉ xét bài toán học khái niệm, độc giả có thể mở rộng thuật toán áp dụng cho bài toán học nhiều lớp.

*3.2.1.2. Mô tả thuật toán*

Thuật toán ID3 thực hiện như sau. Trước hết, mở một nút gốc cho cây, nếu nhãn của *tập mẫu* có giá trị như nhau thì nút n|y được gán nhãn chung này, còn nếu tập thuộc tính là rỗng thì nút được g{n nhãn theo đa số và thuật toán kết thúc. Ngược lại, thuộc tính tốt nhất được chọn làm nút gốc của cây và tạo ra các nhánh từ nút gốc tương ứng với các giá trị của thuộc tính được chọn.

Sau đó, tạo ra một nút con cho mỗi nhánh ứng với tập các mẫu huấn luyện sẽ được sắp xếp phù hợp với các nút con, tức là, xếp các mẫu v|o nh{nh tương ứng với giá trị của mẫu theo thuộc tính này và thuộc tính n|y cũng bị loại khỏi danh sách *tập thuộc tính*.

Toàn bộ qu{ trình trên được lặp lại cho các *tâp mẫu* mới ở c{c nút con để phát triển cây cho tới khi dữ liệu thuần nhất hoặc các thuộc tính đã kiểm tra hết thì nút trở th|nh nút l{ v| g{n nhãn tương ứng hoặc theo đa số. Kết quả cho một cây quyết định phân loại đúng (hoặc gần đúng nếu mẫu có nhiễu) các tập mẫu đ|o tạo. Thuật toán ID3 được mô tả trong bảng 3.2.

#### Ví dụ

Ta trở lại với bài toán dự báo một người chơi cầu lông theo đặc điểm thời tiết dựa trên tập mẫu ***D*** có phân bố [9+,5-+ được cho trong bảng 3.1.

**Bảng 3.2.** Thuật tuán ID3(Tập mẫu, tập nhãn, tập thuộc tính)

|  |
| --- |
| Bước 0. Khởi tạo: D, tập nhãn, tập thuộc tính;  Bước 1. Tạo nút gốc (Root) cho cây;  Bước 2. // Gán nhãn cho nút nếu dữ liệu thuần nhất hoặc tập thuộc tính là rỗng.  2.1. Nếu mọi mẫu đều dương tính thì nhãn nút gốc =    2.2. Nếu tất cả các mẫu là âm tính thì nhãn nút gốc =  2.3. Nếu tập các thuộc tính là rỗng trả về cây một nút gốc có nhãn = giá trị phổ  biến nhất của thuộc tính đích trong tập các mẫu;  2.4 C{c trường hợp kh{c sang bước 3; Bước 3.  3.1. X{c định thuộc tính phân loại tập mẫu tốt nhất trong tập thuộc tính;  3.2. A  thuộc tính phân lớp tốt nhất;  3.4. Với mỗi giá trị có thể vi của thuộc tính A, thực hiện:  3.4.1. Thêm một nhánh mới dưới nút gốc với mỗi điều kiện *A=vi;*  3.4.2. X{c định *Examplesv i* = {**x** *tập mẫu*: **x** có giá trị *vi* ở thuộc tính *A*; 3.4.3. Nếu Examplesv *i* là rỗng thì thêm dưới nhánh một nút lá có nhãn  là nhãn phổ biến nhất của các mẫu trong *tập mẫu;*  3.4.4. Ngược lại, trở lại bước 1 với khởi tạo:  (D = *Examplesv i* , *tập nhãn, tập thuộc tính – {A})* |

Vì tập mẫu không thuần nhất nên ta cần trả lời câu hỏi “Nên kiểm tra thuộc tính nào đầu tiên?” để gán nhãn cho nút gốc. Thuật toán ID3 xác định lượng *Thu hoạch thông tin* của các thuộc tính rồi chọn thuộc tính có lượng *Thu hoạch thông tin* cao nhất. Lượng *Thu hoạch thông tin* cho bốn thuộc tính theo các công thức (3.3) và (3.4):

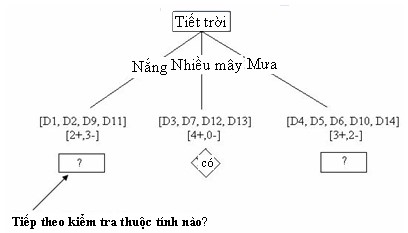
Gain(S, Tiết trời) = 0.246

Gain(S, Độ ẩm) = 0.151

Gain(S, Gió) = 0.048

Gain(S, Nhiệt độ) = 0.029

Thuộc tính *Tiết trời* là tốt nhất theo tiêu chuẩn thu hoạch thông tin cực đại. Vì vậy, thuộc tính Tiết trời được lựa chọn làm thuộc tính quyết định tại nút gốc, và các nh{nh được tạo ra dưới nút này ứng với các giá trị tương ứng bao gồm: Nắng, nhiều m}y, Mưa. Kết quả, một phần của cây quyết định được chỉ ra trong hình 3.4 cùng với các mẫu huấn luyện được phân loại cho mỗi nút con mới.



**Hình 3.4.** Phần cây quyết định với thuộc tính tiết trời tại gốc

Ta thấy mọi mẫu mà *Tiết trời* = nhiều m}y đều là mẫu khẳng định nên nút này của cây trở thành một nút lá với nhãn phân loại *Chơi cầu lông* = *Có*. Ngược lại, các nút con tương ứng với Tiết trời = Nắng và Tiết trời = Mưa vẫn có entropy khác *0*, và cây quyết định cần phát triển thêm nữa xuống dưới c{c nút n|y như trong hình 3.4. Nếu xét tập mẫu có thuộc tính tiết trời là nắng ta có Snắng = [D1, D2, D8, D9, D11] và cần tính độ thu hoạch thông tin của Snắng theo các thuộc tính Gió, Độ ẩm và Nhiệt độ để phát triển theo nhánh này.

Quá trình lựa chọn một thuộc tính mới và việc phân các mẫu huấn luyện sẽ được lặp lại với mỗi nút không phải là nút con cuối cùng và nó chỉ sử dụng các mẫu phù hợp với nút đang xét. C{c thuộc tính đã được tổ chức ở phía trên của c}y được loại ra, vì vậy bất kỳ một thuộc tính n|o được đưa ra cũng có thể xuất hiện nhiều nhất là một lần dọc theo bất kỳ đường đi n|o từ gốc tới lá trên cây. Quá trình này tiếp tục tới mỗi nút lá khi gặp phải một trong 2 trường hợp sau:

1. Mọi thuộc tính đều đã nằm trên đường đi n|y.
2. Các mẫu phù hợp với nút l{ n|y đều thuộc cùng một lớp (tức là entropy của chúng = 0)

Cây quyết định được xây dựng bằng thuật toán ID3 trong ví dụ n|y đã được chỉ ra trong hình 3.1.

*3.2.1.3. Nhận xét về ID3*

ID3 cho ta một phương ph{p hiệu quả để tìm một giả thuyết phù hợp với tập mẫu huấn luyện dưới dạng tập luật được biểu diễn bằng cây quyết định. Các luật phân lớp này tiện dùng trong c{c cơ sở tri thức của các hệ thông minh. Thuật toán có hiệu quả cả khi dữ liệu có nhiễu nhờ quyết định theo đa số khi có xung đột ở nút lá.

Tuy nhiên vẫn còn những vấn đề sau cần được khắc phục.

1. ID3 tìm kiếm giả thuyết trên toàn bộ không gian các hàm có giá trị rời rạc nên không gian giả thuyết này luôn chứa h|m đích. Tuy nhiên, khi gặp thuộc tính có giá trị liên tục thì thuật toán không áp dụng được.
2. ID3 thực hiện x}y dưng c}y từ đơn giản đến phức tạp theo phương thức tìm kiếm leo đồi được định hướng nhờ tiêu chuẩn thuộc tính có *Thu hoạch thông tin* *cực đại* trong không gian giả thuyết. Tiêu chuẩn n|y chưa chắc đã tốt, Chẳng hạn, đối với dữ liệu nhân sự có thuộc tính ngày sinh khác nhau cho mỗi mẫu thì thuộc tính n|y thường có lượng *Thu hoạch thông tin* lớn nhất nên cho ta cây độ sâu bằng 1 v| không có ý nghĩa. Có c{ch gì để khắc phục không?
3. ID3 không bao giờ quay lui trong quá trình tìm kiếm nên nó dễ mắc phải nhược điểm thường gặp của việc tìm kiếm leo đồi là tập trung vào các giải pháp tối ưu địa phương m| không phải là các giải pháp tối ưu to|n cục.
4. ID3 chỉ duy trì một giả thuyết đơn hiện thời khi tìm kiếm trong không gian của các cây quyết định nên nó đ{nh mất các khả năng mô tả rõ ràng tất cả các giả thuyết phù hợp. Chẳng hạn, nó không có khả năng x{c định xem có bao nhiêu cây quyết định có thể lựa chọn phù hợp với dữ liệu huấn luyện hay đưa ra những vấn đề mới đòi hỏi phải giải quyết tối ưu giữa các giả thuyết đang cạnh tranh này.
5. Phương ph{p dựa trên thống kê mà ID3 sử dụng để phát triển giả thuyết hiện thời của nó có ưu điểm là dùng thông tin của tất cả các mẫu, nhờ đó m| ít gặp lỗi hơn l| c{c thuật toán học trên từng mẫu riêng lẻ. Vì vậy ID3 có thể được mở rộng một cách dễ d|ng để dùng được cho dữ liệu huấn luyện có nhiễu bằng cách chấp nhận các giả thuyết không phù hợp lắm với dữ liệu huấn luyện. Tuy nhiên thuật to{n chưa cho c{ch xử lý dữ liệu bị mất.
6. Chiến lược tìm kiếm của thuật to{n ID3 có hai đặc điểm chính: Ưa c{c c}y ngắn hơn l| c{c c}y d|i, ưu tiên hơn cho c{c c}y quyết định nhỏ gọn hơn so với các quyết định phức tạp; Ưa những cây mà các thuộc tính có lượng *Thu hoạch thông tin* cao nhất nằm ở gần nút gốc nhất. Thuật toán tìm kiếm trên toàn bộ không gian giả thuyết nên c}y tìm được có thể rất lớn, khó áp dụng khi tập dữ liệu đ|o tạo lớn và có nhiều thuộc tính. Một câu hỏi đặt ra là: nếu cây quá lớn thì ta có thể “tỉa” cho nhỏ v| l|m đơn giản đi ra sao?
7. Khi dữ liệu chứa nhiễu thì có thể xảy ra hiện tượng *phù hợp trội* (overfitting), tức là cây quyết định tìm được phù hợp tốt trên dữ liệu đ|o tạo nhưng có thể đo{n nhận tồi hơn c}y kh{c đối với mẫu mới. Thuật toán ID3 xử lý theo phương ph{p thống kê nhưng ta chưa cho biết cách tránh hiện tượng phù hợp .

##### **3.2.2. Thuật toán C4.5**

Thuật to{n n|y do Quinlan (1993) đề xuất để khắc phục c{c nhược điểm của chính của ID3. Thuật toán này thực hiện theo lược đồ của ID3 nhưng có c{c cải tiến sau:

1. Ngoài việc áp dụng tiêu chuẩn *Thu hoạch thông tin cực đại,* C4.5 còn đề xuất sử dụng tiêu chuẩn *Tỷ lệ thu hoạch thông tin cực đại* (***Gainratio***)để dùng cho các trường hợp mà tiêu chuẩn trước áp dụng không tốt (nhược điểm thứ nhất của ID3).
2. Áp dụng kỹ thuật chặn sớm sự phát triển của cây dựa trên thống kê để tránh phù hợp trội và cây không quá lớn.
3. Đề xuất giải pháp xử lý trường hợp mẫu có thuộc tính thiếu giá trị 4) Đề xuất phương ph{p {p dụng cho thuộc tính nhận giá trị liên tục.

Ba giải ph{p đầu sẽ được b|n đến trong mục sau, ở đ}y sẽ giới thiệu phương pháp xử lý thuộc tính có giá trị liên tục của C4.5.

Giả sử dữ liệu có thuộc tính *A* có giá trị liên tục, trong tập mẫu quan s{t được , các mẫu nhận các giá trị khi được xếp thứ tự tăng dần là . Thuật toán C4.5 xử lý trường hợp n|y như sau. Đối với mỗi giá trị , chia tập mẫu thành hai lớp để tính thu hoạch thông tin: lớp thứ nhất gồm tập mẫu có giá trị thuộc tính *A* nhỏ hơn hoặc bằng và lớp thứ hai là các mẫu còn lại. *Thu hoạch thông tin/ Tỷ lệ thu hoach thông tin* của thuộc tính sẽ là cực đại của c{c đại lượng tính được theo các cách chia và tương ứng với nó sẽ được chọn l|m điểm chia rẽ nhánh cho thuộc tính này. Nếu thuộc tính liên tục n|y được chọn làm thuộc tính tốt nhất (theo tiêu chuẩn đang xét) thì tại nút tương ứng có thể g{n nhãn l| điều kiện kiểm tra () và hai nhánh tương ứng với giá trị logic như minh họa trong hình 3.2.

##### **Ví dụ**

Để minh họa, ta trở lại trường hợp các mẫu trong bảng 3.1 nhưng thuộc tính *Độ ẩm* nhận giá trị liên tục như được cho trong bảng 3.3.

Trong ví dụ này, sẽ dùng tiêu chuẩn *Thu hoach thông tin cực đại***,** việc dùng tiêu chuẩn *Tỷ lệ thu hoach thông tin cực đại* được tính tương tự. Ta sẽ tính lượng *Thu hoạch thông tin* cho thuộc tính này, tức là trả lời câu hỏi:

Trước hết, ta sắp các giá trị thuộc tính này của tập mẫu theo thứ tự tăng dần như sau: {*65, 70, 70, 70, 75, 78, 82, 82, 82, 86, 90, 90, 95, 96*} Sau đó loại bỏ các giá trị lặp ta có tập giá trị:

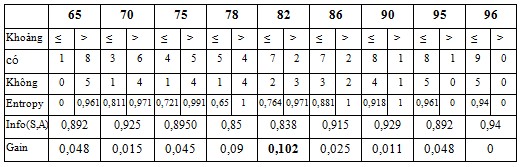
{*65, 70, 75, 78, 82, 86, 90, 95, 96*}

**Bảng 3.3.** Tập mẫu ở bảng 3.1 với giá trị thuộc tính Độ ẩm liên tục

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ngày | Tiết trời | Nhiệt độ | Độ ẩm | Gió | Chơi cầu lông |
| D1  D2  D3  D4  D5  D6  D7  D8  D9  D10  D11  D12  D13  D14 | Nắng  Nắng  Nhiều mây  Mưa  Mưa  Mưa  Nhiều mây  Nắng  Nắng  Mưa  Nắng  Nhiều mây  Nhiều mây  Mưa | Nóng  Nóng  Nóng  Mát lạnh lạnh lạnh Mát lạnh Mát  Mát  Mát  Nóng  Mát | 86  90  78  96  82  70  65  95  70  82  70  90  75  82 | Yếu  Mạnh  Yếu  Yếu  Yếu  Mạnh  Mạnh  Yếu  Yếu  Yếu  Mạnh  Mạnh  Yếu  Mạnh | Không  Không  Có  Có  Có  Không  Có  Không  Có  Có  Có  Có  Có  Không |

Ký hiệu Info(***S****,A*) l| lượng thu hoạch thông tin của tập ***S*** theo thuộc tính *A,* trong đó ***S*** là tập mẫu nhận được từ ***D*** nhờ phân chia nó theo giá trị thuộc tính tương ứng của thuộc tính này. Bảng 3.4 cho ta kết quả tính *Gain*(*D,Độ ẩm*) = 0,102 v| lượng thu hoặc thông tin ứng với mỗi c{ch chia. C{c bước kh{c được thực hiện như trong thuật toán ID3 và không mô tả lại để tránh nhàm chán.

**Bảng 3.4.** Tình Gain(D, Độ ẩm) sử dụng C4.5



Lưu ý rằng, phương ph{p xử lý này cho cây quyết đinh thường được dùng nhiều hơn tiếp cận rời rạc hóa bằng cách chia khoảng như trong thống kê truyền thống.

### 3.3. MỘT SỐ VẤN ĐỀ KHÁC TRONG HỌC BẰNG CÂY QUYẾT ĐỊNH

Bây giờ ta xem xét các vấn đề khác phát sinh khi học bằng cấy quyết đinh theo thuật toán ID3 và cách khắc phục trong C4.5 cùng với các phát triển của nó.

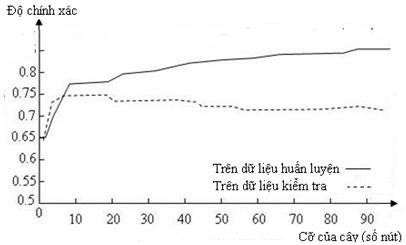
#### 3.3.1. Phù hợp trội và cách khắc phục

Các thuật toán trên phát triển mỗi nh{nh c}y đủ s}u để giả thuyết tìm được phù hợp với tất cả tập mẫu huấn luyện khi dữ liệu không có nhiễu và bài toán thiết lập đúng đắn. Tuy nhiên, khi dữ liệu có nhiễu thì giả thuyết tìm được có thể có lỗi đúng lớn hơn lỗi tìm được bằng tập mẫu ít hơn. Trong trường hợp đó có hiện tượng phù hợp trội. Sau đ}y l| định nghĩa chính x{c cho hiện tượng này.

***Định nghĩa 3.1.*** *(Phù hợp trội)* Cho một không gian giả thuyết ***H***, ta nói một giả thuyết ***h******H*** là phù hợp trội với dữ liệu huấn luyện nếu tồn tại một giả thuyết khác ***h*’** H mà ***h*** có sự sai lệch ít hơn ***h*’** trên tập các mẫu huấn luyện nhưng ***h*’** lại ít sai lệch hơn ***h*** trên toàn bộ tập mẫu có thể gặp.

Để đ{nh gi{ tính phù hợp trội, người ta phải chia dữ liệu quan s{t được thành hai tập, một để huấn luyện còn một (tập kiểm tra) để đ{nh gi{ tính phù hợp theo tiêu chuẩn thống kê.

Ảnh hưởng của sự phù hợp trội trong một trường hợp điển hình của phương pháp học bằng cây quyết định được minh họa trong hình 3.5. Trong đó, trục hoành của đồ thị chỉ tổng số các nút của cây quyết định tại thời điểm đang x}y dựng, trục tung chỉ độ chính xác của các dự đo{n từ c}y, đường nét liền chỉ độ chính xác của các cây quyết định trên tập mẫu huấn luyện, còn đường nét đứt chỉ độ chính xác trên tập mẫu kiểm tra. Thông thường, độ chính xác của cây quyết định trên tập đ|o tạo tăng dần đều khi c}y được phát triển.



**Hình 3.5.** Hiện tượng phù hợp trội khi nghiên cứu bệnh nội tiết

Tuy nhiên độ chính xác khi tính trên tập mẫu kiểm tra độc lập, ban đầu tăng lên sau đó giảm dần. Như đã thấy, khi cỡ của c}y vượt quá 20 nút, việc phát triển thêm nữa của cây làm giảm độ chính xác của nó trên tập các mẫu kiểm tra bất chấp sự tăng độ chính xác của nó trên tập mẫu huấn luyện.

Sở dĩ c}y *h* phù hợp với các mẫu huấn luyện tốt hơn ***h’***, nhưng lại xấu hơn trong các mẫu mới là vì tập dữ liệu huấn luyện chứa các mẫu sai, khi đó, c}y học được phù hợp với các mẫu sai này sẽ cho kết quả kiểm tra nhiều lỗi v| do đó lỗi đúng cũng lớn hơn.

Do tính ngẫu nhiên, nhiều khi dữ liệu không có nhiễu nhưng vẫn có thể xảy ra hiện tượng phù hợp trội. Chẳng hạn, cùng một thời tiết, người chơi có thể đi hoặc không đi với xác suất kh{c nhau. Trong trường hợp này, cây quyết định học được trên tập mẫu huấn luyện có thể có chất lượng tồi hơn một cây xấp xỉ h|m đích trên tập mẫu.

Có hai cách tiếp cận thông dụng để khắc phục hiện tượng phù hợp trội nhờ dựa vào kỹ thuật thống kê: tỉa bớt để giảm lỗi v| lược sau luật.

*Tỉa bớt để giảm lỗi*

Quinlan đề xuất phương ph{p n|y cho C4.5, trong đó, khi ph{t triển một nút mới cần dựa vào kết quả thống kê trên tập kiểm tra để đ{nh gi{ xem nên ph{t triển tiếp hay biến nó thành nút lá.

Đề xuất này thực chất là chặn sớm sự phát triển của cây. Tuy nhiên, trong các ứng dụng, người ta thường áp dụng tỉa c}y sau khi đã học xong cây quyết định người ta cắt bỏ cây con từ các nút trong có nghi vấn. Theo cách tiếp cận này, tập dữ liệu quan s{t được chia thành ba tập con: *tập huấn luyện*, *tập kiểm tra* để ước lượng độ chính xác và *tập đánh giá*. Ban đầu, người ta dùng *tập huận luyện* và *tập kiểm tra* để xây dựng c}y theo phương thức chặn sớm để có một cây với lỗi kiểm tra nhỏ nhất. Sau đó người ta dùng tập đ{nh gi{ để cắt bỏ các cành con (tỉa) nhằm giảm lỗi trên tập đ{nh giá.

Khi muốn *tỉa* một cành/ cây con từ một nút nghi vấn, người ta gán nhãn phổ biễn nhất của tập dữ liệu đ|o tạo phù hợp với nó (tức l| tương thích với đường đi từ gốc tới nút này), nếu cây mới thu được có lỗi thấp hơn trên tập đ{nh gi{ thì c|nh/c}y con n|y được tỉa và nút này trở th|nh nút l{. Trong trường hợp lỗi đ{nh gi{ không giảm thì giữ nguyên cây con trên cây.

Quá trình tỉa tiếp tục đến khi việc tỉa này là có hại (tức là nó làm giảm độ chính xác của cây trên tập đ{nh gi{). Bằng cách này, những nút gần lá bị loại liên tục v| độ chính xác của cây trên tập kiểm tra tăng lên.

Phương ph{p n|y cho phép ta “quay lui” trong việc tìm kiếm giả thuyết, khắc phục được nhược điểm thứ ba và thứ sáu của thuật to{n ID3. Hơn nữa, vì thuật toán ID3 ưu tiên c}y ngắn và các thuộc tính có thu hoạch thông tin cao thì ở gần gốc nên các nút gần gốc ứng với thuộc tính có thu hoạch thông tin cao không được tỉa, phương ph{p lược sau luật là một giải pháp tốt cho ngược điểm n|y v| nhược điểm thứ tư của ID3.

*Lược sau luật*

Mặc dù biện ph{p đầu tiên là trực tiếp hơn, nhưng trong thực tế, kỹ thuật lược sau luật lại th|nh công hơn việc tăng độ chính xác của giả thuyết tìm được. Kỹ thuật này thực hiện như sau:

1. Xây dựng cây quyết định từ tập huấn luyện cho đến khi dữ liệu huấn luyện phù hợp nhiều nhất đến mức có thể và cho phép xuất hiện phù hợp trội.
2. Chuyển đổi cây sang tập các luật { Nếu< thì<} tương đương bằng cách tạo ra một luật cho mỗi đường đi từ gốc đến lá.
3. Đơn giản (tổng quát hoá) mỗi luật bằng cách bỏ ra bất kỳ tiền điều kiện nào dẫn đến sự cải thiện độ chính xác của nó.
4. Sắp xếp các luật đã được lược bớt theo độ chính xác của nó, và xem xét chúng trong chuỗi n|y trong c{c trường hợp phân loại sau đó.

Để minh hoạ, ta xem lại cây quyết định trong hình 3.1. Mỗi thuộc tính kiểm tra theo một đường từ gốc đến l{ tương ứng với một tiền điều kiện của luật và kết luận phân loại tại nút l{ tương ứng. Chẳng hạn, nhánh bên trái nhất của c}y được luật hoá bởi luật:

*Nếu (Tiết trời = Nắng)* *(Độ ẩm = cao) thì ( chơi cầu lông = không)* Luật này có thể được tổng quát hóa thành một trong hai luật:

*Nếu (Tiết trời = Nắng) thì ( chơi cầu lông = không), hoặc Nếu Độ ẩm = cao) thì ( chơi cầu lông = không).*

Luật mới nào khi thay vào luật ban đầu trong tập luật tìm được mà làm giảm lỗi trên tập đ{nh gi{ nhiều nhất thì được thay thế. Còn không có luật nào giảm được thì thôi. Sau đó rút gọn tập luật mới tìm được.

#### 3.3.2. Tiêu chuẩn chọn thuộc tính

Ở trên, ta đã thấy rằng khi có một thuộc tính nhận quá nhiều giá trị thì lượng *Thu hoạch thông tin* của thuộc tính này thường lớn nhưng lại cho ít thông tin khi chọn nó làm nút gốc. Để khắc phục, Quinlan đề xuất có thể dùng *Tỷ lệ thu hoạch*( Gain Ratio) thay cho lượng *Thu hoạch thông tin*.

Tỷ lệ thu hoạch này phạt các thuộc tính có nhiều giá trị bằng cách thêm vào vào một hạng tử gọi là *thông tin chia* (SplitInformation), đại lượng này rất nhạy cảm với việc đ{nh gi{ tính rộng v| đồng nhất khi chia tách dữ liệu theo giá trị thuộc tính:

*SplitInformation(S,A)* (3.5)

*S*

*S*

*S*

*S*

*i*

*k*

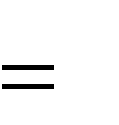
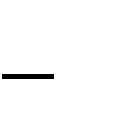
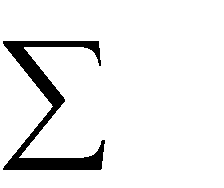
*i*

2

1

1

log



trong đó *S1* đến *Sk* là k tập con của các mẫu huấn luyện khi phân chia *S* theo *k* giá trị của thuộc tính *A* . Lưu ý rằng *SplitInformation(S,A)* thực tế là entropy của *S* ứng với giá trị của thuộc tính *A*. Khi đó *Tỷ lệ thu hoạch thông tin* của tập ***S*** đối với thuộc tính *A* được x{c định như sau:

*Gain*(*S*, *A*)

*GainRatio (S,A*) = (3.6)

SplitInformation (S, A)

Khi sử dụng *GainRatio* thay cho *Gain* để lựa chọn các thuộc tính thì nảy sinh vấn đề là mẫu số ở biểu thức (3.6) có thể bằng 0 hoặc rất nhỏ khi |*Si*| | *S* | . Khi áp dụng, người ta thường khắc phục hiện tượng này nhờ kết hợp cả hai tiêu chuẩn: Đầu tiên, tính *Gain* cho mỗi thuộc tính, sau đó {p dụng *GainRatio* cho các thuộc tính có giá trị *Gain* trên trung bình để chọn thuộc tính tốt nhất.

#### 3.3.3. Xử lý giá trị thuộc tính bị thiếu của mẫu

Trong ứng dụng, nhiều khi dữ liệu quan s{t được của ta có thể có các mẫu bị thiếu giá trị của một số thuộc tính, chẳng hạn, sử dụng bệnh án của bệnh nh}n để dự đo{n bệnh v| phương {n điều trị những kết quả xét nghiệm sinh hóa ở một số bệnh nh}n không có. Trong trường hợp này, các giá trị có được thường dùng để ước lượng giá trị thuộc tính thiếu theo hai cách:

Cách thứ nhất. Gán cho giá trị bị thiếu của mẫu bằng giá trị phổ biến nhất của các mẫu học tại nút đang xét v| cùng lớp với nó.

Cách thứ hai. Áp dụng cho trường hợp có nhiều mẫu bị thiếu giá trị ở cùng một thuộc tính. Lúc đó người ta gán giá trị ngẫu nhiên cho tập mẫu bị thiếu này với phân bố bằng tần suất xuất hiện của giá trị tương ứng trong tập mẫu tại nút đang xét. Chẳng hạn, giả sử thuộc tính *A* giá trị boolean và nếu nút đang xét chứa 70 mẫu đã biết với giá trị thuộc tính này bằng *1* và 30 mẫu đã biết có giá trị thuộc tính này bằng *0*. Khi đó ta ước lượng xác suất để *A*(*x*)=*1* là *0,7* và xác suất để *A*(x) = 0 là 0,3. Bằng cách này, các mẫu thiếu giá trị thuộc tính này ở nút đang xét được được tạo ra với tỷ lệ khoảng 0,7 xuống nhánh ứng với thuọc tính *A=1* còn với tỷ lệ khoảng *0,4* xuống nhánh thuộc tính *A=0* của c}y v| dùng chúng để tính thu hoạch thông tin cho bước phát triển.

#### 3.3.4. Cây phân lớp và hồi quy

Trong chương trước, ta đã l|m quen với phương ph{p giải tích để x{c định hàm hồi quy cho bài toán hồi quy với các thuộc tính v| h|m đều nhận giá trị thực. Nhiều trường hợp, tập thuộc tính có thể có cả thuộc tính nhận giá trị rời rạc và thuộc tính nhận giá trị thực. Trong c{c trường trường hợp này, ta có thể áp dụng kỹ thuật xử lý thuộc tính liên tục của Quinlan trong thuật toán C4.5 nêu trong mục 3.2.2 để xây dựng cây quyết định và gọi nó là cây hồi quy. Khi tập mẫu quan s{t được quá lớn, có thể lấy giá trị trung bình giá trị hàm của các mẫu đ|o tạo trên khoảng nhỏ có nhiều giá trị h|m rơi v|o.

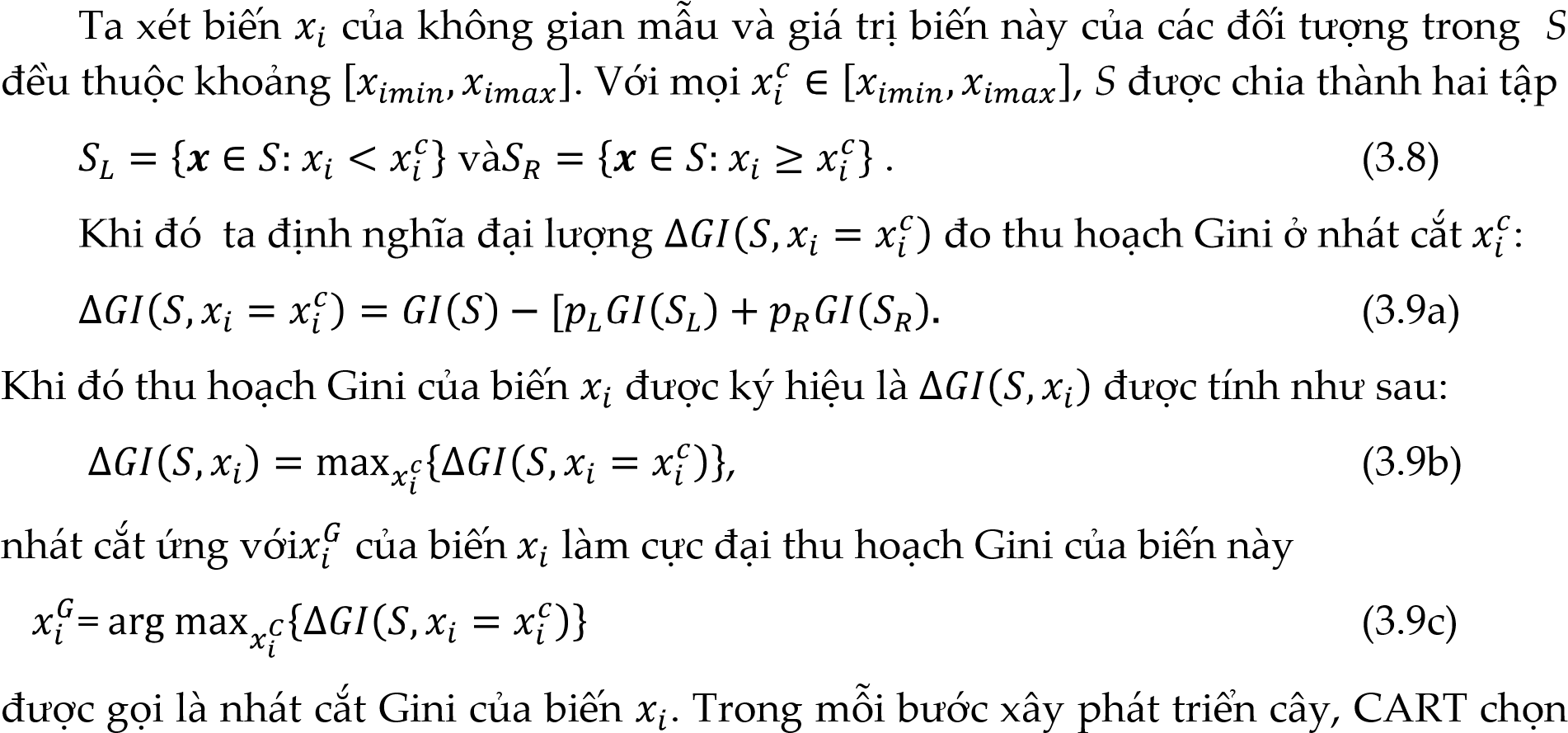
Một tiếp cận kh{c, phương ph{p c}y ph}n lớp và hồi quy (Classification and Regression Trees: CART) do Breiman và các cộng sự (1984) đề xuất, hiện nay thường được áp dụng cho bài toán phân lớp với giá trị thuộc tính thực, đặc biệt là cho các bài toán hồi quy. Phương ph{p n|y x}y dựng cây quyết định nhị phân dựa vào tiêu chuẩn thuần nhất Gini để chọn thuộc tính ưu tiên v| rẽ nhánh. Thuật toán này thực hiện tương tự như ID3, chỉ khác ở chọn thuộc tính kiểm tra và cách rẽ nhánh ở mỗi nút. Để đơn giản cho trình b|y, dưới đ}y xét c{c trường hợp giá trị thuộc tính thực.

*Độ thuần nhất Gini*

Xét tập mẫu quan sát *S* có nhãn thuộc tập *k* giá trị ,ký hiệu là tần suất mẫu có nhãn trong *S*. Khi đó độ thuần nhất Gini của *S* được ký hiệu là GI(S) v| được tính bới công thức:

(3.7)

*Thuộc tính ưu tiên và quy tắc chia*



biến có thu hoạch Gini lớn nhất và nhát cắt Gini của nó l|m điều kiện kiểm tra cho nút mới. Như vậy mỗi nút sé có hai nh{nh tương ứng với .

Cây quyết định trong hình 3.7 biểu diễn cây hồi quy của một hàm hồi quy làm

trơn c

ủ

a

hàm

có đ

ồ

th

ị

trong hình 3.6, trong đó t

ậ

p đi

ể

m đư

ợ

c đ{nh d

ấ

u chéo là d

ữ

li

ệ

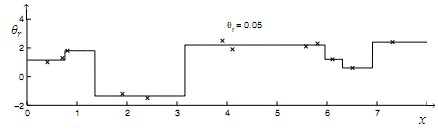
u đ|o t

ạ

o.

**Hình 3.6**

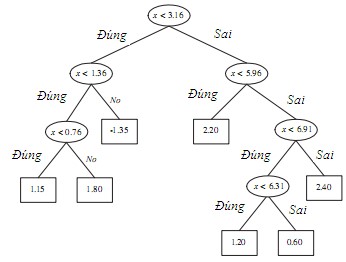
. Đồ thị hàm làm trơn cây hồi quy ở hình 3.7 của hàm của hàm



**Hình 3.7**

. Một cây hồi quy của hàm

có dữ liệu đào tạo trong hình 3.6



Mô tả chi tiết thuật toán và phát triển cho thuộc tính có giá trị rời rách dành cho độc giả như l| b|i tập. Những phát triển của CART, người ta sử dụng thêm các tiêu chuẩn thống kê để chọn nút kiểm tra và nhánh phát triển.

#### KẾT LUẬN

Cây quyết đinh l| một tiếp cận hữu hiệu cho các bài toán phân lớp với giá trị thuộc tính rời rạc. Thuật toán ID3 phát triển cây từ gốc xuống lá theo kiểu ăn tham để chọn thuộc tính phát triển và không quay lui nên có một số nhược điểm mà các thuật toán khác tìm cách khắc phục.

Phương ph{p học bằng cây quyết định có thể áp dụng hiểu quả cả khi dữ liệu quan s{t được bị nhiễu hoặc có mẫu bị thiếu giá trị thuộc tính. Thuật toán C4.5 cải tiến ID3 dùng được cho cả các bài toán dữ liệu có thuộc tính liên tục, đặc biệt, có thể dùng để xây dựng cây hồi quy.

Hiện tượng phù hợp trội là hiện tượng quan trọng trong học có giám sát. Phương ph{p tỉa c}y v| lược sau luật dựa trên kết quả thống kê là các giải pháp cho hiện tượng n|y. Trong đó, phương ph{p lược sau luật cho giả thuyết chính x{c hơn.

#### BÀI TẬP

1. Cho tập mẫu với 2 thuộc tính *A1* và *A2* như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Mẫu | *A1* | *A2* | Lớp |
| x1 | T | T | 1 |
| x2 | T | T | 1 |
| x3 | T | F | 2 |
| x4 | F | F | 1 |
| x5 | F | T | 2 |
| x6 | F | T | 2 |

* 1. Hãy tính Entropy tập dữ liêu theo phân lớp.
  2. Tính độ thu hoạch thông tin của thuộc tính a2

1. Xây dựng cây quyết định với dữ liệu cho trong bảng 3.2 và có thêm dữ liệu hai quan dữ liệu quan s{t được::

D15: <nắng, m{t, thường, yếu, có> và D16: <nắng, m{t,thường, mạnh>.

1. Giải thích vì sao nói tiêu chuẩn thu hoạch thông tin có khuynh hướng ưu tiên chọn thuộc tính có nhiều giá trị?
2. Mô tả thuật to{n CART cho trường hợp giá trị thuộc tính thực.
3. Mô tả thuật to{n CART cho trường hợp giá trị thuộc tính rời rạc.
4. . Mỗi đối tượng có hai thuộc tính đặc trưng, đặc trưng thứ nhất nhận hai giá trị 0 hoặc 1, đặc trưng thứ hai nhận giá trị trong 5 chữ cái. Dự liệu quan s{t được gồm hai lớp cho bởi bảng:

|  |
| --- |
| : 1A 0E 0B 1B 0D  1 |
| : 0A 0C 1C 0B 1D  2 |

1. Xây dựng cây quyết định theo thuật toán ID3 sử dụng tiêu chuẩn thu hoạch thông tin
2. Xây dựng cây quyết định theo thuật toán ID3 sử dụng tiêu chuẩn tỷ lệ thu hoạch thông tin.
3. Xây dựng cây quyết định nhờ dùng các thuật toán ID3 và CART nếu dữ liệu quan sát của hai lớp có hai đặc trưng thu được như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Mẫu | *a1* | *a2* | Lớp |
| ***x1*** | *0,15* | *0,84* | 1 |
| ***x2*** | *0,8* | *0,54* | 1 |
| ***x3*** | *0,28* | *0,36* | 1 |
| ***x4*** | *0,38* | *0,70* | 1 |
| ***x5*** | *0,52* | *0,48* | 1 |
| ***x6*** | *0,58* | *0,74* | 1 |
| ***x7*** | *0,74* | *0,76* | 1 |
| ***x8*** | *0,08* | *0,14* | 2 |
| ***x9*** | *0,24* | *0,16* | 2 |
| ***x10*** | *0,70* | *0,20* | 2 |
| ***x11*** | *0,90* | *0,28* | 2 |
| ***x12*** | *0,64* | *0,90* | 2 |
| ***x13*** | 0,74 | 0,36 | 2 |
| ***x14*** | 0,10 | 0,30 | 2 |

1. Xây dựng cây hồi quy cho hàm số ở bài tập 6 chương 2

**Chƣơng 4**

## PHÂN LỚP MẪU NHỜ HÀM PHÂN BIỆT

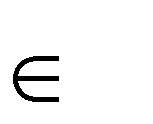
Hai chương trước đã giới thiệu c{c phương pháp phân lớp bằng cách tìm tập luật biểu diễn h|m đích dạng liên kết giá trị-thuộc tính với các thuộc tính nhận giá trị trong tập hữu hạn. Trong thực tế, ta thường gặp bài toán phân lớp m| c{c đối tượng có đặc trưng l| vectơ n-chiều. Chương n|y giới thiệu tiếp cận giải tích để xử lý bài toán này nhờ sử dụng các hàm phân biệt tuyến tính.

### 4.1. HÀM PHÂN BIỆT VÀ MIỀN QUYẾT ĐỊNH

Trong các bài toán phân lớp mẫu, mỗi mẫu mỗi mẫu ***x*** trong tập đối tương ***X*** thường được biểu diễn bởi *n* đặc trưng: ***x*** *= (x1,…,xn)*. Để phân ***X*** thành *k* lớp , người ta thường dùng *k* ***hàm phân biệt*** hay *còn gọi là* ***hàm quyết định***:

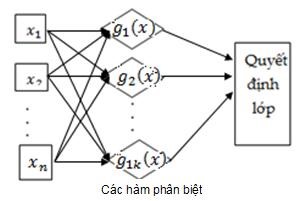
*,* mỗi h|m x{c định một lớp. Mỗi đối tượng có vectơđặc trưng ***x*** được gán

cho lớp *i* nếu giá trị hàm phân biệt lớn nhất, tức là:



***x*** *i* nếu (4.1)

Bằng cách này, bộ phân lớp được xem như một mạng hay là một m{y dùng để tính *k* hàm phân biệt và chọn lớp ứng với giá trị hàm phân biệt lớn nhất để gán cho mỗi đối tượng như được minh họa trong hình 4.1.



**Hình 4.1.** Cấu trúc bộ phân lớp có n đầu vào và k hàm phân biệt

Trong trường hợp c{c đặc trưng nhận giá trị thực, các hàm phân biệt chia không gian thành *k* miền , mỗi miền x{c định một lớp:

v| được gọi là ***miền quyết định*** của lớp *i* tương ứng. Khi đó, quy tắc quyết định là:



*x* thuộc (được gán vào lớp *i* nếu . (4.2a)

Các miền quyết định n|y được tách bởi các ***biên quyết định***, chúng là hợp của các phần mặt/siêu mặt trong không gian đặc trưng chứa c{c điểm có giá trị hàm phân biệt lớn nhất bằng với giá trị của ít nhất một hàm phân biệt khác. Các biên quyết định tách các lớp trong không gian đặc trưng.

Khi chỉ có hai lớp và , biên quyết định còn được gọi là ***mặt/siêu mặt*** (khi *n* > *3*) ***quyết định***. Trong trường hợp này, thay vì dùng hai hàm phân biệt *g1(x)* và *g2(x)* ta chỉ cần dùng một hàm phân biệt *g*(*x*) l| đủ:

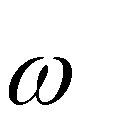
*g(****x****) = g1(****x****) – g2 (****x****)* (4.2b) và quy tắc quyết định lúc đó l|:

***x*** thuộc nếu *g(x)>0* và thuộc nếu *g(****x****) < 0*. (4.2.c)

1



2

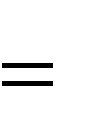
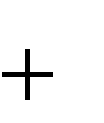
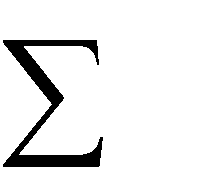


Lúc này mặt/siêu mặt (biên) quyết định là tập ***{x******Rn: g(x) = 0}.***

*Hàm phân biệt tuyến tính*

Trường hợp không gian đặc trưng *n*-chiều, hàm phân biệt tuyến tính được biễu diễn bởi các trọng số ***w0,*** *w1,...,wn* và có dạng:

*g i* (*x*) *w*0 *=* ***w’x*** *+w0.* (4.3a)



*n*

*i*

*i*

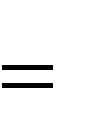
*i*

*x*

*w*

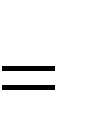
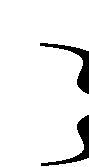
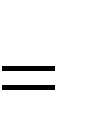
1

Công thức (4.3a) có thể viết gọn lại là:

*gi* (*x*) ***w’x*** *+w0=* ***w\*’ x\****, (4.3b)

trong *đó* ***w\**** *= [w0,w1,...,wd]*’, v| ***x\**** *=* [*1,x1,...,xd*+’ (dấu ’ để chỉ vectơ chuyển vị).

Như đã nói ở trên, trường hợp hai lớp, các *mặt*/*siêu mặt* quyết định được xác định bởi tập *H x*/ *g*(*x*) 0 là các *mặt phẳng/siêu phẳng quyết định* được đặc trưng bởi khoảng cách ***D0*** từ gốc tọa độ v| vectơ ph{p tuyến ***n*** cho bởi công thức:

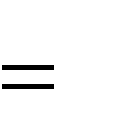


***D***0 ; ***n***.(4.4a)

0

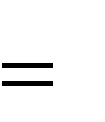
*w*

*w*



*w*

*w*



*H* chia không gian đặc trưng th|nh hai nửa không gian: ứng với lớp và ứng với lớp . Hàm phân biệt đo khoảng cách từ ***x*** tới *H*. Để trực quan hóa, ta xét biểu diễn của ***x*** dưới dạng:

, (4.4b)

trong đó là hình chiếu trực giao của ***x*** lên *H* còn *r* là khoảng c{ch đại số từ ***x*** tới *H* (dương nếu ***x*** thuộc miền dương v| }m nếu ***x*** thuộc miền âm của hàm quyết định).

B

ở

i vì

nên

hay

.

(4.4c)

M

ộ

t minh h

ọ

a hình h

ọ

c đư

ợ

c gi

ớ

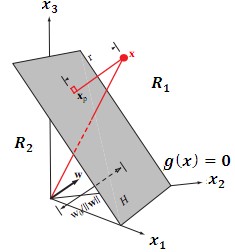
i thi

ệ

u trong hình 4.2.

**Hình 4.2.**

Một mặt phẳng quyết định H trong



**Chú ý**. Khái niệm hàm tuyến tính ở đ}y kh{c với hàm tuyến tính của họ h|m cơ sở trong mục 2.4 ở chương 2.

*Các dạng hàm phân biệt khác*

Phân biệt tuyến tính có cơ sở toán học vững chắc, cho phép ta trả lời được các câu hỏi “ có tồn tại hàm phân biệt không? nếu có thì tìm như thế n|o?”, tuy nhiên miền áp dụng được khá hạn chế. Khi mở rộng không gian giả thuyết nhờ dùng dạng hàm phân biệt phức tạp hơn thì khó x{c định hàm tách các tập mẫu quan s{t được. Vì vậy, dạng hàm phân biệt thường được chọn đủ đơn giản để dễ x{c định nhưng phải đ{p ứng được việc phân biệt các lớp. Hàm phân biệt là tổ hợp tuyến tính của c{c h|m cơ sở cho trước là một dạng h|m được ưa dùng. Cho trước họ h|m độc lập tuyến tính , dựa trên tập mẫu quan s{t được, ta tìm hàm phân biệt dạng:

(4.5.a)

trong đó là các tham số cần tìm. Nói riêng, khi hàm phân biệt l| đa thức thì các tham số là các hệ số của đa thức. Hình 4.5 minh họa biên quyết định cho trường hợp hai lớp trong không gian hai chiều với hàm phân biệt là một đa thức bậc bốn:

*g*(*x*) *w*7 *x*12 *x*2 

0

2

1

1

2

2

1

3

2

2

4

2

1

5

2

2

1

6

3

2

8

3

1

9

3

2

1

10

2

2

2

1

12

4

2

13

4

1

14

*w*

*x*

*w*

*x*

*w*

*x*

*x*

*w*

*x*

*w*

*x*

*w*

*x*

*x*

*w*

*x*

*w*

*x*

*w*

*x*

*x*

*w*

*x*

*x*

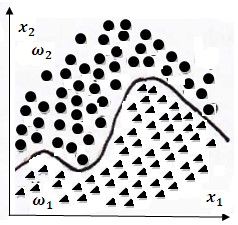
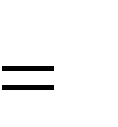
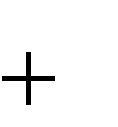
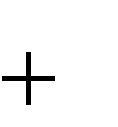
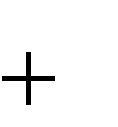
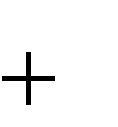
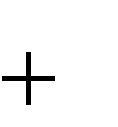
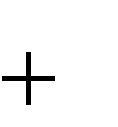
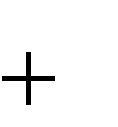
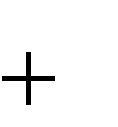
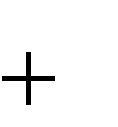
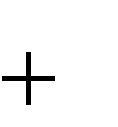
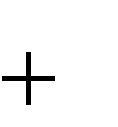
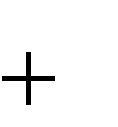
*w*

*x*

*w*

*x*

*w*



(4.5b)

**Hình 4.3.** Miền và biên quyết định đối với hàm phân biệt bậc bốn

Trong không gian đặc trưng 2-chiều, với hàm phân biệt bậc 4 ta cần x{c định 15 trọng số (14 trọng số đa thức và một khuynh hướng). Tổng qu{t hơn, khi dùng h|m phân biệt bậc *k* cho không gian đặc trưng *d*-chiều ta cần x{c định :

*Cdk* hệ số đa thức (4.5c)

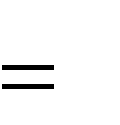
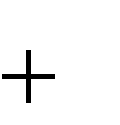


!

*k*

*d*

*k*



*d*!*k*!

Với *d=2; k=4* ta có :

*C k* 15 hệ số da thức .



2

6

.

5

!

4

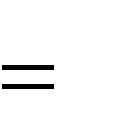
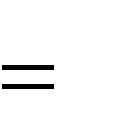
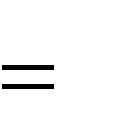
!

2

!

4

2



*k*

*d*

### 4.2. CÁC MÔ HÌNH TUYẾN TÍNH

#### 4.2.1. Tách đƣợc bởi siêu phẳng

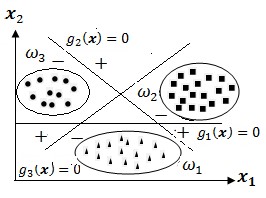
Thông thường các mẫu trong mỗi lớp được đặc trưng bởi một phân bố ngẫu nhiên. Trong hình 4.1, các ellip biểu diễn “***biên***” của phân phối và còn gọi là ***giới hạn lớp***. Khi có nhiều lớp, các hàm phân biệt tuyến tính có thể tách các lớp dưới hai dạng: *tách tuyệt đối* và *tách từng cặp*. Ta gọi chung c{c trường hợp n|y l| t{ch được bởi siêu phẳng hay t{ch được tuyến tính.

*Tách tuyệt đối*

Khi mỗi lớp được tách hẳn khỏi các lớp còn lại ta nói tập mẫu t{ch được tuyệt đối. Hình 4.4 minh họa một trường hợp ba lớp t{ch được tuyệt đối trong .

**Hình 4.4.**

Ví dụ tách tuỵệt đối bới các siêu phẳng trong



Khi các lớp t{ch được tuyệt đối thì thì hệ nhận dạng sẽ có cấu trúc phân cấp mô tả bởi cây quyết định nhị ph}n như minh họa trong hình 4.5, trong đó

*g23(****x****)*= *g2*(***x*)** – *g3*(***x***) để tách

1

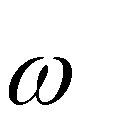


v

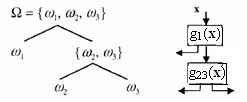
ớ

i

2



.



**Hình 4.5.** Cấu trúc phân cấp của hệ phân 3 lớp tách được tuyệt đối

*Tách theo cặp*

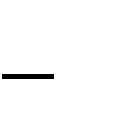
Nhiều khi tập mẫu không t{ch được tuyệt đối nhưng có thể tách theo từng cặp như minh họa trong hình 4.6.

**Hình 4.6.**

Ba lớp tách được từng cặp trong



Chú ý rằng c{c trường hợp t{ch được từng cặp không mô tả được bằng cây quyết *k*(*k* 1)



định. Trường hợp n|y người ta x{c định siêu phẳng tách từng cặp lớp nhờ 2

dùng các hàm phân biệt thỏa mãn:

*gij*(***x***) > 0 và *gij*(***x***) < 0 , (*gij*(***x***) = - *gij*(***x***) ). (4.6a)



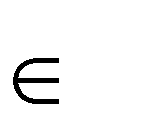
***x***



*i*



***x***



*j*



Miền quyết định lúc này là:

*Ri* ={***x***; *gij*(***x***) > 0, *i,j = 1,...,k, j≠i* (4.6b)

Để xây dựng các bộ phân lớp, ta dùng các thuật to{n tìm vectơ **w** và hệ số w0 của hàm phân biệt như trong biểu thức (4.3a). Các thuật toán này còn gọi là thuật toán học hay đ|o tạo.

#### 4.2.2. Thuật toán học perceptron

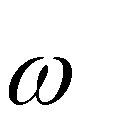
Để đơn giản, ta xét bài toán có 2 lớp v| không gian đặc trưng n-chiều. Cho tập dữ liệu quan s{t được *D = D1*  *D2* trong đó *Di* là các mẫu thuộc lớp tương ứng, ta tìm véctơ ***w*** và hệ số *w0* sao cho:

1



,

2

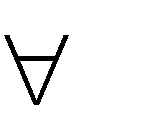
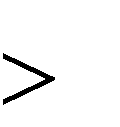
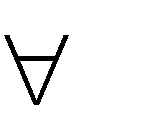
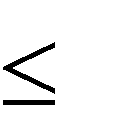
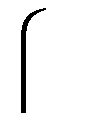
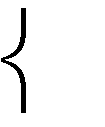
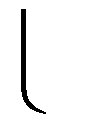


*i*



*g(****x****) =* ***w****’****x*** *+w0 >0* với mọi ***x****D1* và *g*(*x*)<0 với mọi ***x****D2*. (4.7) Trước hết ta định nghĩa hai h|m *hardlim*: *R→ R* và *t*: *D → R* như sau:

*hardlim*(***x***) = (4.8a) *t*(***x***)= 1 (4.8b)



0

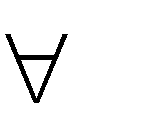
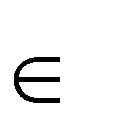
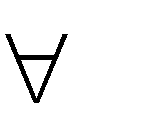
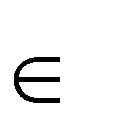
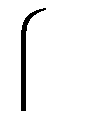
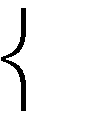
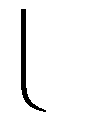
0

0

1

*x*

*x*



0

1

*D*

*x*

*D*

*x*

2

Với mỗi ***w****, w0* đã biết và ***x****D*, đặt:

*a(****x****) = hardlim(****w’x****+w0)* (4.8c) *e*(**x**) = *t*(***x***)- *a*(***x***) (4.8e)

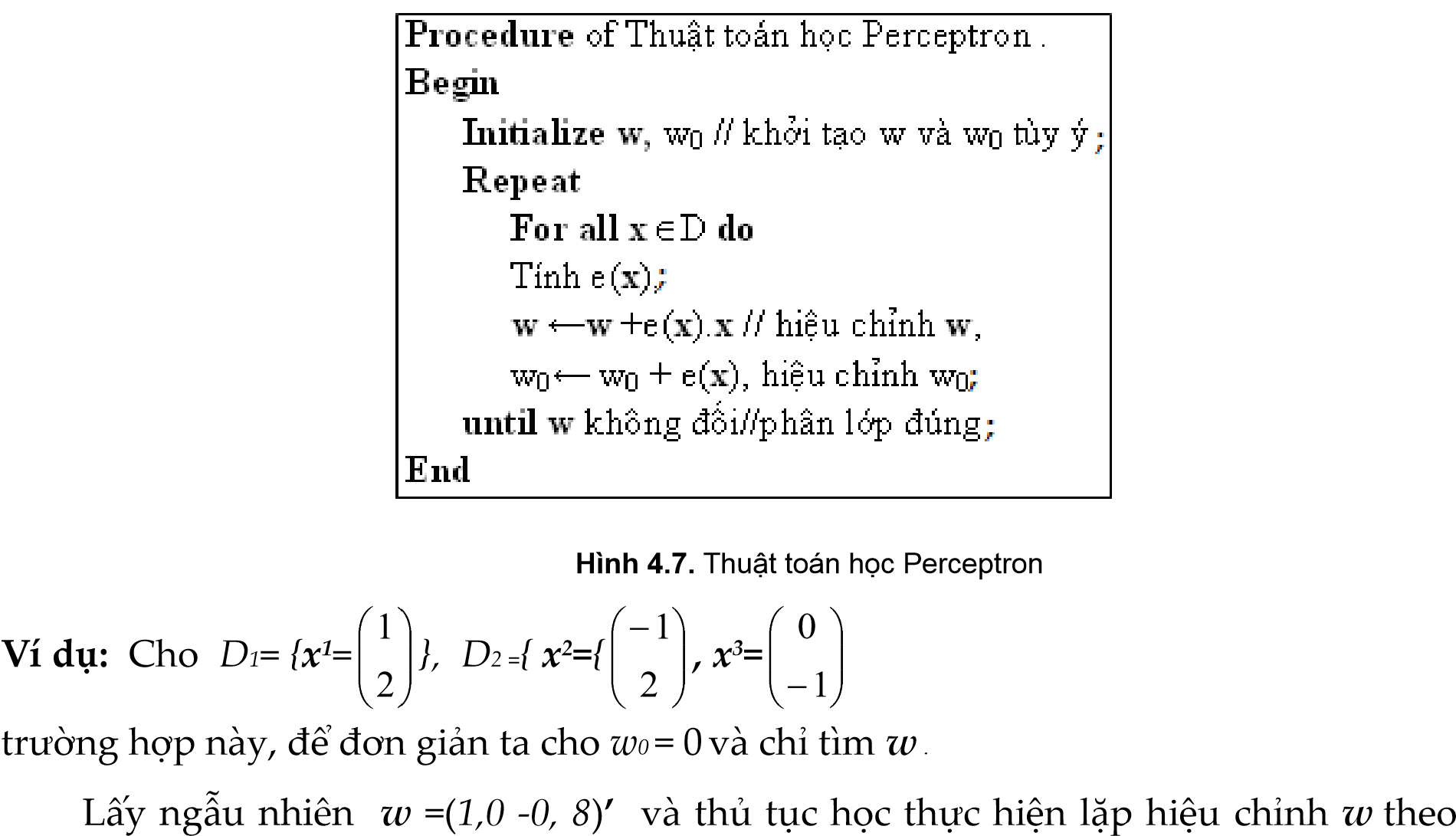
Nếu *D1* và *D2* t{ch được bởi siêu phẳng thì thuật toán học perceptron do Rosenblatt đề xuất (1953) luôn hôi tụ sau hữu hạn bước. Thuật toán khởi tạo ngẫu nhiên các giá trị ban đầu cho ***w*** và *w0*, sau đó thực hiện lặp thủ tục lấy lần lượt các mẫu quan s{t được trong *D* để tính *e(****x****)* và hiệu chỉnh các tham số cần tìm theo quy tắc:

nếu *e(x)=0* (nhận dạng đúng) thì để nguyên (4.9a) nếu *e(x)= 1* thì ***w*** *mới=* ***w****cũ+****x* ;**  *w0mới=w0cũ +1* (4.9b nếu *e(x)=-1* thì***w*** *mới=* ***w****cũ-****x ;***  *w0mới=w0cũ -1* (4.9c) Các công thức (4.14a-c) có thể viết chung là:

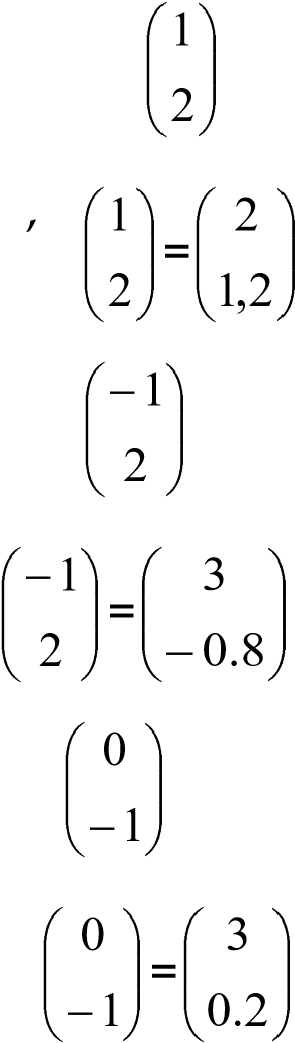


***w*** *mới=* ***w****cũ + e(****x****)****x ;***  *w0mới = w0cũ + e(****x****)* (4.9d)

Thuật to{n được đặc tả trong hình 4.7.

*}* như đặc tả trong hình 4.7. Trong

các mẫu quan sát:

* Cho ***x= x1***khi đó *a*(***x*1**)**=**hardlim(***w’ x*1**)=*hardlim*((*1,0 -0,8*))= *hardlim(-0,6*)=*0*

*e*(***x1*)**=*t-a*=*1-0=1*: ***w****mới* = ***wcũ****+e*(**x1**) ***x1*= *w*** *+* ***x1*=**(*1,0 -0,*8) +

* Cho ***x=x2*** khi đó *a*(***x*2**)**=***hardlim*(***w’x2***)=*hardlim*((2*,0 1,2*))= *hardlim*(0,44) *=1*

*e*(***x2***)=*t-a*=*0-1=-1,* ***w****mới =* ***w****cũ+e*(**x**)***x2*=*w****-****x2=****(2,0 1,2)****’****-*

* Cho ***x=x3***khi đó *a*(**x3**)**=***hardlim*(***w’x3****)*=*hardlim*((*3 -0.8*))= *hardlim*(*0.8*)=*1*

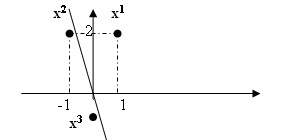
*e(****x2****)= t-a = 0-1= -1,* ***w****mới =* ***w****cũ+e(****x****)****x2=w****-****x2=*** (*3 -0.8*)’-

Bây giờ ở vòng lặp thứ hai ta thấy *e(****x1****)= e(****x2****)=e(****x3****)=0* và ***w*** không đổi. Vậy ta có hàm phân biệt được cho trong công thức (4.14d) v| được minh họa trong hình 4.8.

*g(****x****) = 3x1+ 0,2 x2* (4.9e)

**Hình 4.8**

. Ví dụ tách hai tập



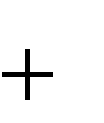
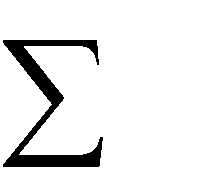
**Chú ý**. Thuật toán này chỉ dừng khi tập mẫu *D* l| t{ch được tuyến tính, ngoài ra, có thể có nhiều siêu phẳng tách khi chọn giá trị khởi tạo khác nhau.

Để khắc phục nhược điểm của thuật toán Peceptron, thuật to{n sau đ}y cho một giải pháp lựa chọn.

#### 4.2.3. Thuật toán bình phƣơng tối thiểu

Ta xét b|i to{n như mục trên với tập quan s{t được *D*= {***xi***} (*i=1,...,N*) đã biết nhãn *t*(***x****i*) tương ứng, *a*(***x***) tính bởi:

*a(****x****)=w*0 *.* (4.10a)



*N*

*i*

*i*

*i*

*x*

*w*

1

Quy tắc phân lớp sẽ là:

(4.10b)

Nói chung tập mẫu n|y không đảm bảo t{ch được bởi siêu phẳng v| không đảm bảo thuật toán học perceptron hội tụ nên Widrow v| Hoff (1960) đề xuất dùng phương ph{p bình phương tối thiểu (LMS) để x{c định ***w*** và *w0*.

*Sai số trung bình phương*

Với tập *D*, vectơ ***w*** và giá trị *w0* đã cho, sai số trung bình phương của bộ phân lớp là:

*E = E (D,* ***w****, w0)=* 1 *N i) – a(****x****i)]2*, (4.11a)



[ *t(****x***

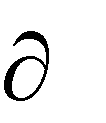
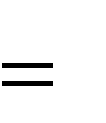
##### *N i* 1

trong đó *t*(***x***) tính theo (4.8b) còn a(**x**) tính theo (4.10a). Khi ***x***lấy ngẫu nhiên cùng phân bố với *D* trong không gian đặc trưng thì *E* (*D,* ***w****, w0*) là xấp xỉ của kỳ vọng bình phương sai số:

*E[t(****x****i) – a(****x****i)]2*. (4.11b)

Các trọng số *wi* (1=0,...,n) sẽ được tìm để *E* trong biểu thức (4.11a) đạt cực tiểu. Tương tự như trong mục 2.4 chương 2, b|i to{n tìm cực tiểu hàm *E* luôn có nghiệm và có thể tìm cực tiểu nhờ giải hệ phương trình:

0 với k=0,...,n (4.12a)



*k*

*w*

*E*

Để ý rằng các *t*(***xi***) là hằng số còn *a*(*xi*) tuyến tính đối với các *wk*. hệ phương trình (4.12a) sẽ tương đương với hệ 4.12b-c), trong đó (4.12c) ứng với đạo hàm theo biến *w0*:

[ *t(****x****i) – a(****x****i)*] *xki =0 , k=1,..., N* (*4.12b*)



*N*

*i*

1



*N*

*i*

1

[*t(****x****i) – a(****x****i)*] *= 0* (4.12c)

Khi *N* lớn, việc giải hệ (4.17b-c) sai số lớn v| độ phức tạp là *O(N3)* nên khó thực hiện. Widrow v| Hoff đề xuất dùng phương ph{p gradient để tìm các *wk*, hơn nữa họ chỉ ra rằng nếu lấy mẫu ngẫu nhiên trong *D* thì thay cho sai số *E (D,* ***w****, w0)* ở bước *k* có thể dùng xấp xỉ bởi sai số *E (****x****) = [t(****y****k)- a(****y****k)]2= e(****y****k)2*trong đó ***y****k*là mẫu lấy ngẫu nhiên trong *D* ở bước lặp *k* . Thuật toán thực hiện như sau. Sau khi khởi tạo ngẫu nhiên các kệ số *w*i (i=0,<,n) ban đ}u, thực hiện lặp theo *k*. Lấy ngẫu nhiên ***y****k* trong tập mẫu *D* và hiệu chỉnh trọng số theo công thức:

***w****mới =* ***w****cũ +2**e(****y****k)* ***y****k* (4.13a) *w0mới =w0cũ + 2* *e(****y****k)* (4.13b)

trong đó  là tốc độ học thỏa mãn điều kiện 0< α <1/ λmax với λmax là giá trị riêng lớn nhất của ma trận tương quan mẫu :

1 *N k(****x****k)’+*  (4.13c)



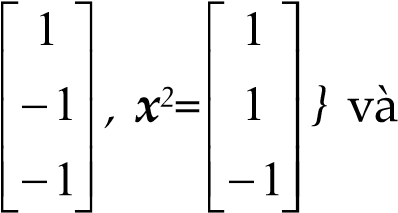
*R =E(****xx’****) =* [ ***x***

##### *N i* 1

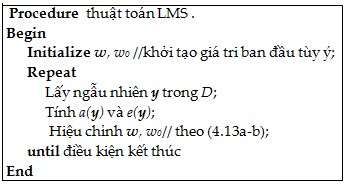
Điều kiện kết thúc có thể là số lần lặp hoặc độ lệch của sai số trung bình phương nhỏ hơn  cho trước. Thuật to{n được đặc tả trong hình 4.9.

**Chú ý**. Tốc độ học α có thể lấy bằng phương ph{p thử sai hoặc giảm dần về không mà không phải tính giá trị riêng của *R*.

**Ví dụ**.

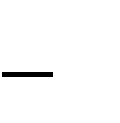
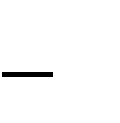
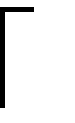
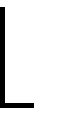
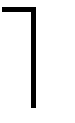
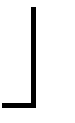
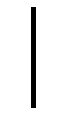
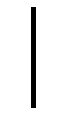
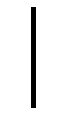
Cho *D* gồm hai vectơ *{****x****1=* *{ t (****x****1) =-1]} t(****x****2)=1* (ở đ}y *t* lấy *±1*

cho chóng hội tụ).



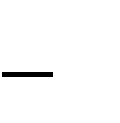
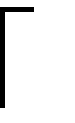
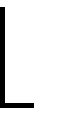
**Hình 4.9.** Thuật toán Widrow-Hoff

1. *1(****x****1)’* ***+*** 1 ***x****2(****x****2)’ =*1 *[1 -1 -1]+[1 1 -1]* Ta có ***R=****E(****xx’****)=*  ***x***



1

1



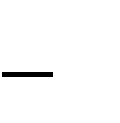
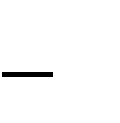
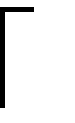
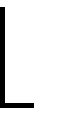
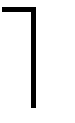
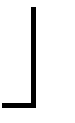
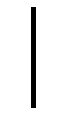
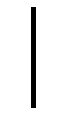
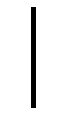
1

1

1

1. 2

=



1

0

1

0

1

0

1

0

1

***R***

*λ*

*1*

*=0*

*,1,*

*λ*

*2*

*=0*

*,*

*λ*

*3*

*=2*

*.*

V

ậ

y t

ố

c đ

ộ

h

ọ

c

ổ

n đ

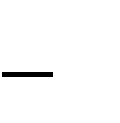
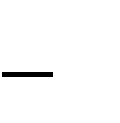
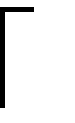
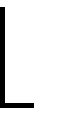
ị

nh là

α <*1/* *λmax = 1/2=0,5.*

Ta lấy α =0,2 và **w**(0)=\*0 0 0+’ để đơn giản luôn lấy w0=0 và tín hiệu vào lần lượt là ***x****1,****x****2,* ***x****1,* ***x****2,....*

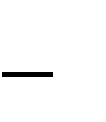
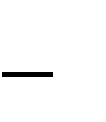
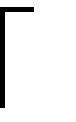
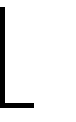
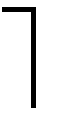
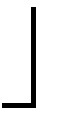
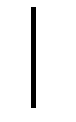
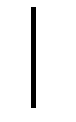
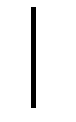
* Với ***y1*** *=* ***x****1=* 1 ; *t(****x****1)=-1* thì ta có



1

1

*a(****y****1)=* ***w*** *(0)’****y1****=* ***w****(0)’* ***x****1 =[0 0 0] = [0].*



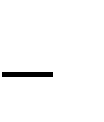
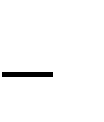
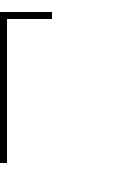
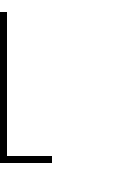
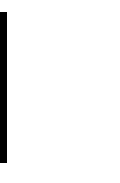
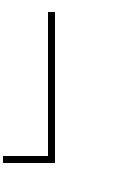
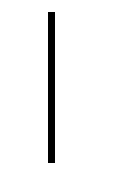
1

1

1

*e(****y1****)=t(****y1****)-a(****y1****) = -1 - 0 = -1*

***w****(1)=****w*** *(0)+ 2αe(****y1****)* ***y1*** *= [0 0 0] +2(0.2)(-1) =[-0,4 0,4 0,4+’*



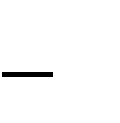
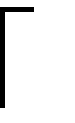
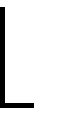
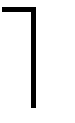
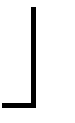
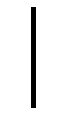
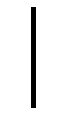
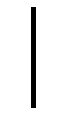
1

1

1

* Với ***y2****=* ***x****2 , t(****x****2)=1* tacó :

*a(****y2****) =* ***w(y2****)’* ***y2****=[-0,4 0,4 0,4] = -0,4;*



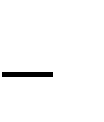
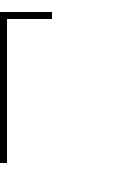
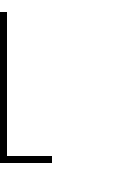
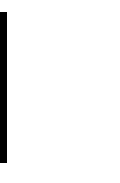
1

1

1

*e(****y2****) = t(****y2****1)-a(****y2****) = 1-(-0,4) = 1,4*

***w****(2)=****w****(1)+ 2αe(****y2****)* ***y2*** *= [-0,4 0,4 0,4] +2(0.2)(1,4) =[0,16 0,96 -0,16+’*

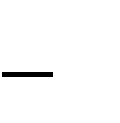
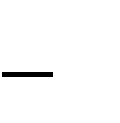
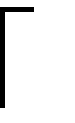
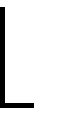


1

1

1

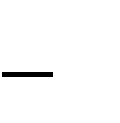
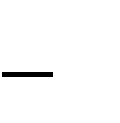
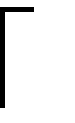
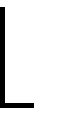
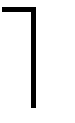
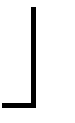
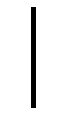
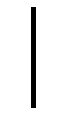
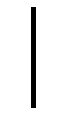
* Với ***y3*** *=* ***x****1=* 1 *; t(****y3****) = -1* ta có :



1

1

*a(****y3****) =* ***w(y3****)’* ***y3****=[0,16 0,96 0,16] = -0,64*



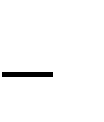
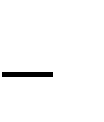
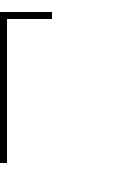
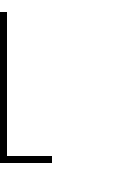
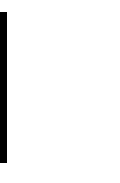
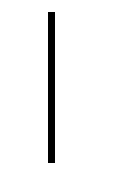
1

1

1

*e(****y3****)= t(****y3****)-a(****y3****) = -1-(-0,64)=-0,36*

***W****(3)=****W****(2)+ 2αe(****y3****)* ***y3****= \*0,16 0,96 0,16+’ +2(0.2)(-0,36) =[0,016 1,1040 -0,0160+’*.



1

1

1

***w****(k)* hội tụ tới *[0 1 0].*

Widrow v| Hoff đã chứng minh rằng với tốc độ học đủ nhỏ thì thuật toán luôn hội tụ tới nghiệm duy nhất của (4.12a).

Hai phương ph{p trên không chú ý tới đặc điểm phân bố của mỗi lớp trong tập *D*, khi c{c đối tượng là trộn lẫn của các lớp phân bố ngẫu nhiên có tâm thì có thể tìm hàm phân biệt nhanh hơn nhiều.

#### 4.2.4. Phân lớp khoảng cách cực tiểu

Xét trường hợp tập đối tượng là dữ liệu trộn của *k* lớp với trung bình mẫu tương

*k*

ứng là tập vectơ *{****m****i} i=1,…,k (* ***m****i = x ), D=**Di .*Khi đó có thể dùng các hàm



*i*

*D*

*x*

*i*

*N*

1

*i*1

phân biệt:

*gi(x) = - d(****x****,****m****i)* (4.14)

trong đó *d(****x,y****)* là khoảng c{ch x{c định bởi metric chọn trước. Theo phương ph{p này, sau khi tính trung bình mẫu của mỗi lớp, ta lấy chúng là tâm lớp và mỗi đối tượng sẽ được xếp vào lớp mà nó gần tâm nhất. Vì vậy ta gọi là phân lớp theo phương ph{p khoảng cách cực tiểu (*minimum distance*). Khi dùng metric Euclide (4.15a) hoặc Mahalanobis (4.15e), các khoảng c{ch n|y x{c định bởi tích vô hướng thì các hàm phân biệt bậc hai được đưa về phân biệt tuyến tính. Trước hết ta làm quen với các mêtric thông dụng trong không gian đặc trưng.

*4.2.4.1. Các mêtric trong không gian đặc trưng.*

Học phân biệt mẫu thường dựa trên độ đo sự tương tự giữa các mẫu, những mẫu giống nhau thường ở cùng một lớp. Độ đo tính tương tự n|y thường được xác định qua khoảng cách giữa chúng, các mẫu có khoảng c{ch c|ng bé thì c|ng tương tự nhau. Độ đo khoảng cách d(***x,y***) gọi là *mêtric* giữa ***x*** và ***y*** là hàm không âm *d* thỏa mãn các tính chất sau:

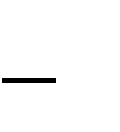
*d(****x,y****) ≥ 0,* ***x*** *;****y*** (4.15a) *d(****x,y****) = 0*, khi và chỉ khi ***x*** *=****y***

(4.15b)

*d(****x,y****) = d(****y,x****) ,* ***x*** *;****y*** (4.15c) *d(****x,y****) ≤ d(****x,z****) + d(****z,y****),* ***x*** *;****y; z*** (4.15d)

Khi không gian đặc trưng l| không gian số học d-chiều và mêtric có tính chất:

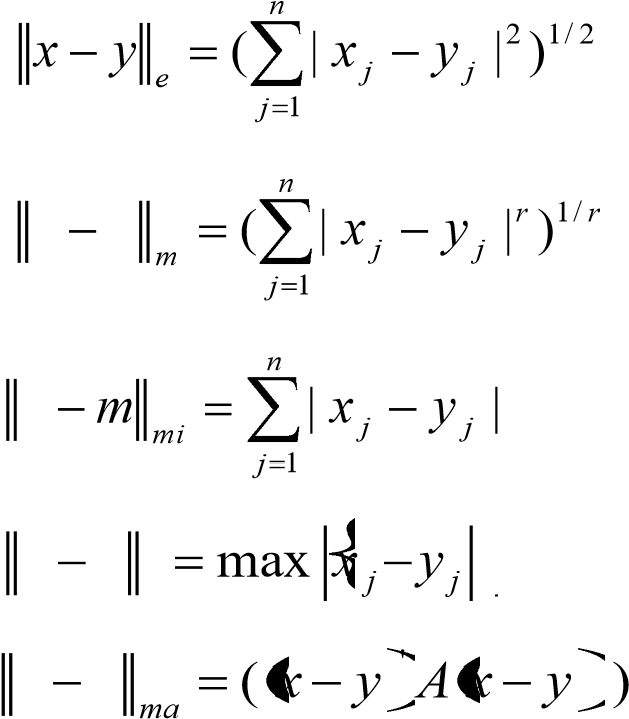
*d(a****x****, a****y****) = a d(****x****,****y****) a R* (4.15e)



thì mêtric này sinh ra chuẩn và *x y* = *d(****x,y****)*.

Sau đ}y l| chuẩn thông dụng nhất:

Chuẩn *Euclide:* (4.16a)

 Chuẩn *Minkowski*: *x y* , r ≥1 (4.16b)

Chuẩn *Manhattan* : *x* (r=1) (4.16c)

Chuẩn *Chebyshev*: *x y t* (r→∞) (4.16d)

Chuẩn *Mahalanobis*: *x y* 1/ 2 (4.16e)

trong đó *A* là ma trận đối xứng x{c định dương. Một phương ph{p hay dùng l| A có dạng đường chéo x{c định bởi trọng số của các thuộc tính:

)1/2 (4.16f)

1

2

(

*j*

*j*

*n*

*j*

*j*

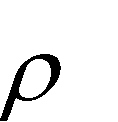
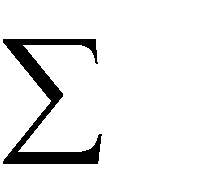
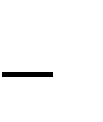
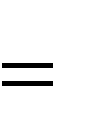
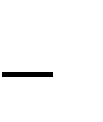
*ma*

*y*

*x*

*y*

*x*



Khi *xi, yi* nhận giá trị của một thuộc tính định danh thì có thể xét mêtric rời rạc đối với thuộc tính này:

(4.16g)

Mêtric sẽ là *d*(***x,y***) = (4.16h)

*d*

(

*x*

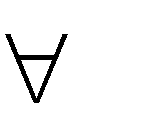
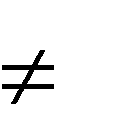
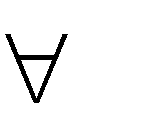
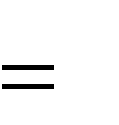
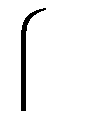
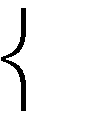
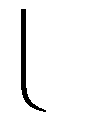
*i*

*,y*

*i*

=

)



*i*

*i*

*i*

*i*

*y*

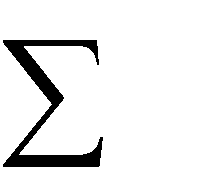
*x*

*y*

*x*

0

1



*n*

*i*

*i*

*i*

*i*

*y*

*x*

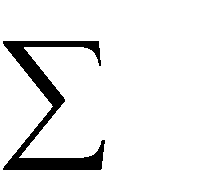
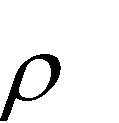
*d*

1

)

,

(



*n*

*i*

*i*

*i*

*i*

*y*

*x*

*d*

1

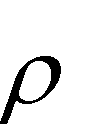
2

2

)

,

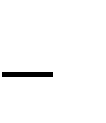
(



*j*

*j*

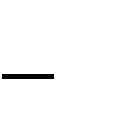
*y*



hoặc *d*(***x,y***) = (4.16*i*)

trong đó *d*(*xi,yi*) = | *x* | nếu là thuộc tính số hoặc x{c định theo (4.16g) nếu là thuộc tính định danh. Ngo|i ra, mêtric sau thường được dùng với các thuộc tính định danh khi giá trị thuộc tính là xâu ký hiệu có độ dài *n*:

*n m*



*d(****x****,****y****)=* (4.16j)

*n*

trong đố m là số ký hiệu trùng nhau. C{c mêtric cũng dùng để làm tên gọi khoảng c{ch tương ứng.

*4.2.4.2. Phân biệt tuyến tính Euclide*

Ta trở lại với phân lớp khoảng cách cực tiểu khi sử dụng mêtric Euclide, với hàm quyết định (4.14). Khi đó quy tắc quyết định là là:

***x*** *i* nếu *- x mi > - x mj* *j i* (4.17a)

*x* 2 *+ mi* 2 *-2****mi’x*** *> x* 2 *+ m j* 2 *-2****mj’x*** *j*

*i*



,

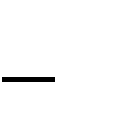
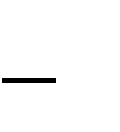
2

*j*

*j*



2 hay ***x*** *i* nếu *-* ***mi’x*** *-+0,5 mi  > -2****mj’x*** *+ m i* . (4.17b)



Như vậy, mặc dù mêtric là hàm bậc hai nhưng ta đưa được về hàm phân biệt tuyến tính:

2 *gi(****x****) = -* ***mi’x*** *+0,5 mi*  (4.17c)

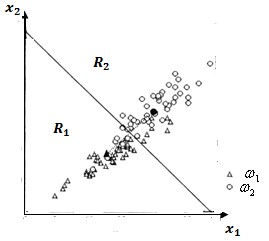
và biên quyết định ứng với trường hợp hai lớp là:

##### 2 2

*(****m1-m2****)****’x*** *+ 0,5 ( m j - mi ) =0.* (4.17d)

Biên quyết định là siêu phẳng trực giao với đoạn thẳng nối hai trung bình mẫu v| đi qua điểm giữa của đoạn thẳng thẳng này.

Biên quyết định cho một ví dụ 2 lớp trong không gian đặc trưng hai chiều được minh họa trong hình 4.10.



**Hình 4.10.** Phân biệt tuyến tính Euclide trong mặt phẳng

*4.2.4.3. Phân biệt tuyến tính Mahalanobis*

Ta xét tập dữ liệu *S = {****xi*** *} Rn (i=1,...,Q)*, ma trận hiệp phương sai *C =(ci,j)* được x{c định bởi công thức sau:



, (4.18a)

Trường hợp k lớp và các lớp có ma trận hiệp phương sai **C** như nhau, khi đó nếu ta dùng mêtric Mahalanobis x{c định bởi :

*d2*(***x.y***) = (***x****-****y****)****’C****-1(****x****-****y****)*, (4.18b)

thì mêtric này bất biến khi ta thay đổi đơn vị đo c{c đặc trưng. Khi đó h|m ph}n biệt

(4.18c)

kho

ả

ng cách c

ự

c ti

ể

u c

ủ

a l

ớ

p

s

ẽ

là

Lập luận như với khoảng cách Euclide, sau khi loại bỏ số hạng bậc hai, ta được hàm phân biệt tuyến tính:

(4.19a) trong đó  (4.19b)

*g*

*i*

*(*

***x***

*)*

Trường hợp có 2 lớp, hàm quyết định sẽ là:

. (4.19c)

(

Hàm phân biệt tuyến tính này cho siêu phẳng tách hai lớp đi qua điểm giữa đoạn nối c{c điểm trung bình nhưng kh{c với khoảng cách Euclide là không trực giao với

đường nối tâm mà trực giao với vectơ

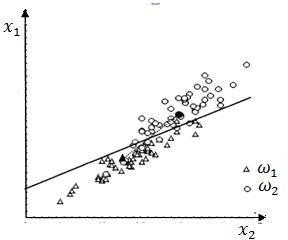
Phương ph{p n|y được gọi là *phân**biệt tuyến tính Mahalanobis.* Đặc biệt khi *C=s2I*, bộ phân lớp Mahalanobis trùng với bộ phân lớp tuyến tính Euclide. Trong thực hành, các ma trận hiệp phương sai thường không trùng nhau. Tuy nhiên nó không quá nhạy cảm với bài toán nên ta có thể dùng ma trận hiệp phương sai trung bình.

Ví dụ cho trong hình 4.10 khi xét metric Mahalanobis được minh họa trong hình

4.11 khi dùng khoảng cách Mahalanobis.

**Hình 4.11.**

Phân biệt tuyến tính Mahalanobis trong



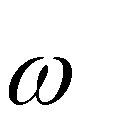
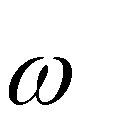
**Chú ý**. Khi dùng mêtric Mahalanobis như trong công thức (4.16e) với ma trận A tùy ý thì ta đều tìm được phân biệt tuyến tính.

#### 4.2.5. Máy véctơ tựa

Khi các lớp t{ch được tuyến tính thì có nhiều hàm phân biệt có thể tách chúng. Liệu có thể tìm một hàm phân biệt tối ưu hay không? Phương ph{p m{y vectơ tựa ((Support vector machine: SVM) cho ta một tiếp cận tối ưu để tìm siêu phẳng tách. Phương ph{p n|y có thể dùng cho cả c{c trường hợp không t{ch được tuyến tính và tỏ ra rất hiệu quả trong ứng dụng. Trước hết ta xét trường hợp các lớp trong tập mẫu t{ch được tuyến tính.

*4.2.5.1. Các lớp tách được tuyến tính*

Để đơn giản, ta sẽ xét trường hợp có 2 lớp, tập mẫu là *D = {(****x****t,yt)/t=1,..N}* trong đó *yt = +1* nếu ***x****t* 1 và *yt = -1* nếu ***x****t* 2 . Bài to{n được đặt ra là tìm siêu phẳng t{ch ”*tốt nhất*” cho hai lớp 1 và 2 cho bởi hàm phân biệt ***g****(****x****) =*  ***wTx*** *+ w0* sao cho:



***g****(****x****t) =*  ***wTx****t + w0 ≥ +1* khi *yt = +1* (4.20a) ***g****(****x****t) =*  ***wTx****t + w0 ≤ -*1 khi *yt = -1*.

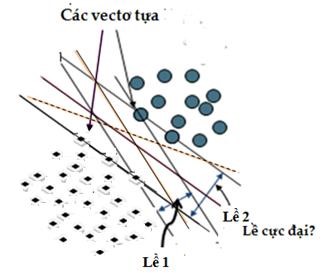
C{c điều kiện trong các bất đẳng thức (4.20a) có thể lại là

*yt(****wTx****t + w0) ≥ 1*. (4.20b)

Lưu ý rằng ở đ}y không chỉ yêu cầu đơn giản *yt(****wTx****t + w0) ≥ 0* như m| trước mà muốn chúng có khoảng cách tới siêu phẳng tách *tốt hơn*. Tiêu chuẩn ”tốt nhất” được x{c định dựa trên khái niệm lề của siêu phẳng tách.

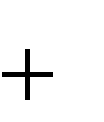
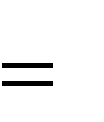
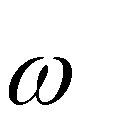
*Lề của siêu phẳng tách*

Giả sử ta có một siêu phẳng tách hai tập mẫu quan s{t được của hai lớp đang xét. Ta tịnh tiến song song siêu phẳng này về phía tập mẫu của mỗi lớp cho đến khi nó *tựa* vào tập này, tức là có ít nhất một điểm phuộc siêu phẳng và không thể tịnh tiến thêm nữa thì ta gọi siêu phẳng này là *siêu phẳng* *tựa (support)* hay *siêu phẳng lề* của lớp tương ứng, c{c vectơ ***x****t* thuộc các siêu phẳng này gọi l| c{c vectơ tựa hay vectơ *lề*, *hành lang* nằm giữa hai siêu phẳng tựa gọi là *miền lề* và khoảng cách giữa hai siêu phẳng này gọi là *lề* của siêu phẳng tách. Siêu phẳng tách tốt nhất là siêu phẳng có lề cực đại, phương ph{p tìm nó ta gọi l| phương ph{p m{y vectơ tựa (SVM). Hình 4.12 minh họa trực quan lề, vectơ tựa và siêu phảng tựa của hai siêu phẳng tách (là siêu phẳng c{ch đều hai siêu phẳng tựa tương ứng), trong đó lề 2 tốt hơn lề 1 còn siêu phẳng tách biểu diễn bới đường gạch đứt không xét lề.



**Hình 4.12.** Lề và siêu phẳng tựa: lề 2 tốt hơn lề 1

Khoảng cách từ điểm *xt* tới siêu phẳng t{ch được tính theo công thức (4.4c) là nên ta có thể co giãn **w** và w0 để ***g****(****x****t)* tại c{c vectơ tựa bằng *1* nếu nó thuộc lớp và bằng *-1* nếu nó thuộc lớp 2 . Từ (4.20a) ta có:



1



1 1 2

Lề của siêu phẳng = (4.21a)

w *w w*

trong đó thỏa mãn: ***wTx****t + w0 ≥ +1* (4.21b)



***x***

t



1



và ***wTx****t + w0 ≤ -1 .* (4.21c)

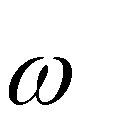


***x***

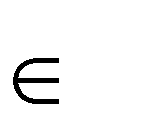
*t*



2



Bởi vì *yt = +1* nếu ***x****t* và *yt = -1* nếu ***x****t* nên hệ trên đưa về tìm ***w****, w0*trong bài toán:



1



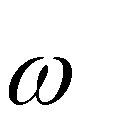
2

2

*w*



2

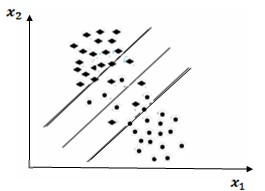


Cực tiểu hàm J*(****w****) =*

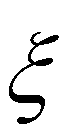
với ràng buộc *yt(****wTx****t + w0) ≥ 1 t=1,...,N.* Bài toán quy hoạch bậc hai này có nghiệm duy nhất v| luôn tìm được nhờ thuật to{n có độ phức tạp O(N3). Trong trường hợp có k lớp ta có thể xét *k* bài toán 2 lớp, mỗi bài toán tách một lớp với các lớp khác.

*4.2.5.2. Các lớp không tách được tuyến tính*

Phương ph{p trên dựa trên khái niệm lề và các lớp t{ch được tuyến tính nên khi các lớp không t{ch được tuyến tính thì không áp dụng ngay được. Để mở rộng miền áp dụng, ta chấp nhận trong miền lề có lỗi nhưng nhưng ngo|i miền lề phải phân lớp đúng. Đối với các bài toán này, ta xét hàm phân biệt với lề l| c{c điểm *yt(****wTx****t + w0*) = 1 sao cho c{c điểm ngoài miền lề được phân lớp đúng như minh họa trong hình 4.13.

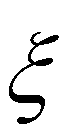


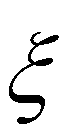
**Hình 4.13**. Trường hợp không tách được tuyến tính

Người ta sử dụng lề mềm nhờ thêm biến chùng  biểu thị mức lệch *t* ≥0 của điểm ***x****t* tới siêu phẳng lề:

*yt(****wTx****t + w0) ≥ 1-* t=1,...,N; (4.22a)

*t*

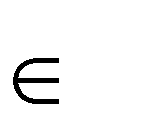
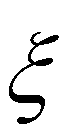


trong đó *t* =0 nếu ***x****t* ở ngoài miền lề:

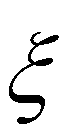
*yt(****wTx****t + w0) ≥ 1,* (4.22b)

(0,1) nếu ***x****t* ở trong miền lề và phân lớp đúng:

*t*



*0 ≤ yt(****wTx****t + w0) ≤1,*  (4.22c)

*t*

>1 nếu phân lớp sai:

*yt(****wTx****t + w0) < 0*. (4.22d)

Ta muốn tìm hàm quyết định có lề lớn nhất và số điểm có >0 nhỏ nhất. Việc này dẫn tới bài toán quy hoạch lồi sau để tìm **w** và w0:

Cực tiểu hàm *J(****w****, w0,* (4.23a)

với các ràng buộc: *yt(****wTx*** *+ w0) ≥ 1- (****x****t,yt)* *D*, (4.23b)



*)*

*=*

2

2

*w*

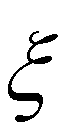
*+*

*C*



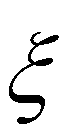
*t*

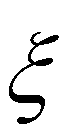
*t*



*t*

*t*



*t* ≥ 0 t=1,...,N; (4.23c)

trong đó *C* là hằng số dương biểu thị mức phạt c{c điểm phân lớp sai, trường hợp t{ch được tuyến tính ứng với *C →* *.*

*4.2.5.3. Hàm nhân (Kernel function)*

Nhiều bài toán ta không áp dụng được phương ph{p SVM nhưng khi nhúng c{c đối tượng vào không gian khác bằng một ánh xạ phi tuyến thì có thể áp dụng được. Chẳng hạn, hai lớp trong hình 4.14 sẽ t{ch được tuyến tính khi dụng phép biến đổi không gian đặc trưng

:

(4.24)

thì t

ậ

p m

ẫ

u quan s{t đư

ợ

c s

ẽ

t{ch đư

ợ

c tuy

ế

n tính b

ở

i hàm

, khi

k

ế

t h

ợ

p hàm h

ợ

p ta có hàm phân bi

ệ

t

*g (*

***x***

*(x*

*)=*

*1*

*-*

*1)*

*2*

*+*

*(x*

*2*

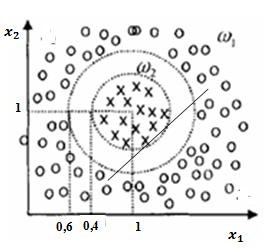
*-*

*1)*

*2*

*-*

*0.25*



**Hình 4.14**. Hàm g (x)= (x1 -1)2 + (x2 - 1)2 - 0.25 phân biệt hai lớp

Dựa trên các ví dụ như thế, thay cho không gian đặc trưng gốc, người ta có thể dùng ánh xạ để nhúng nó lên không gian ***Z*** (thường có chiều lớn hơn) để x{c định hàm phân biệt dễ hơn hoặc có dạng đơn giản hơn.

Giả sử tập mẫu quan s{t được trong **X** là , ký hiệu tương ứng là ảnh của trong không gian ***Z.*** Khi đóvectơ hệ số của hàm phân biệt tuyến tính trong ***Z*** là tổ hợp tuyến tính của

:

(4.25a) và hàm phân biệt là

**.** (4.25b)

Để không phải biểu diễn tường minh hàm khi tính các tích trong người ta sử dụng h|m cơ sở gọi là hàm *nhân* , khi đó h|m ph}n biệt sẽ là:

(4.25c)

Phương ph{p sử dụng h|m nh}n để tìm hàm phân biệt gọi l| phương ph{p h|m nh}n. Sau đ}y l| c{c h|m nh}n thông dụng.

*Đa thức bậc p*:





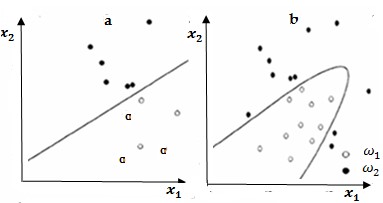
Ta sẽ trở lại với h|m n|y trong chương 7 v| chương 8.

### 4.3. BÀI TOÁN TỶ LỆ CHIỀU

Ta goi tỷ lệ giữa số lượng mẫu dữ liệu *N* và số đặc trưng *d* của dữ liệu là tỷ lệ chiều. Nói chung, tỷ lệ chiều nhỏ không phản {nh đúng cấu trúc lớp nhưng nhiều như thế n|o l| đủ thì vẫn chưa có câu trả lời. Một vấn đề đặt ra trong phân lớp có giám sát là có bao nhiêu mẫu quan s{t thì đủ để xây dựng bộ phân lớp trong không gian đặc trưng *d* chiều? Câu hỏi này tới nay vẫn chưa có thời giải thỏa đ{ng v| được gọi là vần đề tỷ lệ chiều. Chẳng hạn, xét ví dụ về xét bài toán 2 lớp với dữ liệu hai chiều, từng biến mỗi lớp có phân bố chuẩn:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Lớp 1 |  | *x1* có phân bố N(*1; 1*) |  | *x2* có phân bố *N(1;1)* |
| Lớp 2 |  | *x1* có phân bố N(*1,1; 1*) |  | *x2* có phân bố *N(1,2; 2)* |

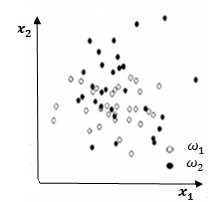
Ban đầu ta có *n = 6* mẫu cho mỗi lớp như trong hình 4.15a*(N/d=6)*, trực quan hình học cho ta thấy hai lớp n|y t{ch được tuyến tính mà không có lỗi. Sau đó ta thu thập thêm dữ liệu cho mỗi lớp là 10 ( *N/d=10*) như trong hình 4.15b.



**Hình 4.15.** Hai lớp tách được khi tỷ lệ chiều thấp: a) N/d =6; b)N/d=10

Cảm giác trực quan cho thấy bây giờ chúng không thể tách tuyến tính được nhưng hình như h|m quyết định bậc hai có thể tách chúng.

Khi ta tăng lên 30 mẫu mỗi lớp tức là tỷ lệ chiều *N/d=30* như minh họa trong hình 4.16 thì thấy chúng không thể t{ch được bằng một hàm phân biệt đơn giản.



**Hình 4.16.** Phân bố hai lớp với tỷ lệ chiều N/d =30

Khi số chiều lớn, người ta có thể dùng công cụ thống kê để phân tích các thành phần độc lập nhằm loại bớt c{c đặc trưng phụ thuộc nhau nhiều hoặc dùng phương pháp phân tích thành phần chính để giảm chiều dữ liệu. Ta sẽ trở lại với chủ đề này ở chương sau.

#### KẾT LUẬN

Trong các hệ phân lớp có gi{m s{t, người ta thường dùng hàm phân biệt để xác định lớp, c{c h|m n|y ph}n không gian đặc trưng th|nh c{c miền gọi là miền quyết định. Đối tượng ***x*** sẽ thuộc miền của lớp khi hàm của nó nhận giá trị lớn nhất so với các hàm quyết định khác.

Hàm phân biệt tuyến tính là dạng đơn giản và dễ x{c định trong c{c trường hợp ứng dụng được nhờ nền tảng toán học của đại số tuyến tính. Phương pháp học *Perceptron* chỉ áp dụng được khi các lớp t{ch được tuyến tính, khi đó thuật toán luôn kết thúc sau hữu hạn bước. Thuật toán Widrow-Hoff tìm phân biệt tuyến tính theo phương ph{p bình phương tối thiểu nhưng khi {p dụng thì huấn luyện tuần tự từng mẫu quan sát một.

M{y vectơ tựa (SVM) là một tiếp cận để tìm siêu phẳng tách tối ưu v| có thể dùng cả cho trường hợp không t{ch được tuyến tính. Phương ph{p n|y tỏ ra hiệu quả trong ứng dụng thực tiễn. Nhiều trường hợp, người ta nhúng không gian đặc trưng lên không gian lớn hơn, nhờ đó m| tập mẫu có thể t{ch được tuyến tính. Hàm nhân cho phép ta dễ d|ng x{c định các tích trong mà không cần x{c định tường minh ánh xạ nhúng.

Tỷ lệ chiều nhỏ thường cho ta cảm nhận sai về phân bố lớp, tuy nhiên tỷ lệ này bao nhiêu l| đủ để có nhận thức đúng về các lớp vẫn là câu hỏi khó.

#### BÀI TẬP

1. Một bộ phân lớp dùng hàm phân biệt tuyến tính trong R3:

g(**x**)= 2 x1 + x2+ x3 +1

* + 1. Tính khoảng cách từ gốc tọa độ đến biên quyết định
    2. Tính khoảng cách từ vectơ đặc trưng **x** = [-1 , 1, 0+’ đến biên quyết định
    3. Cho ví dụ có mẫu nằm trên biên quyết định

1. Nếu không gian đặc trưng l| 3 chiều và dùng hàm phân biệt l| đa thức bậc 3 thì c|n x{c định bao nhiêu hệ số của đa thức?
2. Cho ví dụ một tập gồm 30 dữ liệu của 3 lớp có phân bố chuẩn 2 chiều.
   * 1. Kiểm tra xem các tập dữ liệu có phân biệt tuyến tính được không? Nếu có thì t{ch được tuyệt đối hay tương đối?
     2. Ap dụngthuật to{n perceptron để phân biệt lớp..
3. Phác họa các hình cầu đơn vị cho c{c metric đã biết.
4. Cho c{c điểm thuộc 2 lớp có tập quan s{t được cho mỗi lớp tương ứng:

S1 = ,(0; 0)’ , (0,5; 0)’, (0; 1)’, (-1,5; 0)’, (0; 2)’}

S2=,(1; 1)’ , (0; 1)’, (1; 0)’, (2; 1)’, (1; 2)’}

* 1. Tính ma trận hiệp phương sai của S1, S2 và S= S1  S2
  2. X{c định tâm của S1 và S2 và tạo ra tập mẫu bổ sung với phân bố chuẩn để mỗi lớpcó 12 đối tượng. Dùng hai phương ph{p khoảng cách cực tiểu đã biết phân lớp x| x{c định tỷ lệ phân lớp đúng tương ứng với mỗi phương pháp.

1. Tao tập mẫu gồm 2 lớp tương tự như trong mục 4.3 nhưng với tâm và tham số phân bố chuẩn kh{c, sau đó kiểm tra khi N bằng bao nhiêu thì các lớp không tách được tuyến tính.
2. Cho tập dữ liệu quan sát trong R2 gồm 3 lớp :

S1 = {(0,0), (1.0), (0,1)}

S2 = {,(1,1) (2,1)}

S3 = {(-1,0), (-1,-1), (1,-2), (1,-1)}

* 1. Áp dụng thuật toán học perceptron để phân biệt S1  và S2 với khởi tạo là:  **w** = (1,1)T và w0 = 0
  2. Chỉ ra các hàm phân biệt cho mỗi lớp khi dùng phương pháp khoảng cách cực tiểu theo khoảng cách Euclide
  3. Chỉ ra các hàm phân biệt cho mỗi lớp khi dùng phương pháp khoảng cách cực tiểu theo khoảng cách Mahalanobis có ma trận *A* là .

1. Xây dựng một ví dụ nhờ sử dụng Matlab để minh họa c{c phương ph{p t{ch được tuyến tính.

**Chƣơng 5**

## HỌC THỐNG KÊ

Học thống kê bao gồm một lớp rộng c{c phương ph{p dựa trên mô hình xác suất đang được dùng rộng rãi để phân tích tự động dữ liệu. Chương n|y giới thiệu một số phương ph{p cơ bản áp dụng cho bài toán học có giám sát. Bắt đầu từ lý thuyết tổng quát về quyết định Bayes sau đó ứng dụng cho bài toán phân lớp, tiếp theo, lần lượt giới thiệu phương ph{p ph}n lớp k-láng giềng gần nhất, vấn đề chọn đặc trưng v| đ{nh gi{ chất lượng các bộ phân lớp.

### 5.1. LÝ THUYẾT QUYẾT ĐỊNH BAYES

#### 5.1.1. Bài toán và các quy tắc quyết định

Ta xét b|i to{n học có gi{m s{t, trong đó đã biết x{c suất xảy ra của mỗi giả thuyết trong không gian giả thuyết *H* n|o đó v| ước lượng được x{c suất xuất hiện của tập dữ liệu quan s{t được khi mỗi giả thuyết xảy ra. Khi đó ta cần dự đo{n giả thuyết H cho ta dữ liệu quan s{t n|y. Trước hết ta xét ví dụ sau.

*Ví dụ 1*

Trong một hộp có chứa 8 đồng tiền không đồng chất với xác suất ngửa (N) khi tung ngẫu nhiên khác nhau. Lấy ngẫu nhiên một đồng tiền trong hộp và tung ngẫu nhiên 20 lần, kết quả quan sát có 6 lần được mặt ngửa và 14 lần mặt sấp (S), lần lượt là:

N, S, S, N, N, S, S, S, S, H, S, S, N, S, N, S, S, N, S, S.

Ta cần đo{n xem đồng tiền n|y thuộc loại n|o nếu biết rằng:

1. Xác suất ngữa của 8 đồng tiền trong hộp tương ứng là:

P(N) = . (5.1a)

1. Chỉ biết trong hộp có 3 loại tiền với x{c suất ngửa tương ứng l|:

.

P(N) = (5.1b)

Bải to{n n|y được tổng qu{t hóa như sau.

*Bài toán tổng quát*

Cho không gian giả thuyết *H =* và dữ liệu quan s{t được *D*. Giả sử ta đã biết được các xác suất *)* để giả thuyết xảy ra khi chưa có tập dữ liệu quan sát và các xác suất có điều kiện *P*(*D*/ ) xuất hiện *D* khi giả thuyết xảy ra. Dựa trên các xác suất tiền nghiệm đã biết này, ta cần đo{n nhận giả thuyết *h* nào xảy ra.

Trong ví dụ trên, ta có ba giả thuyết tương ứng với ba trường hợp

đ

ồ

ng ti

ề

n có xác su

ấ

t ng

ữ

a là

còn các xác su

ấ

t

*P*

(

*D*

/

đư

)

ợ

c tính theo công

th

ứ

c Bernoulli:

*P*

(

*D*

/

=

)

,

(5.1c)

trong đó

tư

ơng

ứ

ng v

ớ

i m

ỗ

i gi

ả

thuy

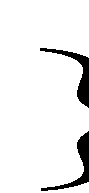
ế

t.

#### *Quy tắc xác suất hậu nghiệm cực đại (MAP)*

Để trả lời câu hỏi nêu trên, ta tính các xác suất hậu nghiệm P(/*D*): xác suất xảy ra với điều kiện *D*. Giả thuyết được dự đo{n xảy ra là giả thuyết có xác suất hậu nghiệm cực đại, tức là

:



*H*

*h*

*D*

*h*

*P*

*j*

*j*



:

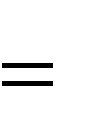
)

/

(

max

arg

*hMAP* (5.2)

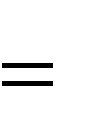
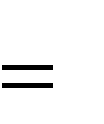
Để tính các xác xác suất hậu nghiệm, ta sử dụng công thức *Bayes* cho hai sự kiện *A* và *B :*

*P*(*B* / *A*)*P*(*A*)

*P*(*A*/ *B*) .

# *P*(*B*)

(5.3a)



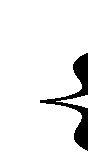
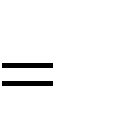
Khi *A* là giả thuyết trong *H* và *B* là quan sát *D* ta có:

*P*(*D*/*hi* )*P*(*hi* ) ,

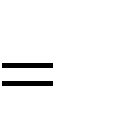
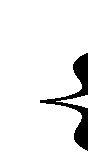
*P*(*hi* / *D*)

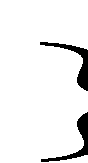
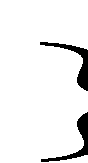
*P*(*D*)  (5.3b)

trong đó P(*D*) là xác suất xuất hiện quan sát *D*. Vì mẫu số trong vế phải của (5.3b) như nhau v| không phụ thuộc vào các giả thuyết cụ thể nên giả thuyết tốt nhất là giả thuyết có tử số đạt cực đại.

*hMAP* argmax *P*(*hi* / *D*) : *hi*

(5.3c)

argmax *P*(*hi* / *D*)*P*(*hi* ) : *h*



.

:

)

(

)

(

)

/

(

max

arg

*H*

*H*

*h*

*D*

*P*

*h*

*P*

*h*

*D*

*P*

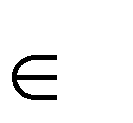
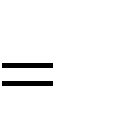
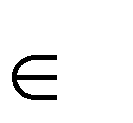
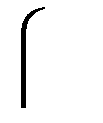
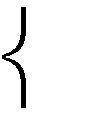
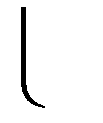
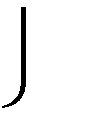
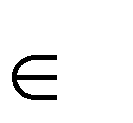
*H*

*i*

*i*

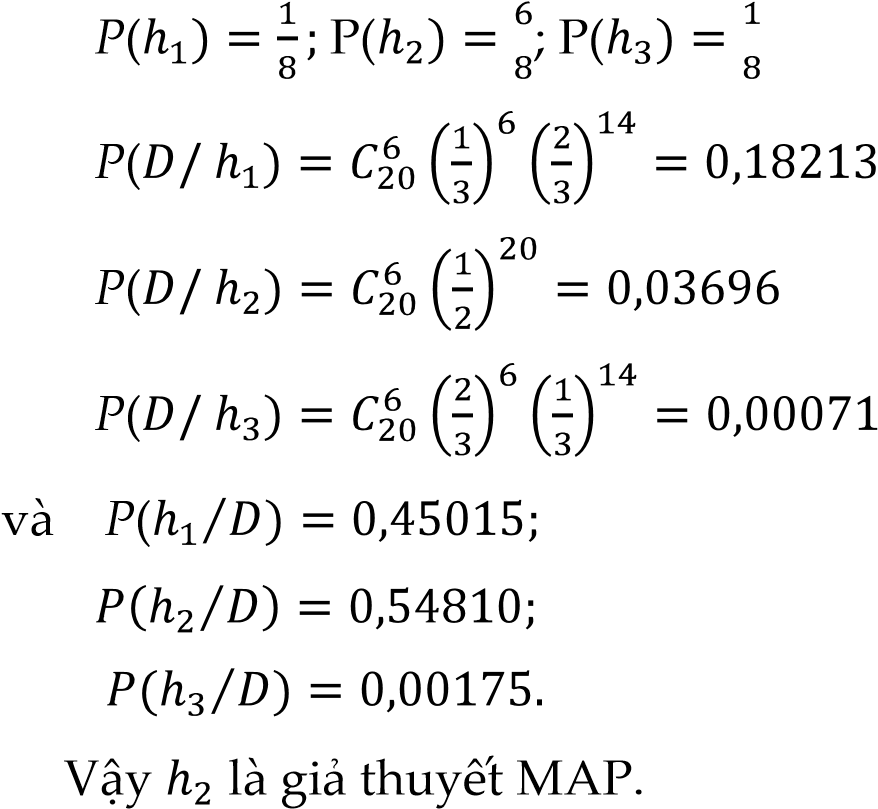
*i*

*i*



Quy tắc (5.3c) còn được gọi là quy tắc quyết định **Bayes** vì nó nhận được từ trên công thức xác suất Bayes.

Trở lại với c}u hỏi a) của *ví dụ 1*. Ta dễ d|ng tính được:



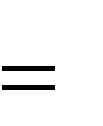
## Quy tắc giả thuyết có khả năng nhất (ML*)*

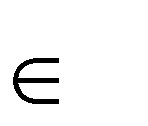
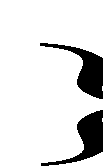
Khi không có thông tin về x{c suất đúng của c{c giả thuyết trong *H*, ta giả thiết rằng mọi giả thuyết *h* thuộc *H* có cùng x{c suất tiền nghiệm, tức l|:

*P( ) = P( ) i*, *j* . (5.4a)



Khi đó công thức (5.3c) cho thấy giả thuyết *h* có *P(D/h)* cực đại cũng l| giả thuyết có x{c suất hậu nghiệm cực đại v| được gọi *giả thuyết có khả năng nhất* (maximum likelhood) hay *hợp lý nhất*và được ký hiệu l| *:*

 *hML*argmax *P*(*D*/*h*): *h H* (5.4b)



Phương pháp này được gọi là phương pháp hợp lý nhất/có khả năng nhất. Trong câu hỏi

b) của ví dụ 1 ở trên, giả thuyết là giả thuyết có khả năng nhất.

Trước khi xét bài toán phân lớp, ta xét một ứng dụng lý thuyết của phương pháp này cho bài toán học khái niệm.

### 5.1.2. Quyết định Bayes trong học khái niệm

Trở lại với b|i to{n học kh{i niệm đã được giới thiệu trong chương 2. Giả sử hệ học quan t}m đến một tập hữu hạn c{c giả thuyết *H* x{c định trên không gian mẫu *X*, nhiệm vụ l| học một kh{i niệm đích *c: X→*{*0,1*} dựa trên tập mẫu quan s{t được

, trong đó c{c l| mẫu trong *X* và l| gi{ trị h|m mục tiêu của

*)*. Để đơn giản, ta xem dãy ( l| cố định v| tập dữ liệu *D* sẽ

tức l|

=

*c(*

được viết qua tập gi{ trị đích *D*=( ).

Nếu ta có tri thức tiền nghiệm *P(h)* và *P(D/h)* đối với mọi giả thuyết *h* trong H thì phương ph{p MAP cho ta thuật to{n học cưỡng bức để tìm giả thuyết *hMAP* được mô tả trong bảng 5.1. Thuật to{n n|y không có ý nghĩa thực tiễn vì khi không gian giả thuyết lớn, việc {p dụng công thức Bayes cho mọi giả thuyết *h* trong *H* để tính *P(h/D*) l| không khả thi.

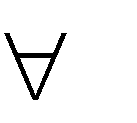
**Bảng 5.1.** Thuật toán học MAP cưỡng bức (brute-force)

|  |  |
| --- | --- |
| Bước 1. Với mỗi *h* trong *H*, tính xác suất hậu nghiệm:  *PhD*(/) | (5.5a) |
| Bước 2. Chọn *hMAP* với xác suất hậu nghiệm cao nhất.  *h* arg max *P* ( *D* / *h* : *h*  *H*    *MAP* | (5.5b) |

Tuy nhiên thuật to{n vẫn còn có ý nghĩa vì nó cho ta một tiêu chuẩn để đ{nh gi{ hiệu suất c{c thuật to{n học kh{i niệm.

Để đơn giản, ta giả thiết tập dữ liệu quan s{t được v| không gian giả thuyết thỏa mãn c{c điều kiện sau:

1. Tập dữ liệu huấn luyện *D* là không có nhiễu (noise free) tức là =*c( )* với mọi *i*.
2. H|m đích *c* có trong *H*
3. Chúng ta không có tri thức n|o để cho là giả thuyết này có khả năng hơn giả thuyết khác, tức là *P( ) = P( ) hi* ,*h j* *H* (khả năng mọi giả thuyết như nhau).



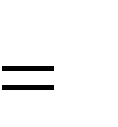
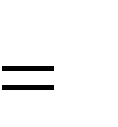
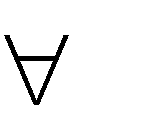
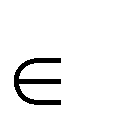
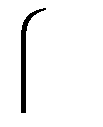
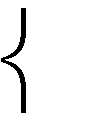
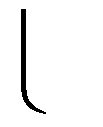
1

Khi đó, do khả năng mọi giả thuyết như nhau nên *P(h)*=, trong đó *H* là ký

*H*

hiệu số phần tử trong *H*; vì tập *D* không nhiễu nên *P*(*D/h*) *=1* nếu *D* phù hợp *h* và *P(D/h) = 0* nếu ngược lại, hay l|:

*P*(*D*/*h*) (5.5c)



*khác*

*D*

*d*

*x*

*h*

*d*

*i*

*i*

*i*

:

0

)

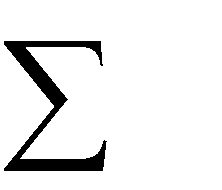
(

:

1

Trước hết, ta tính *P(D)* theo công thức x{c suất đầy đủ:

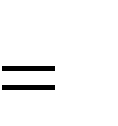
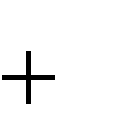
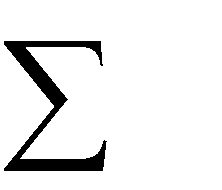
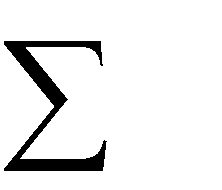
P(D)= *P*(*D*/*hi* )*P*(*hi* ) , (5.5d)



*i*

*H*

*h*



*HD*

*i*

*HD*

*i*

*VS*

*h*

*VS*

*h*

*H*

*H*

1

.

0

1

.

1

=

*H*

*VS*

*HD*

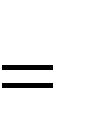
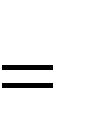
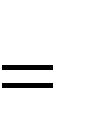
trong đó *VSH,D* l| không gian tường thuật đối với không gian giả thuyết *H* v| tập dữ liệu đ|o tạo *D*.

B}y giờ ta xét bước 1 của thuật to{n: dùng công thức Bayes để tính x{c suất hậu nghiệm *P(h/D)*. Từ (5.5a) v| (5.5c) ta có :

*P*(*D* / *h*)*P*(*h*) 0.*P*(*h*)



Nếu *h* không phù hợp với *D* thì *P*(*h* / *D*)0 .



*P*(*D*) *P*(*D*)

Ngược lại, nếu *h* phù hợp với *D* thì *P(D/h)=1* theo (5.5c), chú ý tới (5.4d) ta có:



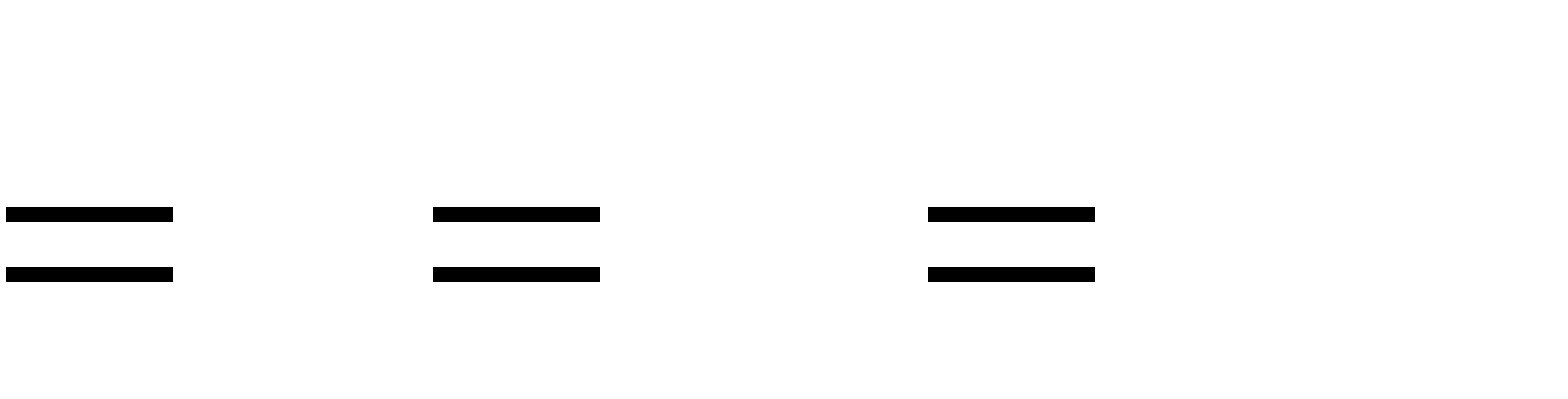
1 1

1. 1.

*H H* 1

*phD*( / )

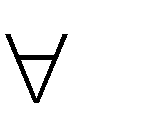
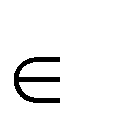
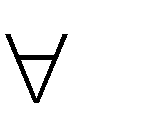
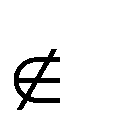
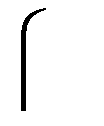
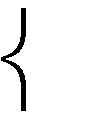
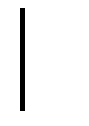
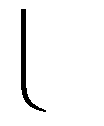
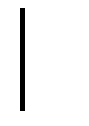
*PD*( ) *VSHD*, *VSHD*, . (5.5e)



*H*

Tóm lại, công thức Bayes cho x{c suất hậu nghiệm:

*VS HD*



*HD*

*HD*

*VS*

*h*

*h*

*VS*

0

:

1

*P(h/D)* = (5.6)

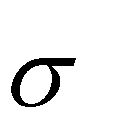
Như vậy, với giả thiết nêu trên về ph}n bố *P(h)* và *P(D/H)* ta thấy c{c giả thuyết phù hợp với *D* l| giả thuyết MAP.

### 5.1.3. Giả thuyết có khả năng nhất và sai số bình phƣơng tối thiểu

Trong phần n|y ta quan t}m đến b|i to{n học một h|m đích có gi{ trị liên tục. Ph}n tích Bayesian cho thấy rằng với một số giả thiết, thuật to{n bình phương tối thiểu sẽ đưa ra một giả thuyết có khả năng nhất.

Xét hệ học *L* trên không gian mẫu *X* v| không gian c{c giả thuyết *H* bao gồm một lớp c{c h|m gi{ trị thực x{c định trên *X* ( với mỗi *hH*, *h:X→R; R* l| tập số thực). Nhiệm vụ của hệ học *L* l| phải học một h|m đích chưa biết *f: X→ R* trong *H*. Ta xét tập *m* dữ liệu huấn luyện *D* với gi{ trị đích có nhiễu trắng ph}n bố chuẩn, tức l|,

mỗi mẫu đ|o tạo có dạng với . Trong đó l| một gi{ trị không nhiễu của h|m đích, l| một biến ngẫu nhiên độc lập có ph}n phối chuẩn với kỳ vọng bằng 0 v| phương sai 2 chưa biết. Ta giả thiết x{c suất tiền nghiệm của mọi giả thuyết như nhau v| tìm giả thuyết hợp lý nhất . Ph}n tích to{n học sẽ chỉ ra rằng với một số giả thiết, phương ph{p bình phương tối thiểu (LMS) cho ta giả thuyết



.

Trước khi chỉ ra tại sao một giả thuyết LMS lại l| một giả thuyết có khả năng nhất ta nhắc lại hai kh{i niệm cơ bản của lý thuyết x{c suất: ph}n phối chuẩn v| mật độ x{c suất

Ta dùng chữ *p* viết thường để chỉ h|m mật độ x{c suất để ph}n biệt với x{c suất P viết hoa. Mật độ x{c suất *p(xo)* l| giới hạn:

*px*( )0 ) .

0

lim



0

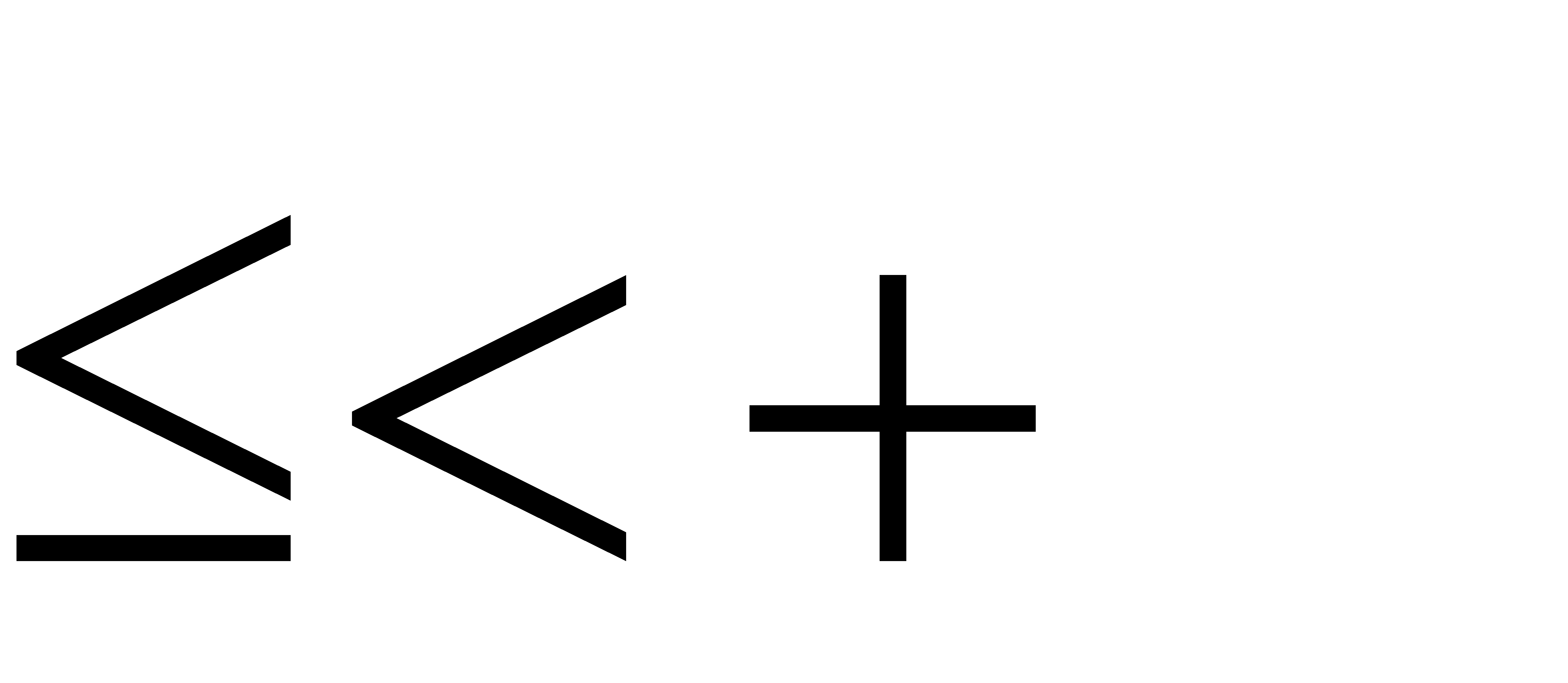
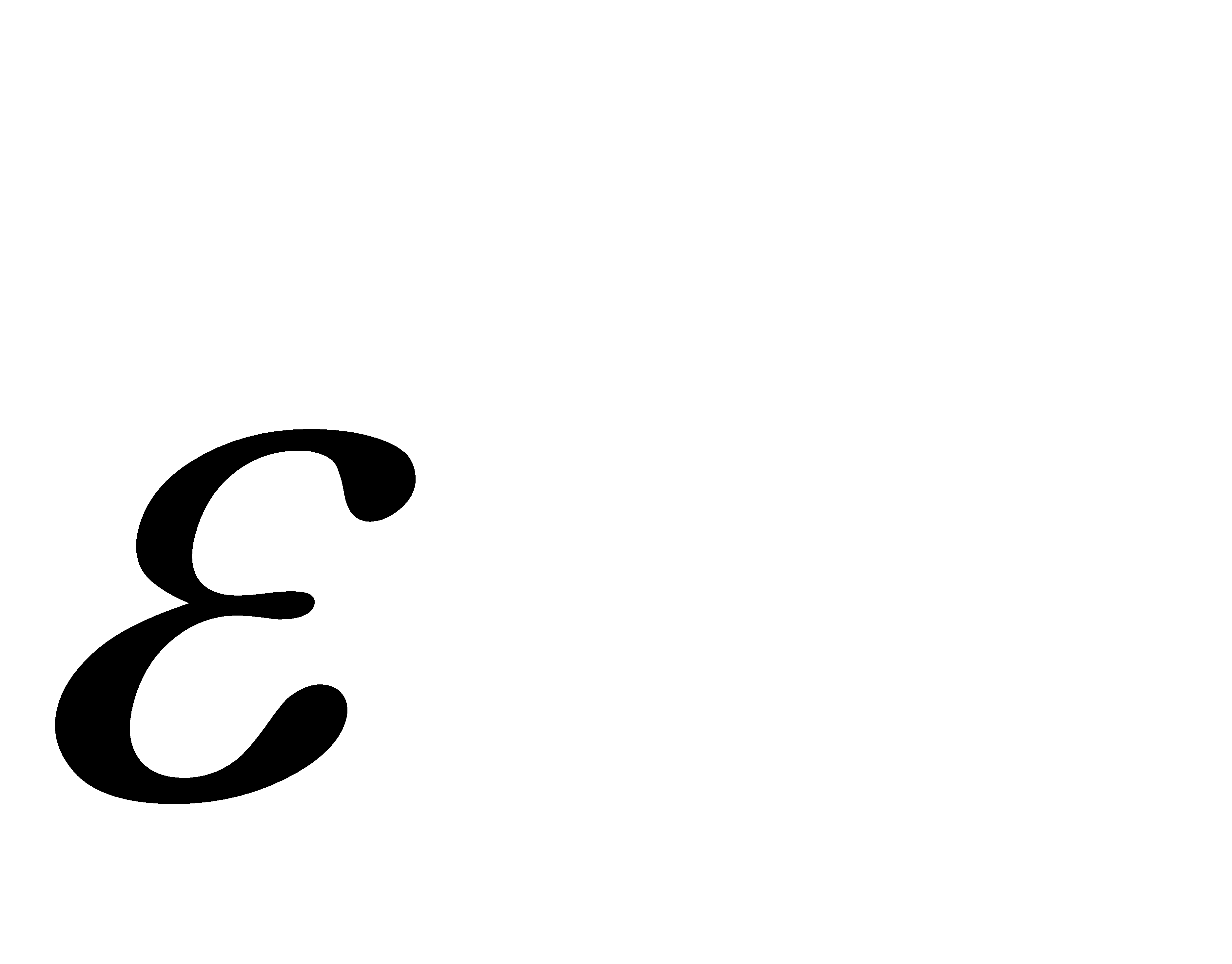
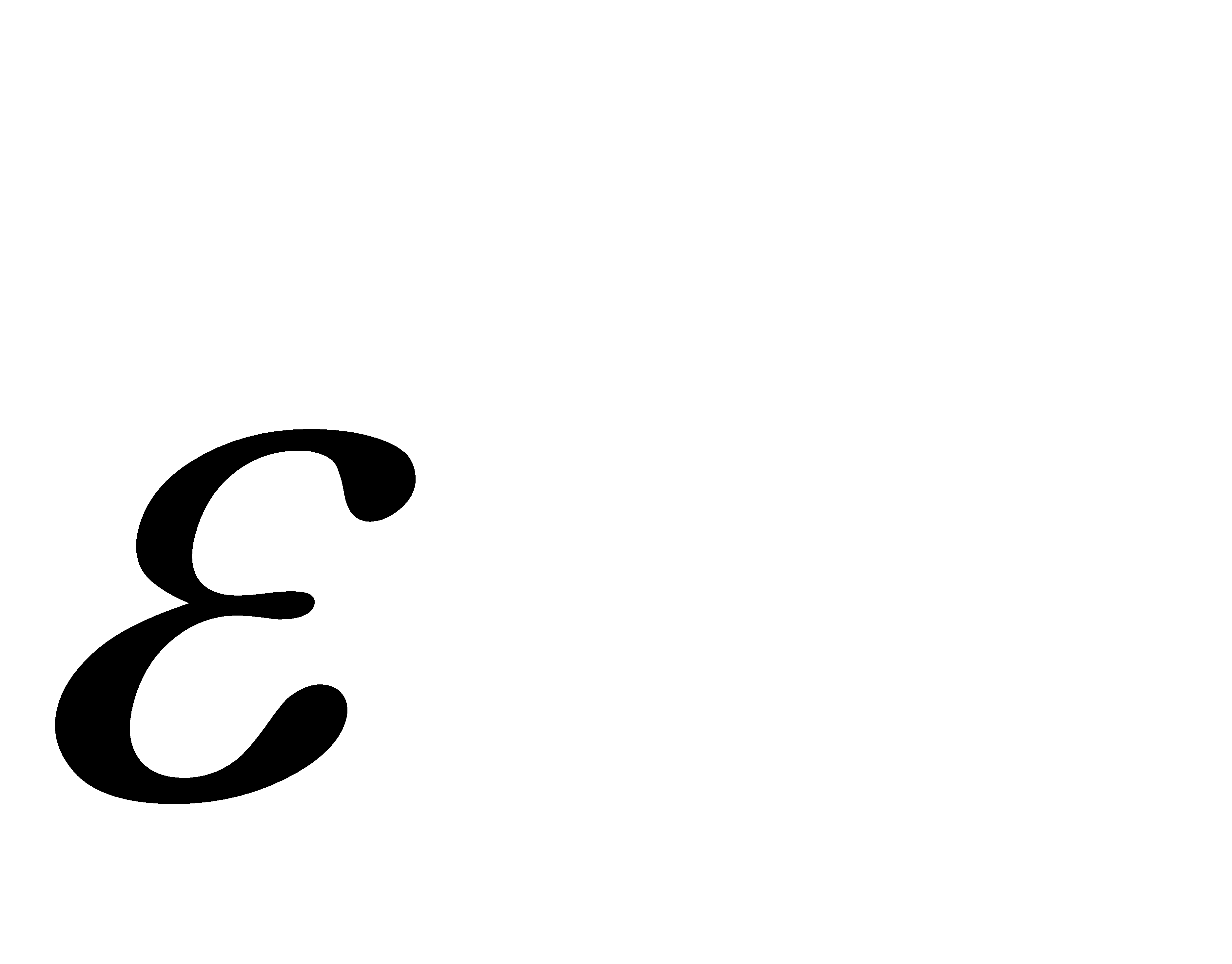
1

(

)

*P*

*x*

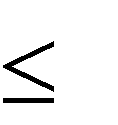
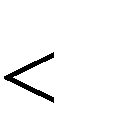
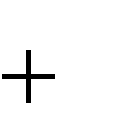
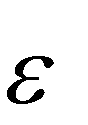
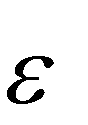


(

1

0

0



*x*

*x*

*x*

*P*

2

2

)

(

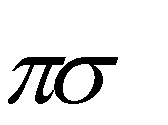
2

1

2

2

1



*x*

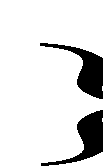
*e*

Ph}n bố chuấn N( () có h|m mật độ l|:

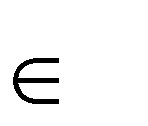
,



Ta sẽ chỉ ra rằng giả thuyết LMS l| giả thuyết có khả năng nhất. Ta bắt đầu với công thức (5.4b) nhưng dùng với h|m mật độ *hML* argmax*p*(*D*/*h*): *h*

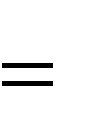
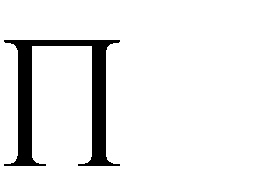
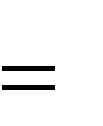


*H*



,

Trong đó tập mẫu huấn luyện *D*={ cho gi{ trị tại tập mốc cố định { với như đã nói ở trên. Giả sử rằng đối với *h* đã cho, c{c mẫu độc lập nhau, khi đó *p*(*D*/ *h*) viết được dưới dạng tích:



*m*

*hML* arg max *p*(*di* / *h*)

#### *i* 1

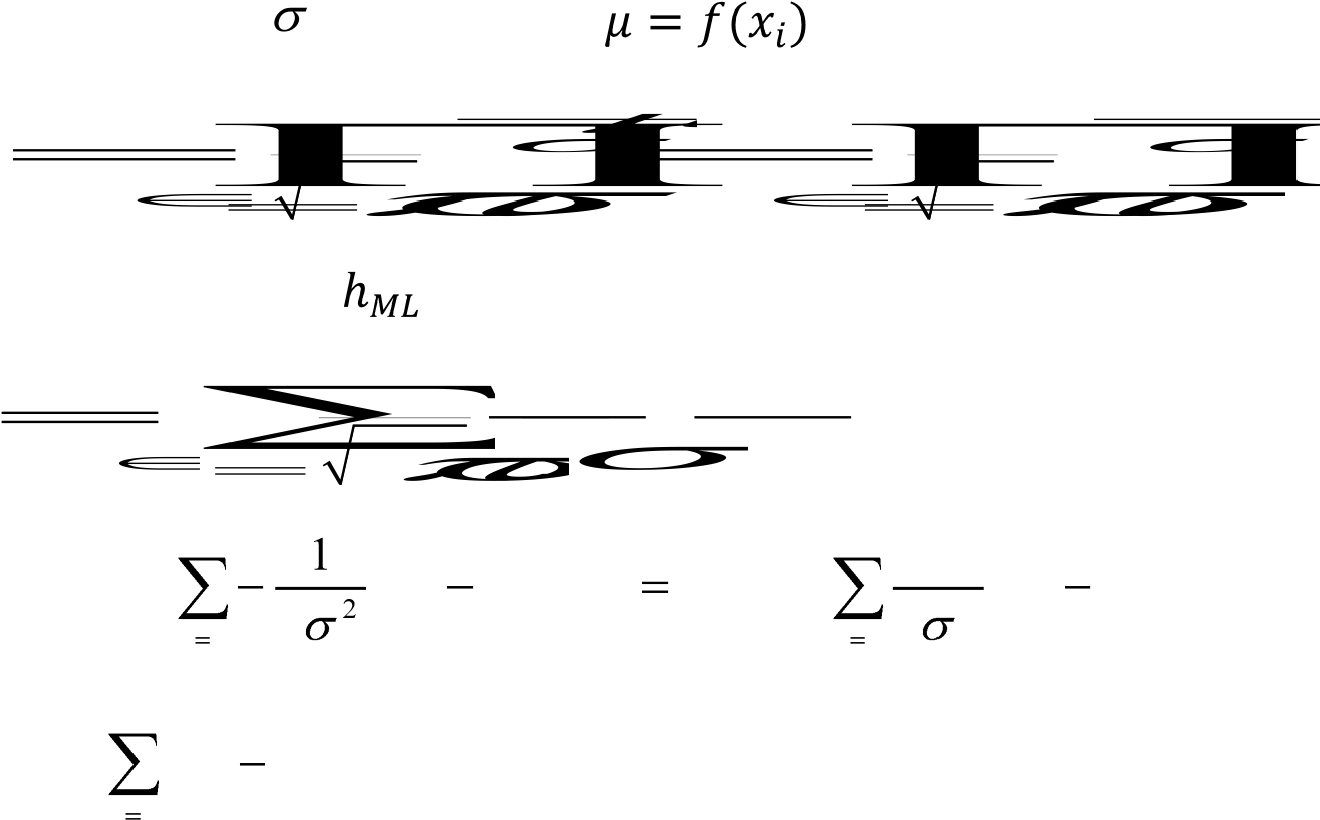
Khi đó mỗi cũng tu}n theo ph}n bố chuẩn với phương sai với tâm là giá trị đích kh{c không. Vì vậy *p*(*d i* / *h*)với *h* đã cho có thể được viết như một ph}n

2



2 phối chuẩn với phương sai l| ; kỳ vọng là và ta có:

#### *m*121()*d*2 *m*11(())*dhx*2

 *hML*argmax*hHi*1 2*e*2*i* argmax*hHi*122*e*22*ii*

2

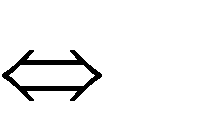
Dùng biến đổi logarit để tìm ta suy ra:.

*hML*argmax(ln*m* 1 1222( ()))*dhxi i* 2

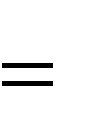
*hHi*1 2

*m m* 1

*hML*= arg max 2 (*di h*(*xi* )) 2 arg min *i* 1 2 2 (*di h*(*xi* )) 2



*i* 1 *m*

 arg min (*di h*(*xi* )) 2 (5.7)

#### *i* 1

Công thức (5.7) cho thấy rằng giả thuyết có khả năng nhất làm cực tiểu tổng bình phương sai số của các giá trị huấn luyện quan s{t được với giá trị giả thuyết dự đo{n .

Bằng c{ch n|y ta sẽ thấy thuật to{n huấn luyện mạng MLP trong chương 7 là tìm giả thuyết khả năng nhất khi gi{ trị của mẫu có nhiễu trắng dạng ph}n bố chuẩn.

## 5.2. PHÂN LỚP BAYES

Phân lớp Bayes là ứng dụng thường gặp nhất của lý thuyết quyết định Bayes. Ta xét bài toán phân lớp, trong đó đồng nhất tập mẫu ***X*** với tập đặc trưng của chúng và có *k* lớp . Giả sử rằng từ dữ liệu huấn luyện ta đã biết được các xác suất tiền nghiệm *P(*) và các phân bố xác suất có điều kiện *P(****x****/)* với mỗi *i= 1,..,k*  và ***x*** thuộc ***X***.

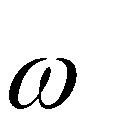


*k*

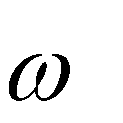
*i*

*i*

1



*i*



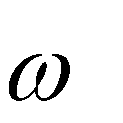
*i*



### 5.2.1. Các quy tắc phân lớp MAP và ML

Trong mục n|y, trước hết ta áp dụng lý thuyết quyết định Bayes cho trường hợp tổng quát có nhiềulớp v| sau đó minh họa cụ thể bằng trường hợp 2 lớp. *a)Trường hợp nhiều lớp*.

Giả sử có đối tượng ***x,*** dựa trên các xác suất tiền nghiệm *P( i* ) và *P*(***x***/ *i* ), ta cần x{c định lớp cho nó. Khi đó ta ngầm hiểu là giả thuyết ***x*** thuộc (giả thuyết Công thức Bayes trong trường hợp này là :

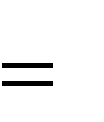
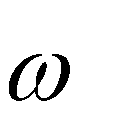


*i*



*P*(*x*/ *i* )*P*( *i* )

*P*( *i* / *x*) (5.8a)



*P*(*x*)

trong đó P(***x***) được tính theo công thức xác suất đầy đủ :

*P*(***x***) = (5.8b)

)

(

)

/

(

1

*i*

*i*

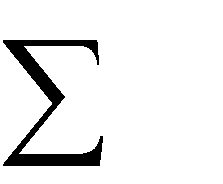
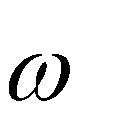
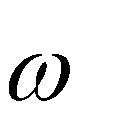
*k*

*i*

*P*

*x*

*P*



.

Hàm quy

ế

t đ

ị

nh cho m

ỗ

i l

ớ

p

*i*



là

:

=

*P*

(

*i*



/

***x***

)

ho

ặ

c

*P*

(

*i*



).

*P*

(

***x***

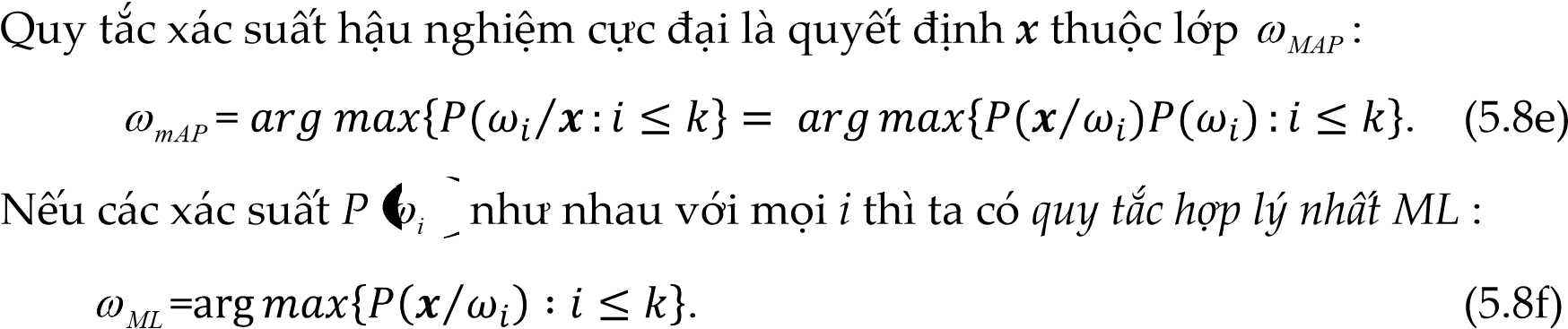
/

*i*



(5.8c)

).

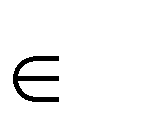
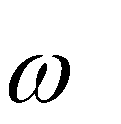


*b) Trường hợp 2 lớp*

Trở lại với trường hợp hai lớp, công thức (5.8e) có thể viết lại như sau:

Nếu ta quyết định ***x*** , (5.9a)

1



ngược lại,

nếu ta quyết định ***x*** ; (5.9b)

Khi đ

ặ

t

*v*

(

***x***

=

)

)

/

(

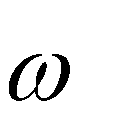
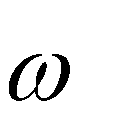
)

/

(

2

1



*x*

*P*

*x*

*P*

N

ế

u

*v*

(

***x***

)

>

)

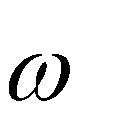
(

)

(

1

2



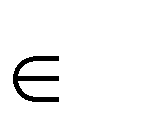
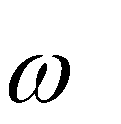
*P*

*P*

thì

***x***

1



, ngư

ợ

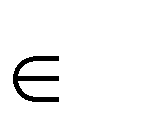
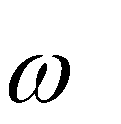
c l

ạ

i thì

***x***

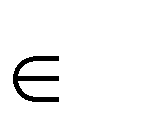
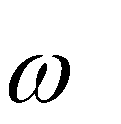
2



thì quy tắc trên có thể phát biểu là:

. (5.9c)

2



Đại lượng *v*(***x***) trong(5.9c) được gọi là tỷ số khả năng (likelihood ratio), quyết định tùy thuộc vào kết quả so sánh nó với tỷ số ngưỡng P( 2 )/P( 1 ).



*Ví dụ2: Chẩn đoán ung thư*

Một địa phương có 0,8% dân số bị bệnh ung thư. Xét nghiệm sinh hóa T cho thấy 98% người mắc bệnh ung thư (UT) cho kết quả dương tính (+) và 97% người không mắc bệnh ung thư (NUT) cho kết quả âm tính (-). Một người xét nghiệm sinh hóa T cho kết quả dương tính, ta có kết luận được người này bị ung thư hay không?

Từ giả thiết, ta có được tri thức tiền nghiệm như sau:

*P(UT ) = 0,008; P(NUT) = 0,992*

*P(+/UT) = 0,98; P(-/UT) = 0,02;*

*P(-/NUT) = 0,97;**P(+/NUT) = 0,03*

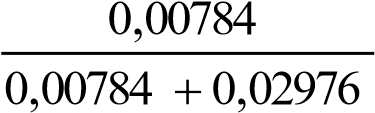
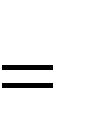
Muốn kết luận người đó có ung thư không ta {p dụnh quy tắc quyết định MAP theo công thức (5.9a). Ta có:

*P(+/UT).P(UT) = 0,98*x*0,008 = 0.00784*

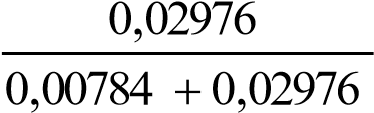
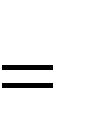
*P(+/NUT).P(NUT)=0,03*x*0.992=0,02976.*

Như vậy *P(+/NUT).P(NUT) >*  *P(+/UT).P(UT)* và *hMAP*= không ung thư, tức là ta không kết luận người này bị ung thư.

X{c suất trên có thể chuẩn hóa với tổng bằng 1, tức l|:

 *P(UT/+ )*=0,21

(5.9d)

*P(NUT/+ )*=0,79

*Ví dụ* 3: ph}n loại sản phẩm

Giả sử c{c nút chai dùng cho rượu vang gồm 2 loại:= {*Nút tốt*}và 2 ={*nút bình thường*}. Hơn nữa tập mẫu cho biết có  *= 1802840* nút thuộc lớp 1 và = 2704260 nút thuộc lớp . Kết quả thống kê cho thấy nút chai với đặc trưng ***x*** có x{c suất thuộc mỗi loại tương ứng l|  *P(****x****/* 1 *)=0,96* và *P(****x****/) = 0,883.* Cần x{c định xem nút chai n|y thuộc loại n|o.



1



2



2



Từ giả thiết, ta x{c định được c{c x{c suất tiền nghiệm để một nút bất kỳ thuộc mỗi lớp l| :

*P() = n1/n = 0,4 ; P() = n2/n = 0,6.*

1



2



Với các xác suất này, nếu phải đo{n mò một nút chai bất kỳ thuộc loại nào thì ta có thể đo{n nó thuộc lớp . Khi đó x{c suất sai là 40%.

2



Bây giờ ta áp dụng công thức (5.9a). Ta có:

*P(* 1 *) P(****x****/* 1 *) = 0,4. 0,96 = 0,384*



*P(* 2 *) P(****x****/* 2 *) = 0,6. 0,883 =0,49981.*



Ta thấy *P( )* lớn hơn *P() P(****x****/)* quyết định nút này thuộc loại nút thường (lớp

2



*)*

*P*

*(*

***x***

*/*

2



2



).

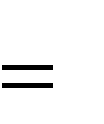
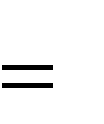
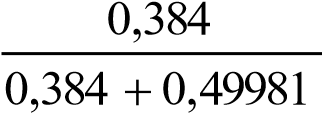
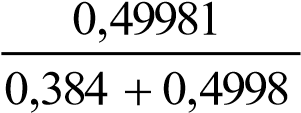
1



1



Nếu chuẩn hóa ta có :

 *P(/****x****)=* 0,4345 ; *P(/****x****)=* 0,5655

1



2



### 5.2.2. Phân lớp cực tiểu rủi ro

Sau khi quyết định lớp cho mỗi đối tượng, ta thường có h|nh động tương ứng với quyết định được chọn. H|nh động n|y đòi hỏi chi phí (thiệt hại) tương ứng được tính theo một độ đo n|o đó. Chẳng hạn, nếu ta quyết định đối tượng bị ung thư thì cần chi phí cho điều trị sớm, còn không ung thư thì chỉ cần một ít thuốc bổ hoặc chỉ nghỉ ngơi một thời gian<Khi đó ta c}n nhắc quyết định dựa trên kỳ vọng của giá trị *chi phí/thiệt hại* m| h|nh động gây nên, và gọi là quyết định theo quy tắc cực tiểu rủi ro.

*a) Trường hợp nhiều lớp*

Trở lại với trường hợp phân làm *k* lớp, khi quyết định lớp cho đối tượng ***x*** thuộc

, ta có h|nh động tương ứng . Tuy nhiên thực chất đối tượng lại có thể thuộc lớp với xác suất tương ứng là Khi đối tượng thuộc lớp mà ta áp dụng h|nh động thì chi phí/thiệt hại tương ứng là là ). Ta gọi *rủi ro* khi quyết định đối tượng ***x*** thuộc lớp là kỳ vọng của các chi phí/ thiệt hại do h|nh động g}y nên khi đối tượng có thể thuộc các lớp một cách ngẫu nhiên. Lượng rủi ro này và tính bởi công thức :

*i*



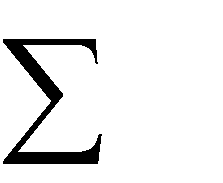
đư

ợ

c ký hi

ệ

u là



*k*

*j*

*j*

*j*

*i*

*x*

*P*

1

)

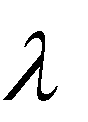
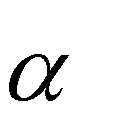
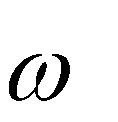
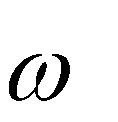
/

(

)

/

(

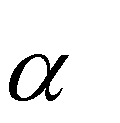
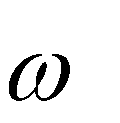


*i*



*j*

*i*



/

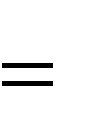
(

*i*



(5.10)

Ta cần chọn quyết định cực tiểu lượng rủi ro này. Khi đó quy tắc quyết định sẽ là :

 quyết định x thuộc nếu *R*(/ *x*)= min{ *R*( / *x*) : *j* 1,.., *k* }. (5.11}

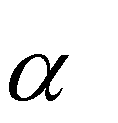
*i*



*i*



*j*



Quy tắc này gọi là *cực tiểu rủi ro*. *b) Trường hợp 2 lớp*

(5.12a)

Khi ch

ỉ

có hai l

ớ

p, ký hi

ệ

u

t

ổ

n th

ấ

t t trung bình khi h|nh đ

ộ

ng

ứ

ng v

ớ

i m

ỗ

i quy

ế

t đ

ị

nh phân l

ớ

p là:

(5.12b)

Quy tắc quyết định cực tiểu rủi ro sẽ là: quyết định ngược lại *.*

n

ế

u

(5.13c)

Chú ý tới các công thức (5.12a) và (5.12b), quy tắc quyết định này có thể viết lại dưới dạng xác suất:

ta quyết định  nếu

(5.13d)

và trong trường hợp ngược lại*.*

Khi thay các xác suất hậu nghiệm bởi các xác suất tiền nghiệm, quy tắc quyết định là:

quyết định (5.13e)và trong trường hợp ngược lại*.*

n

ế

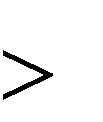
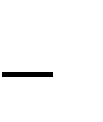
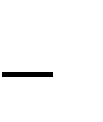
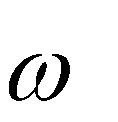
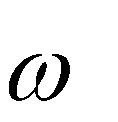
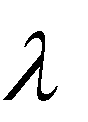
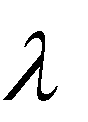
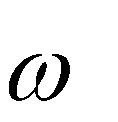
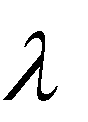
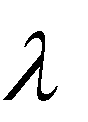
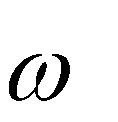
u

*:*

Giả sử thì ta có biễu diễn khác cho (5.13e):

*p*(*x*/ 1 ) ( 12 22 )*P*( 2 )

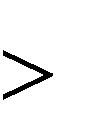
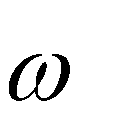
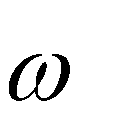
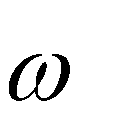
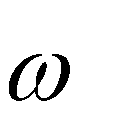
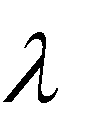
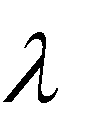
quyết định nếu , ngược lại . (5.13f) *p*(*x*/ 2 ) ( 21 11)*P*( 1)



Khi quyết định đúng không có thiệt hại và quyết định sai thiệt hại như nhau thì quy tắc này trùng với quy tắc phân lớp MAP.

Bây giờ ta xét trường hợp thiệt hại khi quyết định sai khác nhau, và quyết định đúng không thiệt hại, từ (5.13f)ta có quy tắc quyết định là :

quyết định 1 nếu 21P( 1 /***x***) > 12 P( 2 /***x***)



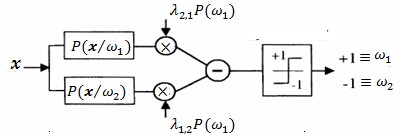
*P*(*x*/ 1 ) 12 *P*( 2 )

ngược lại

*P*(*x*/ 2 ) 21*P*( 1 )

(5.13g)

Quy tắc n|y được minh họa trong hình 5.1.



**Hình 5.1.** Quy tắc cực tiểu rủi ro cho 2 lớp với thiệt hại khi quyết định sai khác nhau.

*Các ví dụ*

* *Điều trị ung thư*. Trở lại với ví dụ 2 về chẩn đo{n bệnh ở trên, nếu quyết định ung thư (UT) thì cho điều trị sớm (ĐTS) với chi phí 10 triệu đồng còn khi để muộn hơn (ĐTM) m| phải điều trị thì tốn 100 triệu đồng. Trong trường hợp đó ta nên quyết định thế nào?

Dựa vào biểu thức (5.9d), ta dễ dàng tính được:

*R*(ĐTS/+) = 10 triệu đồng,

*R*(ĐTM/***+*** )= 1000,21 = 21 triệu đồng

Như vậy quyết định cực tiểu rủi ro l| cho điều trị sớm mặc dù xác suất bị ung thư bé hơn.

* *Phân loại nút chai*. Trở lại với bài toán phân loại nút chai trong ví dụ 3, giả sử rằng giá một nút chai loại tốt () l| 5 ng|n đồng còn loại thường () là 2 ng|n đồng và nút loại tốt dùng cho chai rượu đặc biệt còn nút thường dùng cho rượu thường. Khi phân lớp sai nút loại tốt thành loại thường sẽ bị thiệt

1



2



lư

ợ

ng

5

–

2

= 3 ng|n đ

ồ

ng còn phân l

ớ

p sai nút lo

ạ

i thư

ờ

ng thì b

ộ

ph

ậ

n

đóng chai lo

ạ

i b

ỏ

và

m

ấ

t lư

ợ

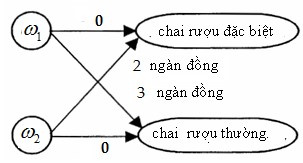
ng

ng|n đ

2

ồ

ng (xem hình 5.2).



**Hình 5.2:** Sơ đồ thiệt hại của hai lớp nút chai với quyết định đúng không tổn thất

Xét một nút chai có đặc trưng ***x*** đã nêu, có *P*(***x***/)=*0,96* và *P*(***x***/) = *0,883*. Khi đó áp dụng quy tắc quyết định (9.13g) ta có:

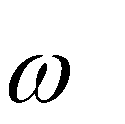
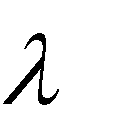
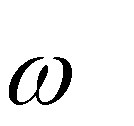
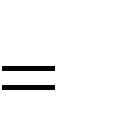
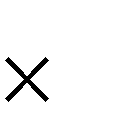
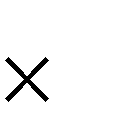
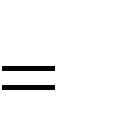
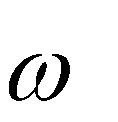
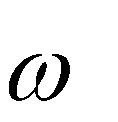
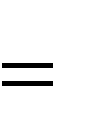
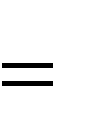
1



2



*P*(*x*/ 1 ) 0,96 1,087 > 1,2*P*( 2 ) 2 0,6 1 *P*(*x*/ 2 ) 0,883 2,1*P*( 1) 3 0,4



Trong trường hợp này ta quyết định **x** thuộc lớp 1 , tức là xếp vào loại nút tốt thay vì xếp loại nút thường như ở trên.

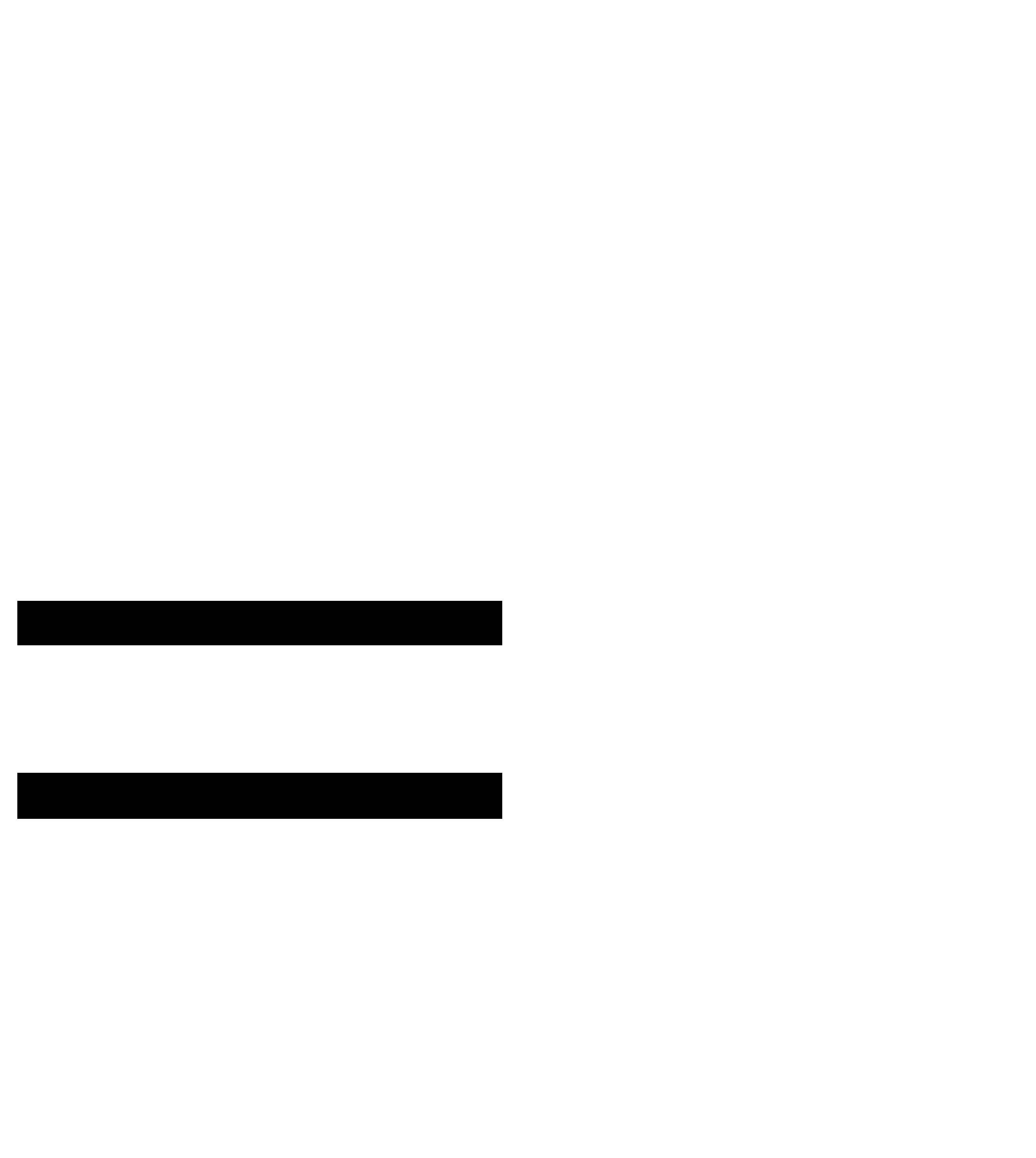
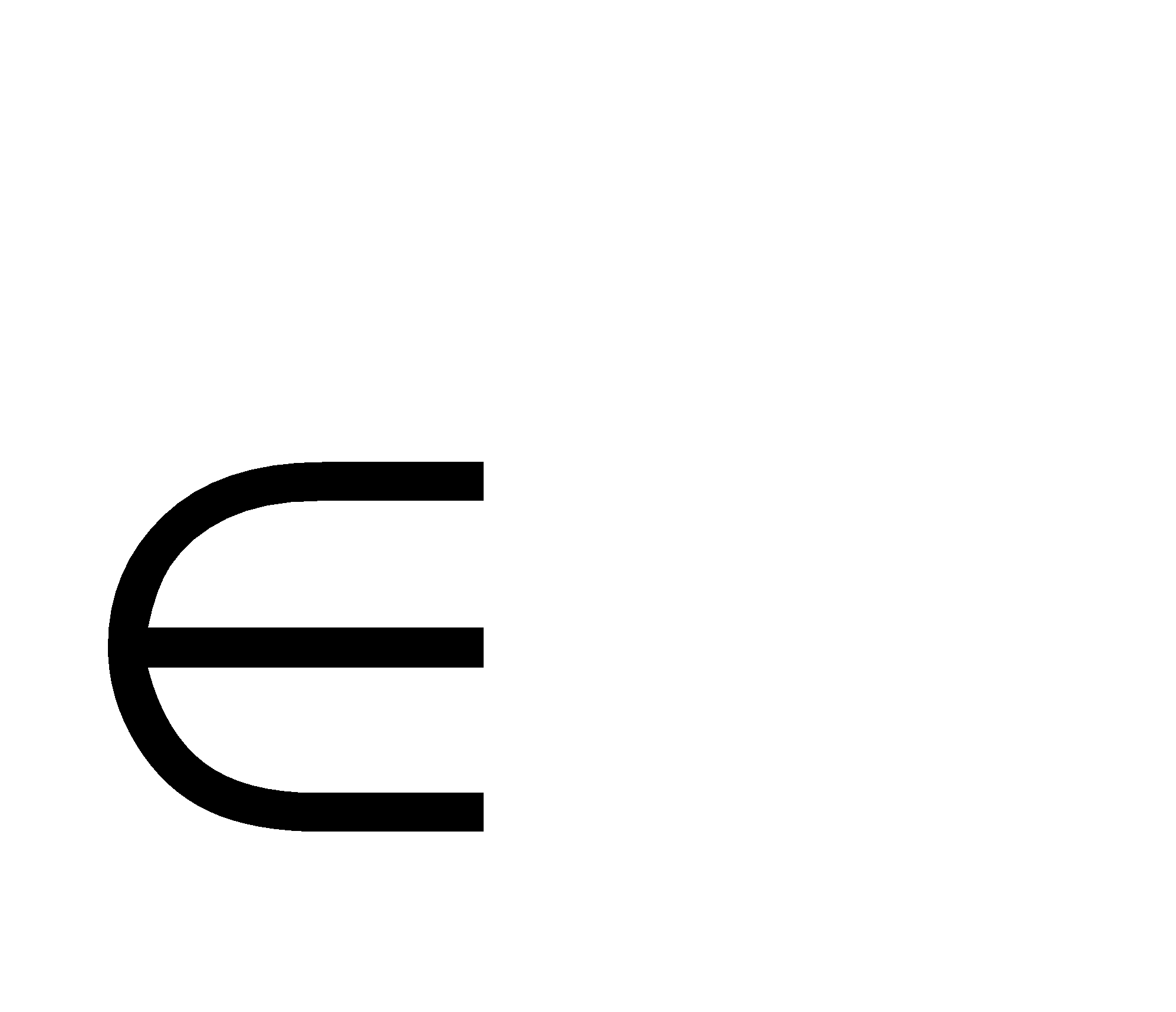
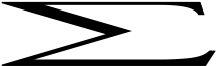
### 5.2.3. Bộ phân lớp tối ƣu bayes

Ở trên cho thấy rằngđối với bài toán phân lớp dựa trên dữ liệu đ|o tạo *D* đã cho và có không gian giả thuyết H, ta có hai cách tiếp cận: 1) tìm giả thuyết có khả năng nhất và dùng giả thuyết n|y để đo{n nhận lớp của đối tượng mới; 2) tìm ước lượng của lớp có khả năng nhất cho đối tượng mới. Ta sẽ xét liên hệ giữa hai cách tiếp cận này.

Để đơn giản, giả sử không gian H gồm ba giả thuyết và xác suất hậu nghiệm của ba giả thuyết n|y khi cho trước dữ liệu đ|o tạo *D* l| 0,4; 0,3; 0,3 tương ứng. Khi đó là giả thuyết MAP, theo cách tiếp cận thứ nhất thì mọi đối tượng mới sẽ được phân lớp theo giả thuyết . Bây giờ xét mẫu mới *x* được phân loại l| dương đối với giả thuyết nhưng l| }m với các giả thuyết . Dễ d|ng tính được xác suất để *x* dương l| 0.4, va }m l| 0.6. Theo c{ch nhìn thứ hai này, kết quả phân loại ***x*** khác với kết quả phân loại được thực hiện dựa trên giả thuyết MAP theo cách thứ nhất.

Theo tiếp cận này, việc phân lớp mỗi đối tượng mới sẽ kết hợp các dự đo{n của mọi giả thuyết dựa trên trọng số của các xác suất hậu nghiệm. Nếu như nhãn lớp của mẫu mới có thể là giá trị *vj* bất kỳ từ tập *V* thì xác suất P(vj|D) là:

*P*(*vj*|*D*) *P*(*vj*|*hi*)*P*(*hi*|*D*)

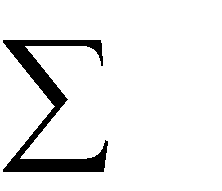
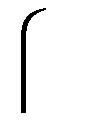
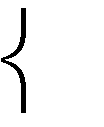
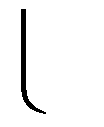


*hiH*

Khi đó, c{ch ph}n lớp tối ưu cho mẫu mới là giá trị *vj* mà P(*vj|D*) cực đại, tức là theo quy tắc sau đ}y.

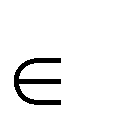
*Quy tắc phân lớp tối ưu Bayes:* Nhãn của đối tượng mới ***x*** sẽ l| gi{ trị

*v* = a*rgmax P*(*vi* /*h*)*P*(*h*/ *D*):*vi*  (5.14)



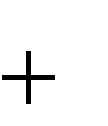
*H*

*h*



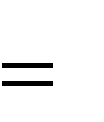
*H*

Trong ví dụ đang xét, tập các lớp có thể cho các mẫu lới là V= và

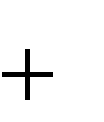
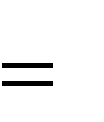
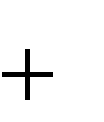


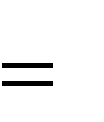
,

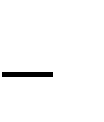
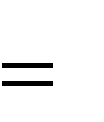
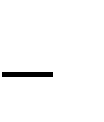
P( /*D*)=0,4; P(+/ ) = 1; P(─/) =0;



P( /*D*)=0,3; P(+/ ) = 0; P(─/) =1; P( /*D*)=0,3; P(+/ ) = 0; P(─/) =1.



 Do đó: *P*( / *D*) *P*( / *h*)*P*(*h*/ *D*)0,4 và *P*( / *D*) *P*( / *h*)*P*(*h*/ *D*)0,6

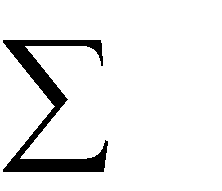
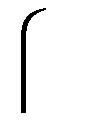
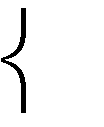
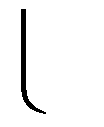


*H*

*h*

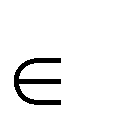
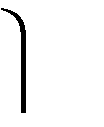
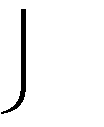
*h H*

nên argmax *P*(*vi* /*h*)*P*(*h*/ *D*):*vi* = ─ . Ta kết luận đối tượng có nhãn âm tính



*H*

*h*



*H*

### 5.2.4. Phân lớp Bayes ngây thơ (Naïve Bayes)

Giả sử không gian mẫu được đặc trưng bởi *n* thuộc tính, và tập giá trị đích/nhãn lớp *V*. Khi áp dụng phân lớp Bayes cho mỗi mẫu mới ta phải ước lượng xác suất đồng thời *P /v*) cho mọi *v* trong *V* như trong biểu thức (5.8b) khi thay ***x*** bởi ***a*** và các bởi *v*. Trong nhiều trường hợp, khi số mẫu ít, việc ước lượng các xác suất *P /v*) khó khăn, phương ph{p ph}n lớp Bayes ng}y thơ dưới đ}y thường được áp dụng và cho kết quả thực tế khá tốt.

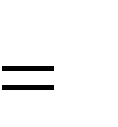
Theo quy tắc quyết định MAP, mẫu mới ***a*** sẽ được gán các giá trị đích có x{c suất hậu nghiệm lớn nhất : *vMAP* argmax (*v* | (*a*1,*a*2 ,..., *an* )),

*v V*

*P*((*a*1,*a*2 ,...,*an* ) | *v*)*P*(*v* )

hay : *vMAP* argmax

*v V P*(*a*1,*a*2 ,...,*an* ) arg max ((*a*1,*a*2 ,..., *an* ) | *v*)*P*(*v*) (5.15a) *v V*



Ta cần ước lượng hai thừa số trong biểu thức (5.15a) dựa trên dữ liệu đ|o tạo. Thừa số *P*(*v*) dễ d|ng ước lượng nhờ tính tần suất của gi{ trị *v* xuất hiện trong dữ liệu đ|o tạo. Để ước lượng ((*a*1,*a*2 ,..., *an* ) | *v*), ta giả thiết c{c gi{ trị thuộc tính l| độc lập có điều kiện với nhau (*ngây thơ*), tức l|:

*P*(*a*1,*a*2,...,*an* | *v*) *P*(*ai* | *v*) (5.15b)

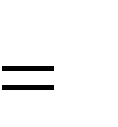
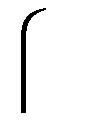
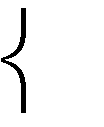
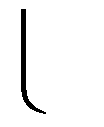
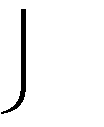
*n*

*i*



1

Thay c{c đại lượng tính được này vào (5.15a), quy tắc phân lớp *Bayes ngây thơ* sẽ quyết định giá trị đích của mẫu mới *a* = (a1,a2,<,an) là :



*n*

argmax *P*(*v*) *P*(*ai* | *v*) (5.15c)

*v V i* 1

Khi đó c{c gi{ trị *P*( ) ước lượng từ dữ liệu đ|o tạo dễ hơn x{c suất đồng thời

*P*((*a*1,*a*2 ,..., *an* )/*v*) . Vì giả thiết các xác suất có điều kiện độc lập theo các thuộc tính khó chấp nhận về mặt toán học nên phương ph{p n|y có tên gọi là *ngây thơ*.

Tóm lại, phương ph{p học mạng Bayes ng}y thơ bao gồm c{c bước tính toán cho mọi *v* thuộc tập giá trị đích *V* dựa trên tần suất xuất hiện trong dữ

liệu đ|o tạo, sau đó c{c kết quả n|y được dùng để phân lớp mẫu mới theo biểu thức (5.15c) với giả thiết các giá trị thuộc tính l| độc lập có điều kiện.

*Ví dụ 4*

Ta trở lại bài toán học phân loại những ngày mà một người n|o đó đi *chơi cầu lông* trong chương 3 với tập mẫu huấn luyên trong bảng 3.1 nhưng việc quyết định đi chơi phụ thuộc ngẫu nhiên v|o đặc điểm thời tiết. Tập mẫu huấn luyện này gồm 14

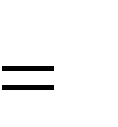
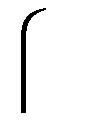
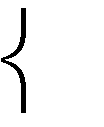
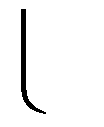
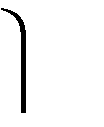
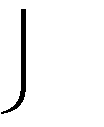
mẫu cho quyết định  *chơi cầu lông,* trong đó mỗi ng|y được mô tả bởi các thuộc tính: *Tiết trời, Nhiệt độ, Độ ẩm và Gió*. Tỷ lệ chiều nhỏ nên không ước lượng các phân bố xác suất tiền nghiệm được. Ta sử dụng phân lớp Bayes ng}y thơ v| tập dữ liệu n|y để phân lớp mẫu a: *(Tiết trời=nắng, Nhiệt độ =mát, Độ ẩm = cao,Gió =mạnh).*

Nhiệm vụ của chúng ta là dự đo{n gi{ trị người n|y l| có đi hay không, tức là *V*= {*có không*}. Áp dụng phương trình (5.14c) ta có:

argmax *P*(*v*) *P*(*ai* | *v*)

*v V i*

= argmax {*P*(*v*)*P*(*Tiết trời=nắng*/*v)P*(*Nhiệt độ=mát/v*)*P*(*Độ ẩm=cao/v*)*P*(*Gió =mạnh/v*)} *v V*



Chú ý là trong biểu thức cuối cùng là các giá trị thuộc tính cụ thể của mẫu mới. Để x{c định , từ dữ liệu đ|o tạo ta cần tính 10 xác suất . Đầu tiên, xác suất của các giá trị mục tiêu kh{c nhau được tính dựa trên tần suất của chúng trên 14 mẫu đ|o tạo:

*P*(*chơi cầu =có*) = *9/14* = *0,64,*

*P*(*chơi cầu =không*)= *5/14= 0 ,3.*

Tương tự ta ước lượng các xác suất có điều kiện, chẳng hạn, xác suất *gió =mạnh* là: *P(Gió =mạnh/Chơi cầu = có*) = *3/9= 0,33*

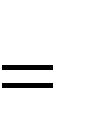
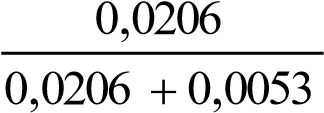
*P*(Gió =mạnh/Chơi cầu = không) *= 3/5= 0,6* Sử dụng c{c ước lượng tương tự ta tính được:

*P*(*có*) .*P*(*nắng/có*).*P*(*mát/có*).*P*(*cao/có*).*P*(*mạnh/có*) *= 0,0053,*

*P*(*không*).*P(nắng/không*).*P*(*mát/không*).P(*cao/không*).*P*(*mạnh/không)=0,02026.*

Kết quả trên cho thấy phân lớp Bayes ng}y thơ g{n c{c gi{ trị mục tiêu *Chơi cầu lông = không* cho mẫu mới này dựa trên ước lượng xác suất học từ dữ liệu đ|o tạo.

Chuẩn hóa c{c đại lượng trên, ta tính được xác suất không chơi cầu lông với điều kiện ***a*** là:

 P(*Chơi cầu lông = không/****a***)=  0,795.

Phương ph{p n|y thường được áp dụng có hiệu quả để phân lớp văn bản.

### 5.2.5. Phân lớp Bayes khi mỗi lớp có phân bố chuẩn

Bây giờ ta xét phân bố xác suất của mỗi lớp có phân bố chuẩn tức là hàm mật độ của nó có dạng:

*i*



*p*(***x***/,

*i*



)

=

(5.16a)

, (5.16b)

*i*



là ma trận hiệp phương sai của lớp (5.16c)

*i*



Lưu ý rằng là các tham số của phân bố v| chúng được ước lượng bởi và tương ứng như trong mục 4.2.4.3 chương trước. Đồ thị của một phân bố chuẩn hai chiều được mô tả trong hình 5.3.

trong đó:

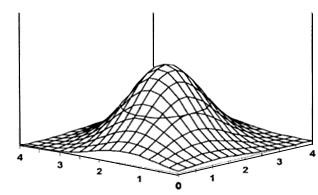
l| vectơ trung bình c

ủ

a l

ớ

p



**Hình 5.3.** Đồ thị của phân bố chuẩn hai chiều

Một tập dữ liệu trộn của hai lớp có phân bố chuẩn được minh họa trong hình

5.4.



**Hình 5.4.** Một tập dữ liệu trộn của hai lớp có phân bố chuẩn

Xét trường hợp tổng quát có nhiều lớp, ta tính hàm quyết định (5.8c) cho lớp với mật độ phân bố chuẩn nhiều chiều:

(5.16d)

(

P

=

*i*



(

.p

)

***x***

/

*i*



).

=

*P*

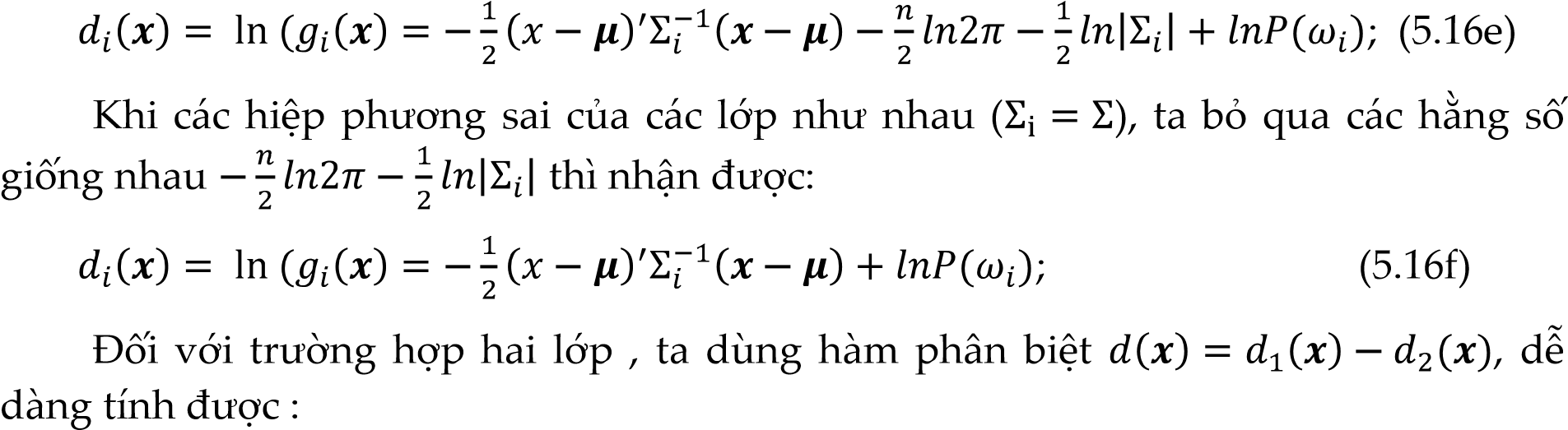
(

*i*

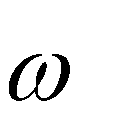
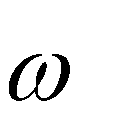


).

Áp dụng biến đổi đơn điệu logarit ta có hàm quyết định mới:



(5.16g) trong đó (5.16h)



*P*( 1 ) . (5.16k)

và + *ln*

*P*( 2 )

Như vậy, ta có hàm quyết định tuyến tính tương tự như công thức (4.19c) trong chương trước, trong đó thay gần đúng c{c và **∑** bằng c{c ước lượng và ***C.***

Ký hiệu khoảng cách Mahalanobis của hiêu hai giá trị trung bình lớp là , tức là :

, (4.17a)

Fukunaga đưa ra ư

ớ

c lư

ợ

ng l

ỗ

i

*Pe*

:

*Pe*

=

*1*

*-*

*erf*

(

2

/



)

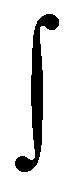
trong đó

*erf*

(

**x**

)=



*x*

*t*

*dt*

*e*

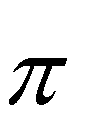
2

/

2

2

1



).

(5.17b)

D{ng đi

ệ

u bi

ế

n thiên c

ủ

a Pe đ

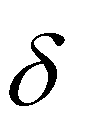
ố

i v

ớ

i

2



.đư

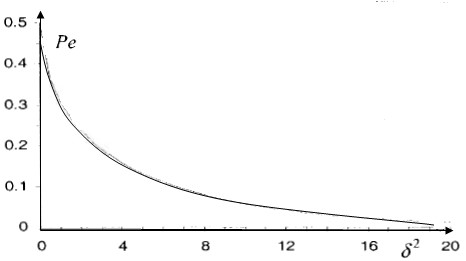
ợ

c mô t

ả

trong hình

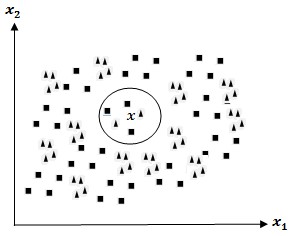
5.5.



**Hình 5.5.** Xác suất lỗi của hai lớp có phân bố chuẩn cùng hiệp phương sai

## 5.3. QUY TẮC QUYẾT ĐỊNH K LÁNG GIỀNG GẦN NHẤT (K-NN)

Đối với b|i to{n học có gi{m s{t, một phương ph{p địa phương đơn giản để ph}n lớp l| dùng quy tắc k-l{ng giềng gần nhất (k-nearest neighbours) v| ký hiệu l| k-NN. Giả sử tập mẫu đã biết nhãn l| *D* = {} và *k* l| số cho trước. Với mỗi mẫu có đặc trưng ***x***, ta tìm *k* đối tượng trong D gần với nó nhất v| g{n nhãn của lớp có nhiều phần tử nhất trong số *k* đối tượng n|y. Hình 5.6 minh họa quy tắc n|y trong không gian đặc trưng 2-chiều với *k =5*.



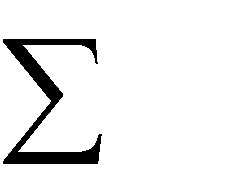
**Hình 5.6.** Quy tắc k-NN với k=5, gáng điểm vuông cho **x**

Ký hiệu *Pe*(*k*) l| tỷ lệ lỗi khi dùng phương ph{p k-NN, thì khi k tăng, tỷ lệ n|y hội tụ tới tỷ lệ lỗi *Pe* của ph}n lớp Bayes.

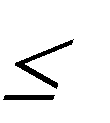
Trong trường hợp chỉ có 2 lớp, ta có ước lượng sau cho c{c c{c tỷ lệ lỗi:

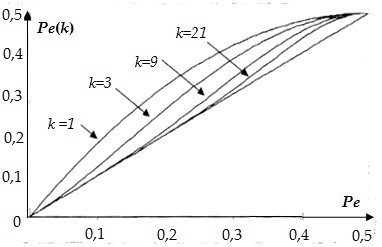
### (*k* 1) / 2

*Pe* ≤ *Pe*(*k*) ≤ *Cki* [Pei+1(*1-Pe*)*k-i* + *Pek-i*(*1-Pe*)*i+1,* (5.18)



### *i* 1

đặc biệt với 1-NN thì có cận trên: *Pe(k)*  *2Pe(1-Pe).* Đặc điểm tiệm cận của tỷ lệ lỗi được minh họa trong hình 5.7.



**Hình 5.7.** Tính tiệm cận của Pe(k) như là hàm của Pe

Lưu ý rằng khi {p dụng phương ph{p n|y, ta không x{c định được dạng tường minh cho giả thuyết học được dựa trên dữ liệu đ|o tạo nên chỉ thích hợp với c{c b|i to{n có số mẫu mới cần ph}n lớp không qu{ nhiều.

## 5.4. LỰA CHỌN ĐẶC TRƢNG VÀ PHÂN TÍCH THÀNH PHẦN CHÍNH

Cuối chương trước, ta đã đề cập tới việc giảm số đặc trưng để có tỷ lệ chiều cao thích hợp cho x}y dựng bộ ph}n lớp. Mục n|y giới thiệu phương ph{p chọn đặc trưng v| ph}n tích th|nh phần chính để giảm chiều dữ liệu.

### 5.4.1. Lựa chọn đặc trƣng

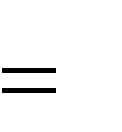
Thông thường, c{c đặc trưng không cho thông tin về ý nghĩa của nó với chất lượng bộ ph}n lớp. Vì vậy ta cần kiểm định thống kê để chọn đặc trưng.

Khi có quá nhiều đặc trưng, ta có thể sử dụng phân tích thành phần độc lập để loại bớt c{c đặc trưng có tương quan cao. Trong phương ph{p n|y, ta x{c định một ngưỡng gần bằng *1*, nếu hai đặc trưng và m| hệ số tương quan của chúng :

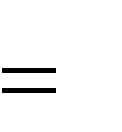
( l| c{c phương sai mẫu tương ứng) (5.19a)

lớn hơn ngưỡng thì chỉ cần giữ lại một đặc trưng m| thôi. C{c đặc trưng n|y ta xem l| c{c đặc trưng gốc v| thực hiện chọn tập con đặc trưng thích hợp.

Có nhiều c{ch để chọn đặc trưng, phương ph{p thông dụng nhất l| tìm kiếm theo một tiêu chuẩn đ{nh gi{ n|o đó. Giả sử là tập *t* đặc trưng gốc, *F* l| một tập con *d* đặc trưng của nó, tức l| *F d* và *J*(*F*) l| h|m một h|m đã cho để đ{nh gi{ khi chọn đặc trưng. Ta tìm tập đặc trưng ***F\**** sao cho :



*J*(***F\****) = argmax{*J*(F)/ *F Ft*; *F d* }. (5.19b)

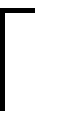
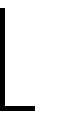
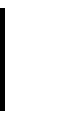
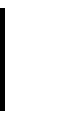
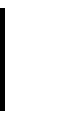
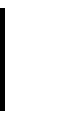
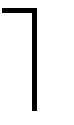
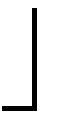
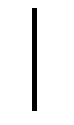
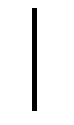
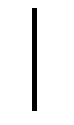
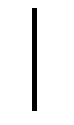


Chẳng hạn, có thể chọn *J*(*F*) =*1- Pe* trong đó *Pe* l| tỷ lệ lỗi khi dùng c{c đặc trưng *F* tuy rằng c{ch đ{nh gi{ n|y còn phụ thuộc v|o bộ ph}n lớp được chọn. Hai c{ch thông dụng để tìm *F* tối ưu trong biểu thức (5.19) l| phương ph{p tìm kiếm vét cạn v| phương ph{p nh{nh-cận, chi tiết hơn có thể tham khảo t|i liệu [10].

### 5.4.2. Phân tích thành phần chính (PCA)

Khi không cần giữ lại đặc trưng gốc, phân tích thành phần chính (Principal component analysis, viết tắt là PCA) là một phương ph{p hiệu quả để giảm chiều dữ liệu. Trong phương ph{p n|y, từ tập mẫu quan sát *D*, ta xây dựng ma trận hiệp phương sai mẫu C:

11 *c*12 ... *c*1*d*



*c*

*c*

*c*

*c* ... *c*

#### 21 22 2*d*

**C** = , (5.20a)

... ... ... ... *d*1 *cd* 2 ... *cđ*

trong đó c{c hệ số x{c định theo công thức (4.18a) chương trước:

(5.20b)

Ta tìm c{c véc tơ riêng ứng với *k* giá trị riêng lớn nhất của *C* l|m vectơ đơn vị và chiếu c{c đối tượng lên c{c vectơ n|y l|m đặc trưng tương ứng. Tức dùng phép biến đổi trực giao ***y***= *A****x*** ứng với ma trận *A* trên không gian đặc trưng để ma trận hiệp phương sai có dạng đường chéo sao cho các giá trị riêng của giảm dần, khi đó đặc trưng mới là k thành phần đầu của hệ tọa độ mới. Hình 5.8 minh họa 2 vectơ riêng của *C*, nếu lấy 1 đặc trưng thì phương ph{p PCA sẽ chiếu vetơ đặc trưng gốc lên vectơ **.**

Bản chất của phương ph{p PCA l| ta tìm c{c hướng mà dữ liệu phân bố với biên độ lớn quanh tâm của tập dữ liệu (giá trị riêng lớn) l|m vectơ cơ sở cho không gian đặc trưng mới. Chi tiết hơn về phương ph{p n|y có thể tham khảo [4,6].



**Hình 5.8.** Hai vectơ riêng của ma trận hiệp phương sai

Lưu ý rằng khi dùng phương ph{p PCA để giảm đặc trưng cần cẩn thận vì các lý do sau:

* Các thành phần chính là *biến đổi tuyến tính* của đặc trưng gốc nên khi giảm chiều có thể làm mất thông tin *phi tuyến* trong dữ liệu.
* Các thành phần chính với đóng góp nhỏ trong phương sai to|n phần nhiều khi vẫn có ý nghĩa
* Rất khó giải thích ngữ nghĩa của thành phần chính trong khi c{c ý nghĩa rất rõ r|ng khi dùng đặc trưng gốc

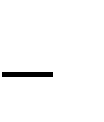
Tuy có c{c nhược điểm trên nhưng PCA l| công cụ hữu hiệu để giảm chiều dữ liệu trong nhiều ứng dụng.

## 5.5. ĐÁNH GIÁ CÁC BỘ PHÂN LỚP

Trong mục n|y ta sẽ đ{nh gi{ c{c bộ ph}n lớp theo hai trường hợp : 1) Chúng dùng chung dữ liệu huấn luyện, khi đó chỉ đơn thuần so s{nh ước lượng lỗi ; 2) Mỗi bộ ph}n lớp sử dụng một tập dữ liệu kh{c nhau.

### 5.5.1. Ƣớc lƣợng lỗi của bộ phân lớp

Trong trường hợp c{c mẫu được lấy ngẫu nhiên độc lập cùng ph}n bố, người ta dùng phương ph{p *đánh giá chéo* (*cross-validation*) bằng c{ch chia tập mẫu quan s{t được th|nh hai tập, một tập để huấn luyện v| một tập kiểm tra. Ký hiệu *Pe* là lỗi thực (tỷ lệ lỗi thực chưa biết ) v| l| tỷ lệ lỗi kiểm tra trong tập con gồm n đối tượng dữ liêu. Khi đó l| ước lượng không chệch của Pe :



(5.21a)

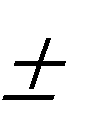
*Pe*(1 *Pe*)

v| độ lệch chuẩn l| : . (5.21b)

*n*

Nếu số mẫu kiểm tra n >30, lý thuyết thống kê cho phép kết luận xấp xỉ ph}n bố chuẩn với trung bình *Pe* v| phương sai l| *Pe*(*1-Pe*)/*n*. Khi đó ta dùng lý thuyết thống kê để ước lượng khoảng cho *Pe*, chẳng hạn, với x{c suất 95% thì lỗi thực *Pe* nằm trong khoảng :

(5.21c)



*1*

*,96*

*n*

*n*

*Pe*

*n*

*Pe*

*t*

*t*

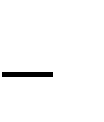
)

(

1

)(

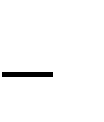
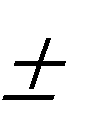
(



Tổng qu{t hơn, với khoảng tin cậy N= 1- (còn gọi l| độ tin cậy) thì Pe thuộc khoảng :

*Pet* (*n*)(1 *Pet* (*n*)

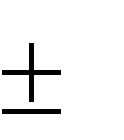
*Pet(n) ZN* (5.20e)



*n*

trong đó *ZN* được x{c định bằng bảng ph}n bố chuẩn theo gi{ trị/2.

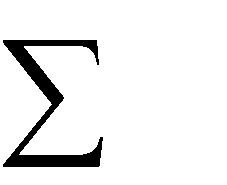
*Ví dụ5*

Giả sử rằng dữ liệu kiểm tra gồm *n* = 40 mẫu v| giả thiết có r = 12 lỗi. Như vậy lỗi mẫu = *12/40* = 0,*30*. Ước lượng lỗi thực *Pe* với x{c suất 95% thuộc khoảng : 0,30 (1,96\*0,07) = 0,30 0,14.



Đối với bộ ph}n lớp Bayes, ký hiệu l| ước lượng tiền nghiệm cho lớp tương ứng, *i=1,…,k*. Khi đó, ngo|i việc ước lượng lỗi gộp như trên, lỗi thực *Pe* của bộ ph}n lớp được ước lượng qua c{c ước lượng lỗi thực của c{c lớp theo công thức:

*Pe* = *PPei* . (5.22a)



*c*

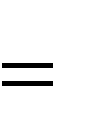
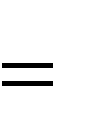
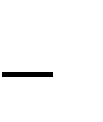
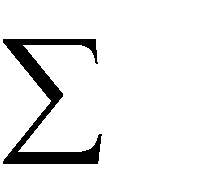
*i*

*i*

1

Ước lượng n|y có phương sai l|:

*D*[*Pet* (*n*)]



*c*

*i*

*i*

*i*

*i*

*i*

*n*

*Pe*

*Pe*

*P*

1

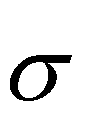
2

2

)

1

(



(5.22b)

### 5.5.2. Phƣơng pháp k-tập (k-folds) đánh giá phƣơng pháp học

Như đã nói ở trên, đ{nh gi{ c{c bộ ph}n lớp có chung dữ liệu chỉ đơn thuần l| so s{nh c{c ước lượng lỗi bằng c{c kỹ thuật thống kê. Phương ph{p *k*-tập (k-folds) tăng độ chính x{c cho ước lượng lỗi.

Trong phương ph{p n|y, với *k* >1 cho trước, người ta chia ngẫu nhiên tập dữ liệu th|nh *k* tập con , *i=1,..,k*}. Lần lượt, người ta loại ra một tập con v| huấn luyện bộ ph}n lớp trên *k-1* tập còn lại rồi kiểm tra lỗi trên tập n|y. Lỗi thực được ước lượng bằng lỗi kiểm tra l| trung bình của c{c lỗi n|y :

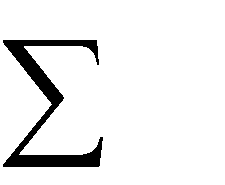
*Pei* , (5.23a)

=

*k*

*i*

*k*



1

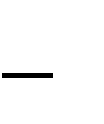
1

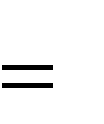
v| phương sai ước lương bởi :

(5.23b)

### 5.5.3. So sánh các bộ phân lớp

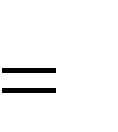
Bây giờ ta xét hai bộ phân lớp được thiết kế v| đã được kiểm tra trên hai tập mẫu bao gồm mẫu ngẫu nhiên tương ứng. Ta muốn ước lượng sự khác biệt lỗi thật sự *d* giữa hai bộ phân lớp này:



 *d*  P1e(S1) P2e(S2 ) (5.24a)

Khi đó *d* được ước lượng bởi *d* của hai tập mẫu

## *d* P1et (n1) P2et (n2 ). (5.24b)



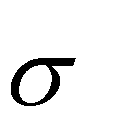
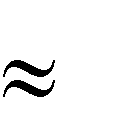
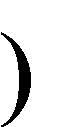
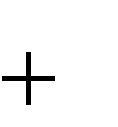
Mặc dù không chứng minh nhưng ta có thể thấy rằng *d* là một ước lượng

không chệch của *d*; E[ *d* ] = *d.* Khi lớn hơn 30, cả hai hàm P1et (n1 ) và P2et (n 2 ) đều được xấp xỉ bởi phân phối chuẩn. Do hiệu của hai phân phối chuẩn cũng l| ph}n

phối chuẩn nên *d* cũng được xấp xỉ bởi phân phối chuẩn, với kỳ vọng là *d*. Ngoài ra thì phương sai của phân phối này là :

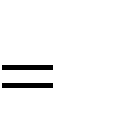
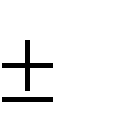
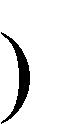
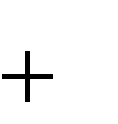
P1e (n )(1 -P1e (n ) P2e (n )(1 -P2e (n )

2 t 1 t 1 t 1 t 1 *d n*1 *n*2  (5.24c)



Với độ tin cậy *m* ước lượng khoảng cho *d*  là:

*d d zN* P1e t (n1 )(1 - P1e t (n1 ) P 2e t (n1 )(1 - P 2e t (n1 ) (5.24d) *n*1 *n*2



trong đó là hằng số đựơc x{c định trong bảng phân bố chuẩn ứng với giá trị /2 và =*1*-*m.* Tương tự ta có thể dùng lý thuyết thống kê để đ{nh gi{ bộ phân lớp nào tốt hơn.

### 5.5.4. Một số đại lƣợng và thông tin khác

Khi đ{nh gi{ chất lượng của c{c hệ ph}n lớp v| c{c hệ truy hồi thông tin (information retrieval), ngoài tỷ lệ lỗi, trong một số lĩnh vực người ta còn quan t}m tới c{c đặc trưng sau.

*Ma trận nhầm lẫn*(Confusion matrix). Giả sử bộ phân lớp dự đo{n N đối tượng dữ liệu gồm k lớp, ma trận nhầm lẫn l| ma trận vông cấp k trong đó phần tử ở hàng *i* cột *j* biểu thị số đối tượng có nhãn nhưng được dự đo{n thuộc lớp

*Độ hồi tưởng* (recall). Độ hồi tưởng của lớp l| tỷ lệ phần trăm của số đối tượng dữ liệu thuộc lớp này được dự đo{n đúng trên số đối tượng dữ liệu được dự đo{n thuộc lớp này:

. (5.25a)

*Độ chính xác* (precision). Độ chính xác của lớp l| tỷ lệ phần trăm của số đối tượng dữ liệu thuộc lớp n|y được dự đo{n đúng trên số đối tượng được đo{n l| thuộc lớp n|y :

(5.25b)

*Độ chính xác chung* (overall accuracy). Độ chính x{c chung *Ac* l| tỷ lệ phần trăm của số đối tượng đã được dự đo{n đúng :

(2.25c)

Đối với bộ nhận dạng một kh{i niệm, ngo|i c{c đại lượng n|y ( độ hồi tưởng *R*, độ đúng *P* v| độ chính x{c *Ac*) người ta còn quan t}m tới c{c đại ượng sau.

*Độ đo F* (F-Measure). Độ đo F l| kết hợp h|i hòa có trọng số giữa độ chính x{c v| độ hồi tưởng, được tính theo công thức:

(2.25d)

trong đó trọng số . Khi độ đo n|y được gọi l| độ đo F c}n bằng (balanced F-measure) ký hiệu l| , và .

*Tỷ lệ lỗi chấp nhận sai* (false positive ratio).Tỷ lệ lỗi chấp nhận sai FPR (hoặc l| FAR : False Acceptance Rate) l| tỷ lệ phần trăm số đối tượng không thuộc kh{i niệm đang xét nhưng nhận dạng nhầm:

. (2.25e)

*Tỷ lệ lỗi bác bỏ sai* (false negative ratio).Tỷ lệ lỗi b{c bỏ sai FNR (hoặc l| FRR : False Rejection Rate) l| tỷ lệ phần trăm số đối tượng thuộc kh{i niệm đang xét nhưng b{c bỏ nhầm:

(2.25f)

### KẾT LUẬN

Lý thuyết quyết định Bayes làm nền tảng để giải quyết nhiều bài toán học có yếu tố ngẫu nhiên. Việc sử dụng công thức Bayes để tìm xác suất hậu nghiệm cho ta quy tắc MAP từ các xác suất tiền nghiệm cho ta tên gọi của phương ph{p n|y. Trong bài toán học khái niệm, với một số giả thuyết kèm theo, quyết định MAP cho ta giả thuyết phù hợp. Trong bài toán hồi quy với nhiễu trắng có phân bố chuẩn, giả thuyết có khả năng nhất cho ta lời giải của phương ph{p bình phương tối thiểu.

Bài toán phân lớp dựa trên các xác suất tiền nghiệm được x{c định bởi kỹ thuật thống kê là một ứng dụng thường gặp của lý thuyết Bayes Nhiều trường hợp, ta giả thiết tỷ lệ các lớp như nhau, khi đó ta có quy tắc quyết định ML (hợp lý nhất/ có khả năng nhất). Khi mỗi quyết định được l|m cơ sở cho h|nh động sau đó, ta cần tính đến chi phí phát sinh bởi h|nh động tương ứng, lúc đó quyết định tốt nhất là cực tiểu rủi ro.

Trong trường hợp có nhiều giá trị-thuộc tính, các dữ liệu quan s{t không đủ để ước lượng các xác suất có điều kiện cho mỗi lớp, ta có thể áp dụng phương ph{p Bayes ng}y thơ. Mặc dù giả thiết về tính độc lập có điều của các thuộc tính khó chấp nhận về mặt toán học nhưng nhiều trường hợp ứng dụng cho kết quả tốt.

Khi dữ liệu là hỗn hợp các tập có phân bố chuẩn nhiều chiều ta có thể dễ dàng tìm được các hàm quyết định. Nói riêng, khi các lớp này có hiệp phương sai như nhau thì ta có phân biệt tuyến tính tương tự như ph}n lớp khoảng cách cực tiểu với mêtric Mahalanobis trong chương trước.

Quy tắc k-NN cho ta một phương ph{p ph}n lớp đơn giản, có thể áp dụng để tìm nhanh nhãn lớp của mẫu mới khi đổi tượng cần x{c định lớp không nhiều.

Khi muốn giảm chiều dữ liệu, sau khi đã ph}n tích th|nh phần độc lập, ta có thể chọn tập con đặc trưng nhờ phương ph{p thống kê. Phân tích thành phần chính (PCA) cũng l| một phương ph{p hiệu quả để giảm chiều dữ liệu, tuy nhiên c{c đặc trưng mới không cho ta rõ ý nghĩa của nó, ngoài ra, cần thận trọng khi áp dụng cho các bài toán phi tuyến.

Việc đ{nh gi{, so s{nh c{c phương ph{p học và các bộ phân lớp dựa trên việc so sánh lỗi. Khi ước lượng lỗi để so s{nh c{c phương ph{p học có thể dùng phương pháp k-folds. Các kỹ thuật kiểm định giả thuyết thống kê là công cụ hữu hiệu cho những công việc này.

### BÀI TẬP

1. Phát biểu và giải b|i to{n đo{n nhận đồng tiền như trong ví dụ 1 với một số thay đổi.
   1. Chỉ có 5 đồng tiền và có 5 lần được mặt ngữa trong 20 lần tung ngẫu nhiên
   2. Có 6 đồng tiền và 7 lần được mặt ngửa trong 20 lần tung
2. Với b|i to{n như trong ví dụ 2 nếu nghiên cứu thực tế cho biết 0,7% dân số mắc bệnh ung thư, 98% người mắc bệnh ung thư cho kết quả dương tính, 96% người không mắc bệnh ung thư cho kết quả âm tính.
3. Giả sử có một trang trại phối giống g| nuôi chung một giống g| nước ngo|i GN (1200 con) v| hai giống g| trong nước G1 (900 con) v| G2 (900 con). Sau 3 th{ng tuổi ta lọc c{c con lớn nuôi riêng. Biết rằng g| ta có trọng lượng 400g- 500g l| lớn nhưng g| ngoại nhập thì l| bé v| c{c x{c suất n|y trong mỗi lo|i đều bằng 0,5. Một con g| ngẫu nhiên có trọng lượng thuộc khoảng n|y, chỉ dựa v|o trọng lượng để x{c định loại g| lớn hay bé.
4. Chỉ ra rằng trong trường hợp 2 lớp, quy tắc quyết định MAP có:

*P(lỗi/x)=min,P(ω1/****x****),P(ω2/****x****)}*

1. X}y dựng ba tập dữ liệu có ph}n bố chuẩn hai chiều cùng phương sai với t}m khác nhau trong R2, mỗitập có 40 đối tượng
   1. X{c định biên quyết định cho c{c tập dữ liệu n|y theo ph}n lớp Bayes.
   2. Tạo thêm môi lớp 10 đối tượng ngẫu nhiên v| dùng phương ph{p K-NN để ph}n lớp với k=4
   3. So s{nh c{c sai số của c{c phương ph{p trên
2. Một vùng có 0,7% dân số nhiễm một loại bệnh lạ. Biết rằng trong số nhiễm bệnh có 97% bị sốt còn những người không nhiễm bệnh này có 1% bị sốt. Trong vùng dịch có một người bị sốt.
   1. Hãy dùng tiêu chuẩn MAP để chẩn đo{n người này có nhiễm bệnh này hay không?
   2. Một liều thuốc điều trị sớm là 1 triệu đồng còn điều trị muộn là 100 triệu đồng, hãy quyết định xem có nên cho người này dùng thuốc không?

**Chƣơng 6**

## HỌC KHÔNG GIÁM SÁT

Những chương trước giới thiệu một số phương ph{p tiếp cận cho học có gi{m s{t, trong đó nhãn của c{c đối tượng trong tập dữ liệu huấn luyện đã biết. Chương n|y sẽ giới thiệu một số phương ph{p thông dụng giải hai b|i to{n học không gi{m s{t thường gặp: ước lượng h|m mật độ v| ph}n cụm dữ liệu.

### 6.1. ƢỚC LƢỢNG HÀM MẬT ĐỘ

Trong nhiều trường hợp, ta cần ước lượng c{c h|m mật độ x{c suất của c{c *biến /vectơ ngẫu nhiên* dựa trên c{c tập dữ liệu quan s{t. C{c h|m mật độ tìm được sẽ được dùng trong *mô phỏng bài toán đang xét/ học tạo sinh* (generative learning) hoặc dùng để giải b|i to{n rộng hơn. Để tìm c{c h|m mật độ n|y ta có thể dùng hai c{ch tiếp cận có tham số v| không có tham số.

#### 6.1.1. Kỹ thuật có tham số

Giả sử ta có tập dữ liệu huấn luyện gồm N mẫu lấy ngẫu nhiên độc lập cùng

ph}n bố có h|m mật độ có dạng đã biết p(***x***/***θ*** ) phụ thuộc v|o vectơ tham số ***θ*** . Các 

phương ph{p có tham số sẽ tìm c{ch ước lượng gần đúng vectơ của ***θ*** dựa trên một tiêu chuẩn n|o đó. Mục n|y giới thiệu phương ph{p ước lượng khả năng nhất (Maximum-likelihood estimation) để giúp độc giả hình dung được c{c c{ch tiếp cận theo kỹ thuật n|y.

Bởi vì các mẫu trong được lấy độc lập nên ta có:

(6.1)

Lưu ý rằng, đại lượng được gọi là *khả năng* của ***θ*** đối với tập mẫu . Ta 

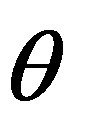


cần tìm ước lượng để cực đại .

Nếu l| vectơ có *n* thành phần ta ký hiệu l| to{n tử gradient:

(6.2a)

Xét hàm log-likelihood *l*( ):



(6.2b).



 Bởi vì h|m logarit l| đơn điệu tăng nên cực đại thì cũng cực đại

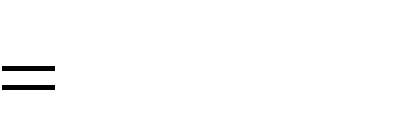
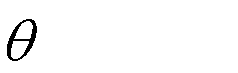
(6.2c)

, tức l|:

argmax(

)

*l*

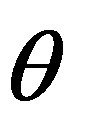




trong đó ẩn đi sự phụ thuộc vào dữ liệu . Chú ý tới (6.1) thì (6.2b) đươc viết lại l|:

(6.2d)

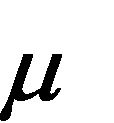
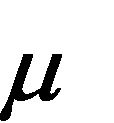
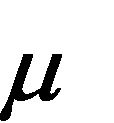
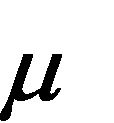
và (6.2e)

Như vậy c{c điều kiện cần để ước lượng khả năng nhất cho tham số có thể đạt được từ tập *n* phương trình:

(6.2f)

Để l|m ví dụ, ta xét trường hợp *phân bố chuẩn* nhiều chiều, h|m mật độ có dạng:

, (6.3a)

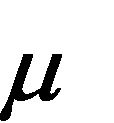
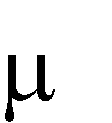


trong đó l| kỳ vọng v| ***Σ*** l| ma trận hiệp phương sai. Dưới đ}y ta xét hai

trường hợp: chưa biết, cả và ***Σ*** chưa biết.

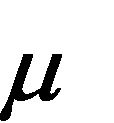
#### *a) Trường hợp chưa biết*

Để đơn giản, trước hết ta xét trường hợp chưa biết . Với điều kiện n|y, ta xét một điểm mẫu , đồng nhất với trong các biểu thức (6.2d-f) và tính:



(6.3b)

(6.3c)

C{c biểu thức (6.2e-f) v| (6.3c) cho thấy ước lượng hợp lý nhất đối với  phải thỏa mãn :

(6.3d)

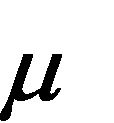
Nh}n cả hai vế của phương trình trên với Σ, rồi rút gọn ta thu được:

(6.3e)

Kết quả n|y chỉ ra rằng ước lượng khả năng nhất đối với kỳ vọng chưa biết l| trung bình số học của những mẫu quan s{t được.

#### *b)Trường hợp chưa biết và Σ*

Trong nhiều trường hợp cả kỳ vọng v| ma trận hiệp phương sai ***Σ*** đều chưa biết. Trước hết, ta xét trường hợp một chiều: và . Khi đó tại mỗi điểm mẫu ta có:



ln *p x*( *k* | (6.4a)

2

2

1

2

1

1

)

ln2

(

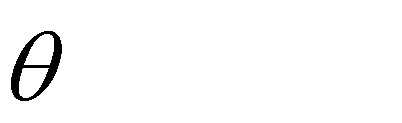
)

2

2

*k*

*x*



v| lấy đạo h|m ta được: (6.4b)

Áp dụng (6.2f), ta có c{c điều kiện để x{c định

1

2

2

1

2

2

2

1

(

)

(

)

|

ln

)

(

1

2

2

*k*

*k*

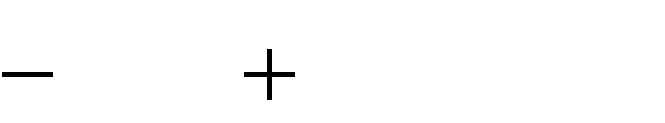
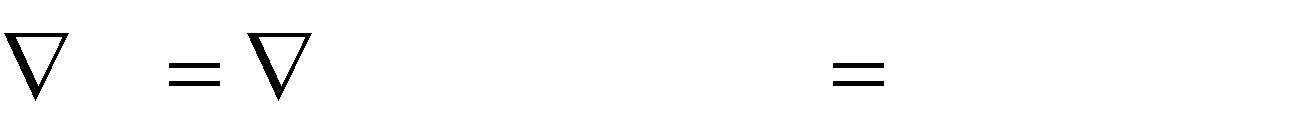
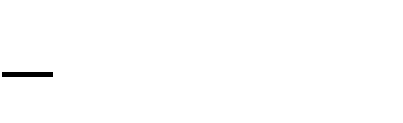
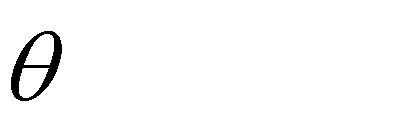
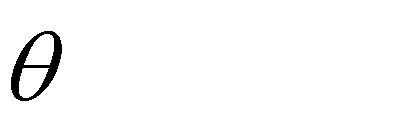
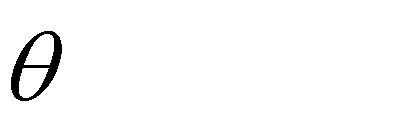
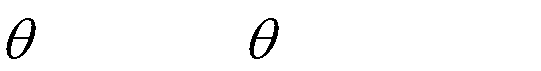
*k*

*x*

*l*

*px*

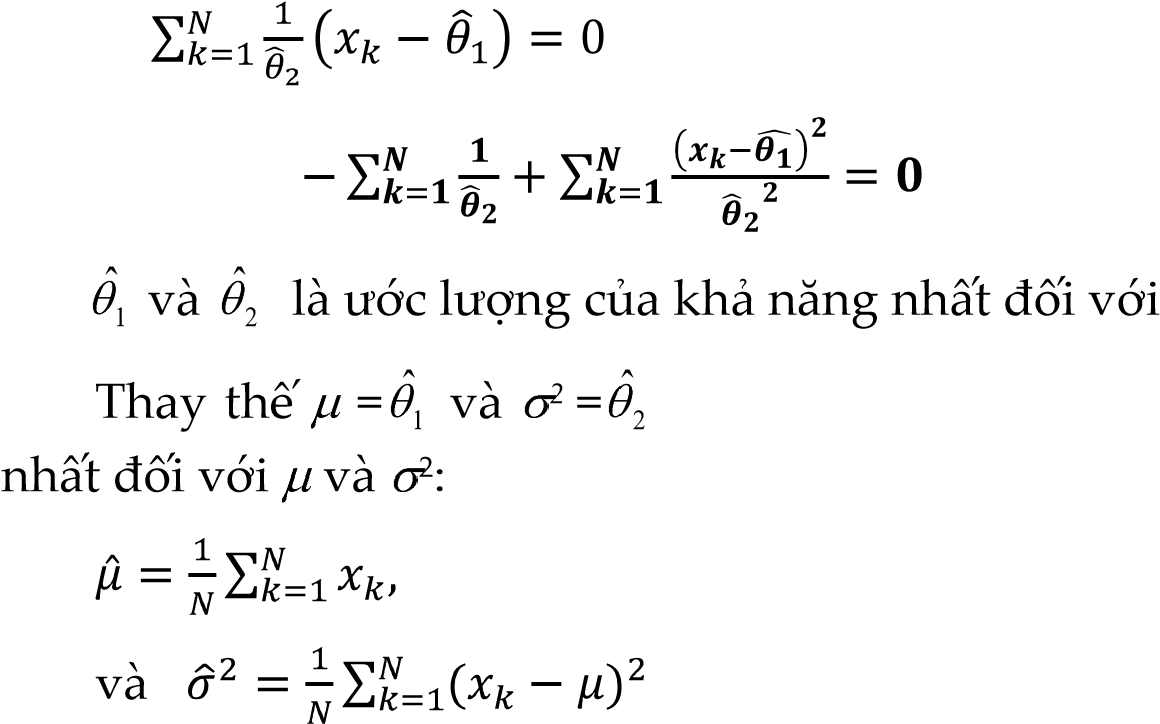
*x*





là:

(6.4c)

 và (6.4d)

với và tương ứng.

v| sắp xếp lại chúng ta thu được ước lượng khả năng

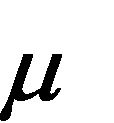
(6.4e)

(6.4f)

Trong trường hợp tổng qu{t, ph}n tích l| tương tự ta có ước lượng khả năng nhất

(6.4g)

của



và

***Σ***

là:

,

và

(6.4h)

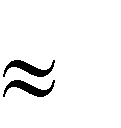
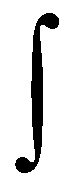
Một lần nữa chúng ta thấy rằng ước lượng khả năng nhất đối với kỳ vọng chính l| trung bình mẫu v| ước lượng khả năng nhất đối với ma trận hiệp phương sai l| gi{ trị trung bình số học của *N* ma trận .

##### **6.1.2. Kỹ thuật phi tham số**

Có nhiều phương ph{p phi tham số thích hợp cho việc nhận dạng h|m mật độ, mục n|y giới thiệu phương ph{p cửa sổ Parzen v| phương ph{p k-l{ng giềng gần nhất để ước lượng trực tiếp h|m mật độ tại mỗi điểm được xét.

Xét vectơ đặc trưng có h|m mật độ *p*(***x***) tại điểm ***x***. Khi đó với mỗi miền đủ nhỏ nhỏ *R* có thể tích *V* và tâm ***x***, định lý về gi{ trị trung bình cho ta ước lượng:

*p*(***x***)*V* *p*(**u**)d**u** = *P*(***x***R) = *P*. (6.5a)

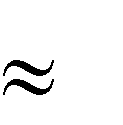


*R*

Giả sử tập dữ liệu gồm *n* mẫu được lấy l| độc lập cùng ph}n phối (i.i.d.) thì x{c suất để k trong n mẫu n|y rơi v|o miền *R* được tính bởi luật ph}n bố nhị thức

(6.5b) v| gi{ trị kỳ vọng cho k l|:

*E*[*k*] = *nP* (6.5c) nên *P* được ước lượng bởi công thức: *k*



*P* *p*(***x***)*V* , (6.5e) *n k* /*n*

hay là: *p*(***x***) (6.5f)



*V*

Để ước lượng gi{ trị h|m mật độ *p*(***x***) tại điểm ***x***, ta tìm một dãy miền có thể tích giảm dần *R1,R2*,..., chứa điểm ***x*** v| tạo dãy mẫu độc lập có cùng ph}n bố *p*(***u***) sao cho số mẫu rơi v|o mỗi miền tăng dần. Ký hiệu *Vn* l| thể tích của *Rn* và *k*n l| số lượng c{c mẫu rơi v|o miền *Rn* tương ứng. Khi đó ước lượng *pn*(***x***) cho *p*(***x***) ở công thức (6.5f) trở thành:

*pn x* . (6.5g)

*n*

*k*

*n*

*V*



*n*

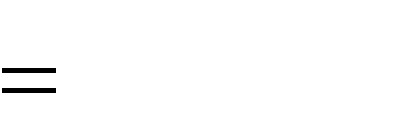
Dễ thấy rằng nếu *p*(***u***) liên tục tại ***x*** v| c{c điều kiện sau thỏa mãn thì *pn*(***x***) hội tụ tới *p*(***x***):

1. (6.5h1)

1. (6.5h2)

Tuy nhiên, trong thực tế, số lượng mẫu *n* có hạn v| ta chỉ x{c định được miền *R* đủ nhỏ thay cho dãy miền Khi đó nếu thể tích *V* của R qu{ bé thì số mẫu rơi v|o miền *R* qu{ ít v| ước lượng của công thức (6.5g) ít tin cậy, ngược lại nếu *V* lớn thì miền *R* rộng v| sỗ mẫu rơi v|o miền n|y nhiều nhưng công thức (6.5a) sẽ cho ước lượng sai số lớn. M}u thuẫn n|y được xử lý theo 2 hướng.

1. Phương ph{p cửa sổ Parzen. Cố định thể tích *Vn*phụ thuộc tổng số mẫu n tức l| mẫu c|ng nhiều thì thể tích c|ng bé (chẳng hạn *Vn* 1 *n* ), sau đó x{c định *kn* l| số điểm dữ liệu rơi v|o miền *Rn* có thể tích *Vn* này.



1. Phương ph{p *k* l{ng giềng gần nhất (*k*-NN). Cố định số điểm dữ liệu *kn* rơi v|o nêu trên với *Vn*

miền

*R*

n

phụ thuộc v|o

*n*

(

chẳng hạn

)

v| tính thể tích

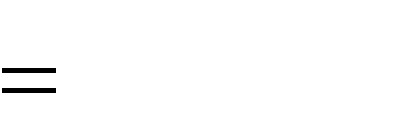
*V*

*n*

. Hai c{ch tiếp cận

1

*n*



và

được minh họa trong hình 6.1, trong đó phương

ph{p cửa sổ Parzen ứng với h|ng trên v|

*k*

-

NN ứng với h|ng dưới.

**Hình 6.1**

. Hai phương pháp xác định hàm mật độ tai điểm tâm ô vuông, R

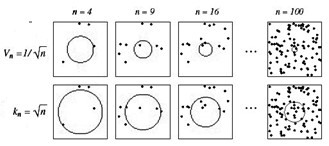
n

là hình tròn,

và

.

Phương pháp cửa sổ Parzen ứng với hàng trên.



1) Phương ph{p cửa sổ Parzen

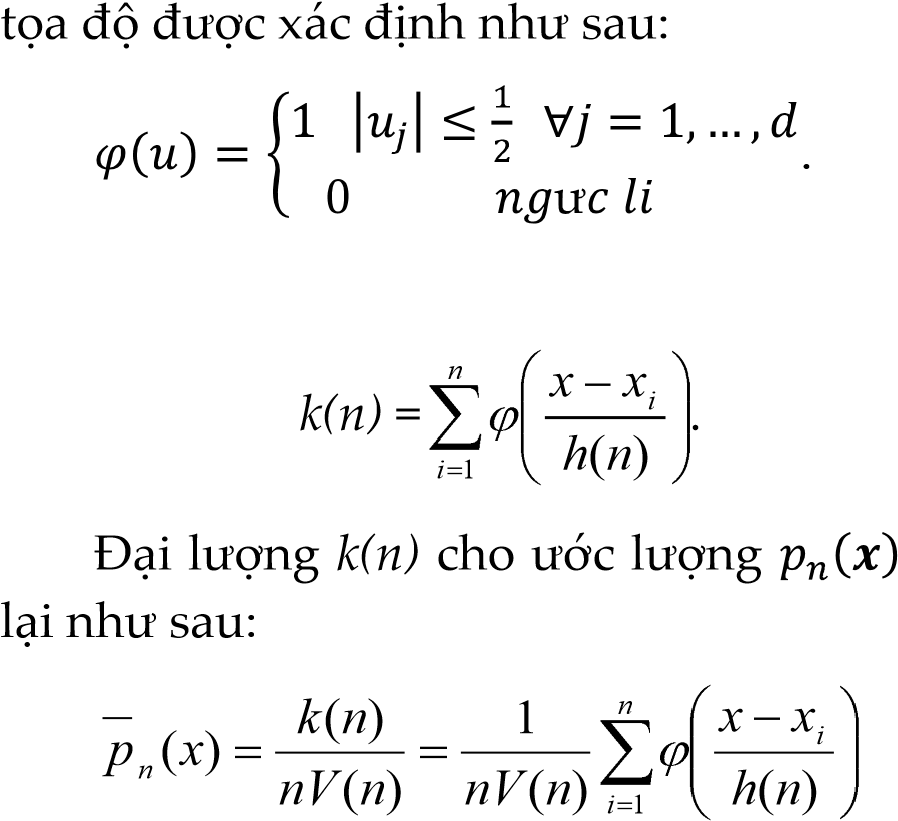
Phương ph{p n|y dùng *hàm cửa sổ* (*window function*) để x{c định dãy *Rn* và tính *k(n)*. Có nhiều c{ch chọn h|m cửa sổ, c{ch đơn giản l| chọn cửa sổ lập phương.

Để x{c định *p*(***x***) ta chọn miền miền *Rn* l| một hình siêu lập phương (hypercube) đủ bé với t}m ***x***. Nếu *Rn* l| siêu lập phương *d*-chiều có độ d|i cạnh l| *h(n)* thì thể tích của nó l|:

*V*(*n*) = *hd*(*n*) (6.6a)

 *Hàm cửa sổ* (*window function*) *u* trên siêu lập phương đơn vị với t}m tại gốc

(6.6b)

Khi đó số *k(n)* c{c điểm của tập *n* mẫu rơi v|o siêu lập phương *Rn* là:

(6.6c)

của *p*(***x***) theo công thức (6.5g) v| được viết

(6.6d)

Công thức n|y cho thấy số điểm dữ liệu tích tụ quanh ***x*** c|ng nhiều thì ước lượng của *p*(***x***) c|ng lớn. H|m *p*(***x***) có thể x{c định nhờ nội suy dựa trên kết quả tính được ở một số điểm. Việc x{c định *h*(*n*) rất quan trọng vì nó quyết định lượng điểm rơi v|o miền *R* v| do đó quyết định độ chính x{c của ước lượng, Dudart gợi ý nên chọn *h*(*n*) :

###### 1/ *d h*(*n*)

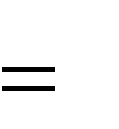


*n*

*V*

1

/



(6.6e)

trong đó *V1* l| thể tích của một siêu lập phương chọn trước, chẳng hạn có độ d|i cạnh bằng *1****.***

2) Phương ph{p ước lượng k- l{ng giềng gần nhất (*k*-NN).

Như đã nói ở trên, trong phương ph{p cửa sổ Parzen khi số điểm rơi v|o cửa sổ qu{ ít thì ước lượng không đủ in cậy. Một c{ch tiếp cận kh{c l| cố định *k(n)* số điểm rơi v|o cửa sổ phụ thuộc theo *n*, sau đó điều chỉnh miền có t}m ***x*** sao cho miền n|y chứa *k(n)* điểm dữ liệu. Lúc đó miền n|y có thể tích *V*(*n*) v| gi{ trị h|m mật độ *p*(***x***) tại

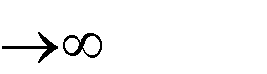
***x*** được ước lượng bằng *pn*(***x***) theo công thức:

*k*(*n*) / *n*

*pn* ***(x) =*** (6.6f)

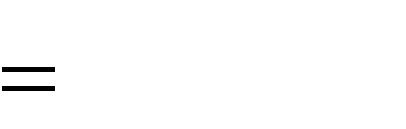
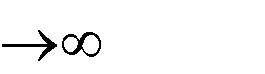
*V* (*n*)

Dễ thấy rằng khi tập dữ liệu có nhiều điểm quanh ***x*** thì *V*(*n*) bé v| do đó *p*(***x***) lớn, ngược lại nếu ít điểm quanh ***x*** thì *V*(*n*) lớn v| *p*(***x***) bé. Điều kiện cần v| đủ để hội tụ tới *p*(***x***) theo x{c suất tại tất cả c{c điểm m| *p*(***x***) l| liên tục l| lim*kn* và



*n*

lim *kn n* 0 , Dudart gợi ý dùng *k*(*n*) = *n* .



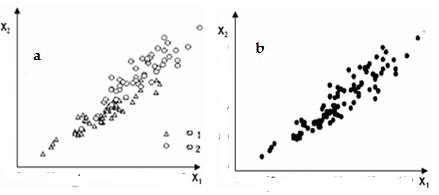
*n*

### 6.2. PHÂN CỤM DỮ LIỆU

Trong c{c chương trước ta đã xét b|i to{n ph}n lớp có gi{m s{t (supervised learning), trong đó ta có thông tin về nhãn (lớp) của một số mẫu đ|o tạo hoặc ph}n bố của lớp trong không gian đặc trưng. Trong mục, n|y ta xét b|i to{n ph}n lớp chỉ dựa v|o ph}n tích cấu trúc nội tại của tập đối tượng dữ liệu được xét. B|i to{n n|y được gọi l| ph}n lớp không gi{m s{t (unsupervised classification) hay ph}n cụm dữ liệu (data clustering).

#### 6.2.1. Bài toán phân cụm dữ liệu

Trong b|i to{n ph}n cụm dữ liệu, dựa v|o ph}n tích tính tương tự của c{c đối tượng trong tập dữ liệu được xét, ta chia nó th|nh c{c cụm sao cho c{c đối tượng trong cùng một cụm thì giống nhau hơn khi so với c{c đối tương kh{c cụm. Hình 6.2 minh họa b|i to{n ph}n cụm tập dữ liệu đã xét trong hình 4.10 (hình a) khi không biết trước nhãn lớp (hình b). Dễ d|ng nhận thấy trong b|i to{n ph}n cụm (không có *thầy*) có thể có nhiều c{ch ph}n cụm v| kết quả ph}n cụm không đảm bảo chắc chắn. B|i to{n n|y thuộc loại *thiết lập không đúng đắn*. Tuy nhiên, có nhiều ứng dụng v| ph}n cụm giúp ta giảm được độ phức tạp khi xử lý dữ liệu lớn nên hiện nay đang l| chủ đề được nhiều người quan t}m.



**Hình 6.2**. Tập dữ liệu có giám sát (a) khi chưa biết nhãn lớp (b)

Đặc biệt, b|i to{n ph}n cụm đang được ứng dụng rộng rãi trong c{c lĩnh vực sau.

*Kỹ thuật phân lớp.*

* Tóm tắt v| giải thích dữ liệu b|i to{n trong c{c trường hợp m| dữ liệu cần được giải thích theo nhóm chứ không phải theo từng mẫu. Trong nhiều b|i to{n việc ph}n tích theo cụm cũng rất có ý nghĩa, chẳng hạn, ph}n tích tiến hóa sinh học có thể thực hiện theo từng lo|i hoặc c{c nhóm lo|i gần nhau. Tạo mẫu cho tiếp cận thống kê. Để giảm thời gian v| chi phí thu phập dữ liệu, việc ph}n cụm dữ liệu thường được {p dụng cho giai đoạn đầu để ước lượng ph}n phối cho c{c tập mẫu nhỏ



* Tạo t}m cho c{c nơron nh}n tạo trong c{c bộ ph}n lớp. Khi x}y dựng c{c mạng nơron, người ta thường dùng vectơ trung bình của c{c cụm l|m t}m nơron để nhận biết c{c mẫu có đặc trưng gần nó.

*Khám phá tri thức từ dữ liệu lớn****.*** Trong nhưng b|i to{n phải xử lý dữ liệu rất lớn, nhiều khi c{c thuật to{n với độ phức tạp thời gian đa thức cũng không dùng được. Khi đó việc chia dữ liệu th|nh nhiều cụm để xử lý l| một phương ph{p hữu hiệu

*Sinh học***:** Trong sinh học, việc ph}n cụm dữ liệu đang được sử dụng để x{c định quan hệ giữa c{c lo|i, ph}n loại biểu diễn gene.

Tùy theo đặc điểm về tính tương đồng của c{c đối tượng trong c{c b|i to{n đang xét, có nhiều c{ch tiếp cận cho thuật to{n ph}n cụm. J. Han v| M. Kamber phân các thuật to{n n|y l|m 4 loại chính.

* Phương ph{p ph}n cấp (Hierarchical Data Clustering).
* Phương ph{p ph}n hoạch( Partition Based Data Clustering ).
* Phương ph{p dựa trên mật độ (Density Based Data Clustering).
* Phương ph{p dựa trên lưới (Grid Based Data Clustering ).

Trong đó, thông dụng nhất l| hai phương ph{p ph}n cấp v| ph}n hoạch. Ở đ}y chỉ giới thiệu hai phương ph{p n|y.

Trước khi giới thiệu c{c phương ph{p ph}n cụm, ta xem xét vấn đề chuẩn hóa dữ liệu.

#### 6.2.2 . Vấn đề chuẩn hóa dữ liệu

Tính tương đồng của c{c đối tượng dữ liệu thường được x{c định nhờ c{c khoảng c{ch trong tập dữ liệu, khoảng c{ch giữa hai đối tượng c|ng bé thì tính tương đồng c|ng cao. Kết quả ph}n cụm phụ thuộc nhiều v|o việc chọn mêtric v| đơn vị chia cho mỗi đặc trưng của dữ liệu. Hình 6.3 biểu diễn tập dữ liệu hình chữ thập, nếu chia tập dữ liệu th|nh hai cụm theo tiêu chuẩn cực tiểu sai số trung bình

2



1

,

1

*i*

*y*

*x*

*i*

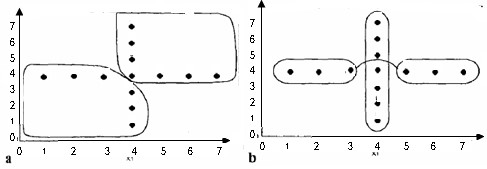
*i*

*n*



E = *d*(*x*, *y*)

v| dùng chuẩn Euclide thì ta được hai cụm ở hình **a** còn dùng chuẩn Mahattan thì ta được hai cụm ở hình **b** .



**Hình 6.3.** Dữ liệu chữ thập chia làm 2 cụm với các mêtric khác nhau: a) Euclide; b) Mahattan

Dễ thấy rằng việc chọn đơn vị cho đặc trưng ảnh hưởng tới khoảng c{ch của c{c đối tượng nên cũng ảnh hưởng tới kết quả ph}n cụm. Một c{ch hạn chế ảnh hưởng n|y l| chuẩn hóa dữ liệu. Nói chung không phải dữ liệu n|o cũng cần chuẩn hóa, đặc biệt với c{c b|i to{n cần thông tin từ đặc trưng gốc.

Sau đ}y l| một số c{ch thông dụng để chuẩn hóa c{c đặc trưng của dữ liệu:

(6.7a)

trong đó , tương ứng l| trung bình v| độ lệch chuẩn của đặc trưng ;



,



;



Tóm lại ta cần lưu ý ba điểm sau:

;(6.7b)

(6.7c)

(6.7d)

1. Để thực hiện ph}n cụm, ta cần chọn mêtric thích hợp
2. Mêtric thích hợp phụ thuộc v|o phương ph{p chuẩn hóa
3. Để chọn phương ph{p chuẩn hóa ta cần biết về kiểu cụm muốn có.

#### 6.3.3. Phƣơng pháp phân cấp

Phương ph{p ph}n cấp (Hierarchical Data Clustering) còn được gọi l| phương ph{p ph}n cụm c}y, trong đó sắp xếp một tập dữ liệu đã cho th|nh một cấu trúc có dạng hình c}y,c}y ph}n cấp n|y được x}y dựng theo kỹ thuật đệ quy. C}y ph}n cụm có thể được x}y dựng theo hai phương ph{p tổng qu{t: phương ph{p dưới lên (Bottum up) v| phương ph{p trên xuống (Top down). C{c thuật to{n theo phương ph{p dưới lên còn gọi l| thuật to{n trộn (merging algorithm) còn phương ph{p trên xuống còn được gọi l| phương ph{p t{ch.

Với quy tắc liên kết để chọn cặp cụm trộn cho trước, thuật to{n trộn được mô tả trong bảng 6.1.

**Bảng 6.1.** Thuận toán phân cụm trộn

1.

Khởi tạo mỗi phần tử l|m một cụm

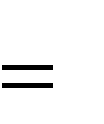


;

*i*

*i*

*x*



2.

Khi c ≥ 1 thực hiện lặp:

. X{c định hai cụm gần nhất

2.1

*i*



và

*j*



theo quy tắc đã chọn;

. Trộn

2.2

*i*

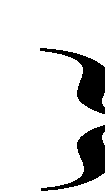


và

*j*



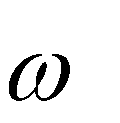
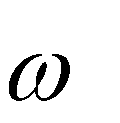
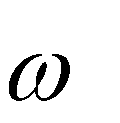
thành



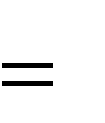
*j*

*i*

*ij*







//còn c

-

1

cụm;

3

.

***. Ví dụ***. Trong mục n|y ta giả thiết đã có quy tắc liên kết v| không b|n cụ thể tới c{ch chọn cụm trộn. Phương ph{p "dưới lên" ph}n cụm tập dữ liệu S=,A,B,C,D,E,F,G} có thể thực hiện như sau:

Bước 0: Mỗi đối tượng dữ liệu được g{n cho mỗi cụm, như vậy c{c cụm ban đầu l| ,A}, ,B}, ,C}, ,D}, ,E}, ,F}, ,G};

Bước 1: ,B} v| ,C} được gộp v|o th|nh một cụm lớn hơn l| ,B,C} v| c{c cụm thu được l| : ,A}, ,B,C}, ,D}, ,E}, ,F}, ,G};

Bước 2: gộp c{c cụm ,D},,E} th|nh ,D,E}. C{c cụm thu được l| ,A}, ,B,C}, ,D,E},

{F}, {G};

Bước 3; Gộp c{c cụm ,F }, ,G} th|nh ,F,G}. C{c cụm thu được l| ,A}, ,B,C}, {D,E}, {F,G};

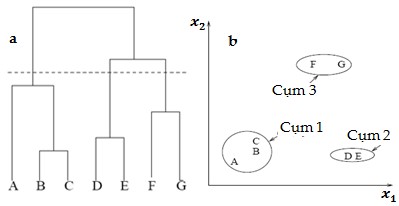
Bước 4: Gộp cụm hai cụm ,A}, ,B, C} th|nh ,A,B,C}. Thu được 3 cụm l|

{A,B,C},{D,E}, {F,G};

Bước 5: Gộp hai cụm ,D,E} v| ,F,G} th|nh ,D,E,F,G}. Thu được hai cụm {A,B,C},{D,E,F,G}.

Gộp hai cụm còn lại ta thu được một cụm.

Qu{ trình n|y được mô tả trong hình 6.4. C}y được mô tả trong hình 6.4a, nếu lấy mức theo đường đứt, ta thu được 3 cụm như trong hình 6.4b.



**Hình 6.4**. Phân cụm phân cấp tập S={A,B,C,D,E,F,G} theo phương pháp dưới lên

#### *Các quy tắc liên kết*

Trong c{c thuật to{n trộn, hai cụm *i* và *j* được chọn để trộn l| cặp có "khoảng c{ch" (hoặc giả khoảng c{ch) d( *i* , *j* ) nhỏ nhất. C{c khoảng c{ch n|y được định nghĩa kh{c nhau cho c{c thuật to{n v| kết quả ph}n cụm kh{c nhau.



Sau đ}y l| một số quy tắc liên kết.

1. *Liên kết đơn**(*Nearest neighbour*).* Còn gọi l| quy tắc l{ng giềng gần nhất v| được được ký hiệu l| NN*.* Trong quy tắc n|y khoảng c{ch *d*() của hai cụm là khoảng c{ch nhỏ nhất của hai mẫu tương ứng thuộc chúng:

*i*



,

*j*



*i*



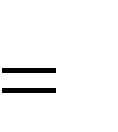
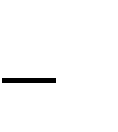
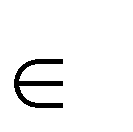
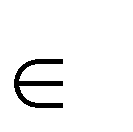
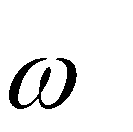
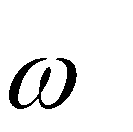
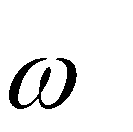
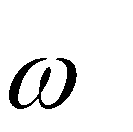
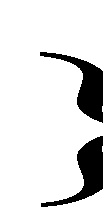
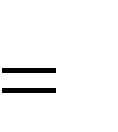
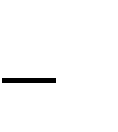
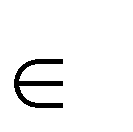
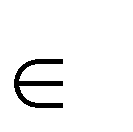
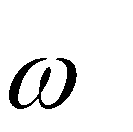
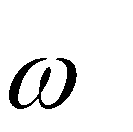
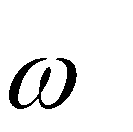
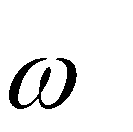
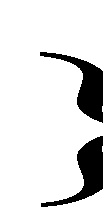
,

*j*



*d*( *i* , *j* ) min **x y** :**x** *i* ,**y** *j* (6.8a)

1. *Liên kết đầy (*Furthest neighbour)*.* Còn gọi l| phương ph{p l{ng giềng xa nhất v| được ký hiệu l| FN, trong đó khoảng c{ch *d*( *i* , *j* ) của hai cụm l| khoảng c{ch lớn nhất của hai mẫu tương ứng thuộc chúng:



*i*



,

*j*



*d*( *i* , *j* ) max **x y** :**x** *i* ,**y** *j* (6.8b)

1. *Liên kết trung bình giữa các nhóm***. (**un-weighted pair-group method using arithmetic averages). Được ký hiệu l| quy tắc *UPGMA,* trong đó khoảng c{ch *d*() của hai cụm l| khoảng c{ch trung bình của c{c cặp mẫu tương ứng thuộc chúng:

*i*



,

*j*



*i*

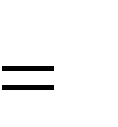
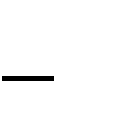
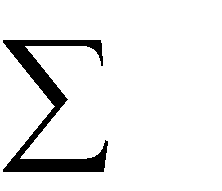
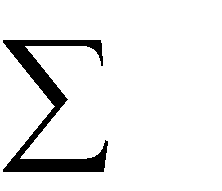


,

*j*



*d*( (6.8c)



*i*

*j*

*j*

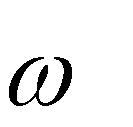
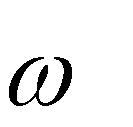
*i*

*j*

*i*

*n*

*n*



**x**

**v**

**y**

**x**

1

)

,

1. *Liên kết trung bình trong phạm vi nhóm.* (un-weighted within-group method using arithmetic averages). Được ký hiệu l| quy tắc *UWGM,* trong đó khoảng c{ch *d*() của hai cụm l| khoảng c{ch trung bình của c{c cặp mẫu thuộc cụm sau khi đã trộn:

*i*



,

*j*



*i*

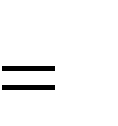
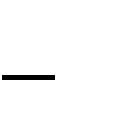
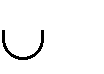
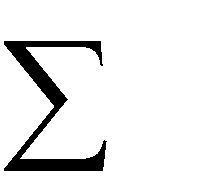


,

*j*



*d*( (6.8d)



*j*

*i*

*j*

*i*

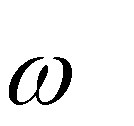
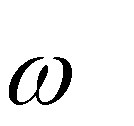
*n*

*n*

*j*

*i*

*C*



**y**

**x**

**y**

**x**

,

2

1

)

,

1. *Phương pháp Ward*. Trong đó khoảng c{ch *d*() của hai cụm là bằng trung bình của bình phương khoảng c{ch của mỗi dữ liệu tới t}m cụm sau khi trộn:

*i*



,

*j*



*i*

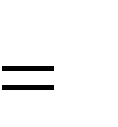
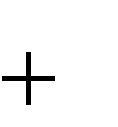
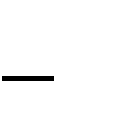
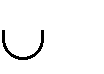
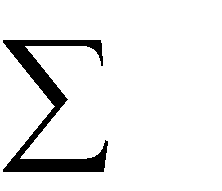


,

*j*



*d*(2(6.8e)



*j*

*i*

*j*

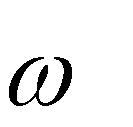
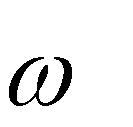
*i*

*j*

*i*

*n*

*n*



**x**

**m**

**x**

1

)

,

##### **6.3.4. Phƣơng pháp phân hoạch**

Ph}n hoạch (Partition) l| phương ph{p ph}n cụm đang được dùng phổ biến nhất, đặc biệt cho c{c tập dữ liệu lớn. Lược đồ chung của c{c thuật to{n n|y như sau: Với tập dữ liệu D gồm *n* đối tượng trong không gian *d* chiều, v| số lượng cụm *k* thường được x{c định trước hoặc đặt dưới dạng tham số, người ta thực hiện một qu{ trình lặp để ph}n c{c đối tượng dữ liệu th|nh *k* cụm cực tiểu mục tiêu chọn trước. Sau đ}y l| một số thuật to{n loại n|y.

**1) Thuật toán k-means**.

*a) Lược đồ tổng quát*

Thuật to{n K-Means (MacQueen, 1967) v| c{c biến thể của nó đang được sử dụng nhiều nhất. Thuật to{n n|y ph}n tập dữ liệu th|nh *k* cụm nhờ dùng gi{ trị trung bình của c{c đối tượng trong một cụm l| t}m cụm nên gọi l| k-mean. H|m mục tiêu được sử dụng trong thuật to{n l| tổng bình phương độ lệch của c{c đối tượng dữ liệu đến t}m cụm:

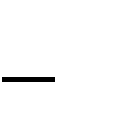
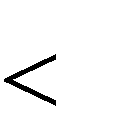
(6.9a)

trong đó . Với số lượng cụm k cho trước, thuật toán này được mô tả trong bảng 6.2.

là tâm cụm

Điều kiện dừng của thuật toán thường chọn một trong các điều kiện sau:

1. *t= Tmax* trong đó *Tmax*là số số lần lặp cho trước
2. *E*  trong đó ∆ l| hằng số bé cho trước



1

*t*

*t*

*E*

1. Tới khi c{c cụm không đổi.

**Bảng 6.2.** Thuật toán k-means

Bước 1. Khởi tạo chọn ngẫu nhiên *k* tâm cum;

Bước 2. Lặp khi điều kiện dừng chưa thỏa mãn

* 1. G{n mỗi đối tượng v|o cụm m| nó gần t}m nhất;
  2. Tính lại c{c t}m cụm của bước 2.1.

Khi tập dữ liệu không qu{ lớn thì người ta dùng điều kiện dừng 3 v| lời giải tìm được l| cực trị địa phương của *E*.

*b) Khi dữ liệu là hỗn hợp các tập tập có phân bố chuẩn nhiều chiều*

Trong thực tế, ta hay gặp tường hợp tập dữ liệu l| hỗn hợp của c{c tập con có ph}n bố chuẩn nhiều chiều, tức l| lớp có phân bố . Khi đó khoảng c{ch của mỗi điểm đến t}m cụm của cụm được tính theo công thức:

(6.9b)

trong đó, c{c ma trận hiệp phương sai được ước lượng theo công thức (6.4h) từ dữ liệu huấn luyện của mỗi lớp nếu có trước tập mẫu cho mỗi lớp.

Thuật to{n k-mean có độ phức tạp l| *O*(*nkt*) trong đó *t* l| số lần lặp trong bước

2. Vì độ phức tạp của thuật to{n thấp nên rất thích hợp khi phải xử lý dữ liệu lớn. Tuy nhiên, thuật to{n có c{c nhược điểm của thuật to{n k-mean cơ bản sau:

1. Nếu t}m khởi tạo không tốt thì có cụm rỗng (hay t}m chết) vì trong mọi lần lặp ở bước 2 không có đối tượng n|o gần t}m n|y.
2. Khi dữ liệu có nhiễu thì có thể l|m lệch t}m nên thuật to{n n|y không ổn định.
3. Thường thì thuật to{n chỉ cho lời giải l| cực trị địa phương của E.

Để khắc phục nhược điểm thứ 3, người ta thường khởi tạo nhiều bộ t}m để chạy thuật to{n nhiều lần v| chọn lời giải có *E* nhỏ nhất. Hai nhược điểm đầu có thể khắc phục bằng thuật to{n k-centroid dưới đ}y .

#### 2) Thuật toán phân cụm k-centroid

Thuật to{n n|y khởi tạo c{c t}m cụm l| c{c đối tượng chọn ngẫu nhiên trong tập dữ liệu v| thay cho việc lấy gi{ trị trung bình l|m t}m, nó dùng một đối tượng trong một cụm gần t}m nhất l|m t}m cụm. Điều n|y l|m cho thuật to{n K-centroid tránh được ảnh hưởng của nhiễu v| phần tử ngoại lai. Với *k* cho trước v| điều kiện dừng như thuật to{n *k*-mean, thuật to{n *k*-centroid thực hiện tương tự như *k*-mean v| được mô tả trong bảng 6.3.

Bảng 6.3. Thuật toán k- centroid

Bước 1. Khởi tạo ngẫu nhiên k đối tượng trong D l|m t}m cụm; Bước 2. Lặp khi điều kiện dừng chưa thỏa mãn:

2.1 G{n mỗi đối tượng v|o cụm m| nó gần t}m nhất;

2.2. Tính trung bình c{c cụm mới ở bước 2.1. v| chọn phần tử trong cụm gần nó nhất l|m t}m mới.

Nhờ thay đổi c{ch khởi tạo v| lấy phần tử t}m người ta đưa ra nhiều biến thể của thuật to{n n|y, chẳng hạn: PAM (Partitioning Around Medoids ); CLARA (Clustering Large Applications); CLARANS<.

1. **Thuật toán phân cụm ngƣỡng**.

C{c thuật to{n loại k-mean thường đòi hỏi biết trước số cụm v| cho phép b{n kính của mỗi cụm lớn tùy ý. Thuật to{n ph}n cụm ngưỡng có c{ch tiếp cận kh{c hơn, nó x{c định trước một ngưỡng “b{n kính” cụm và các *tâm/phần tử tâm* được khởi tạo tuần tự bằng c{ch tăng thêm một *tâm/phần tử tâm* mới mỗi khi khoảng c{ch mẫu mới tới t}m gần nhất lớn hơn ngưỡng n|y. Với ngưỡng cho trước v| điều kiện dừng như thuật to{n *k*-mean, thuật to{n thực hiện như trong bảng 6.4.

**Bảng 6.4.** Thuật toán phân cụm ngưỡng

|  |
| --- |
| Bước 1. Chọn thuộc *D* l|m t}m cụm thứ nhất;  Bước 2. , nếu khoảng c{ch tới t}m gần nhất nhỏ hơn ∆ thì  nó thuộc cụm có t}m n|y, ngược lại nó sẽ l| t}m cụm mới; Bước 3. Lặp khi điều kiện dừng chưa thỏa mãn:  3.1. Tính lại t}m cụm như thuật to{n centroid;  2.2. Ph}n lại c{c mẫu v|o cụm gần t}m nhất |

C{c thuật to{n trên dùng cho dữ liệu có c{c thuộc tính nhận gi{ trị thực. Khi dữ liệu có c{c thuộc tính định danh thì ta không tính được gi{ trị trung bình theo cách thông dụng. Thuật to{n *k*-t}m được dùng cho trường hợp n|y.

1. **Thuật toán k-tâm**.

Trong thuật to{n n|y, ta kh{i niệm mode thay cho trọng t}m của mỗi tập dữ liệu có thuộc tính nhận gi{ trị định danh. Xét tập dữ liệu *D* gồm *n* đối tượng trong đó có c{c th|nh phần của thuộc tính là giá trị định danh với mọi còn khi thì nhận gi{ trị thực. Ta gọi dữ liệu như vậy l| *dữ liệu hỗn hợp*.

*Mode của tập dữ liệu*.

Đinh nghĩa mode của tập dữ liệu hỗn hợp, dựa trên kh{i niệm *j*-mode của c{c thuộc tính ( *j d)*.



*Định nghĩa 6.1*. Cho *C* l| tập con của tập dữ liệu hỗn hợp *D*.

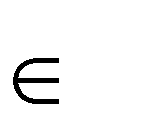
1. Với mọi *j**n* , *j*-mode của *C* ( kí kiệu l| *j*-mode(C)) l| gi{ trị có tần suất nhiều nhất trong tập gi{ trị thuộc tính của C khi l| thuộc tính định danh v| l| trung bình cộng của c{c gi{ trị thuộc tính v|y của *C* khi l| thuộc tính số. Nếu l| thuộc tính định danh v| tập gi{ trị thuộc tính n|y của C có nhiều gi{ trị có tần suất lớn nhất thì *j-*mode(*C*) có thể không duy nhất v| ta chọn gi{ trị n|o cũng được.
2. Mode của tập hợp *C* ký hiệu là mode(*C*) là phần tử z trong đó

= *j*-mode(*C*), *j n* (6.10a)

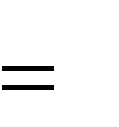
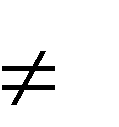


Để tính khoảng c{ch của c{c đối tượng, ta cần định nghĩa mêtric trên dữ liệu hỗn hợp**.** Mêtric n|y được định nghĩa dựa trên mêtric của từng thuộc tính.

*Định nghĩa 6.2.* Giả sử DOM() l| miền gi{ trị của thuộc tính . *x*,*y* DOM() hàm *dj*(*x,y*) x{c định như sau



1. Nếu l| thuộc tính số thì *dj*(*x,y*)= *x y* (6.10b)

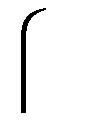
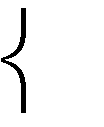
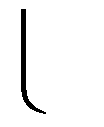
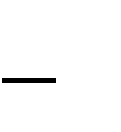


*y*

*y*

*khi*: *x*

1. Nếu j l| thuộc tính định danh thì *dj*(*x,y*)= (6.10c) 1 *khi*: *x*

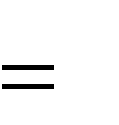
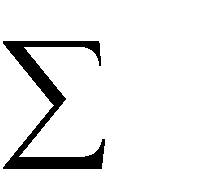


0

B}y giờ ta định nghĩa khoảng c{ch cho mọi cặp đối tượng dữ liệu hỗn hợp.

*Định nghĩa 6.3. Giả sử* ***x*** = (*x1,...,xd*) và **y** = (*y1,...,yd*) l| hai đối tượng dữ liệu hỗn hợp trong *D*, khoảng c{ch *d*(***x,y***) được tính bởi công thức:

*d*(***x,y***) (6.10d)



*d*

*j*

*j*

*j*

*j*

*j*

*y*

*x*

*d*

1

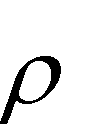
2

2

)

,

(



trong đó c{c *dj*(*x,y*) được tính theo c{c công thức (6.10b-c) và l| c{c trọng số dương cho bởi c{c chuyên gia tuỳ theo mức quan trọng của thuộc tính.

*j*



*Đặc tả thuật toán k-tâm*

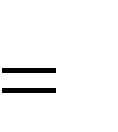
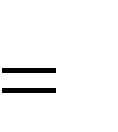
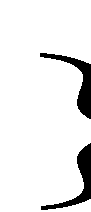
Với định nghĩa Mode v| khoảng c{ch cho tập dữ liệu hỗn hợp đã nêu, thuật toán k-t}m thực hiện như thuật to{n k-means hoặc k-centroid ở trên . Thuật to{n được đặc tả như trong hình 6.5.

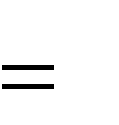
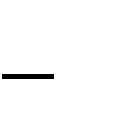
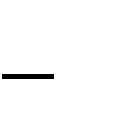
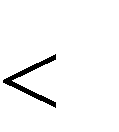
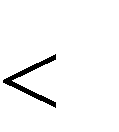
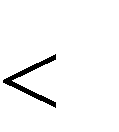
|  |
| --- |
| **Proceduce k-tâm**  **Begin**  Chọn các trọng số , các hàm fj ,xác định k.  *j*  Chọn k phần tử ban đầu *z*  của D làm tâm các cụm  *k*  *j*  *j*  1  Xếp mỗi x D vào cụm Cj mà nó gần tâm nhất;  **For** j=1,...,k **do** *z j* mode(Cj);  **Repeat**  Phân bố lại cụm theo tâm mới// như k-means;  Cập nhật lại tâm cho các cụm // nhờ tính mode  **Until** các cụm không đổi;  Xác định các cụm; **End** |

**Hình 6.5** Đặc tả thuật toán k-tâm

*Xử lý thuộc tính thứ tự*

Nhiều trường hợp, thuộc tính định danh có gi{ trị được sắp thứ tự to|n phần (sắp thẳng), chẳng hạn : DOM(*Aj*)={ không đau, hơi đau, đau v| rất đau}, ta nói thuộc tính n|y l| thuộc tính thứ tự. Khi đó ta có thể {p dụng định nghĩa khoảng c{ch sau cho thuộc tính n|y để tăng chất lượng thuật to{n.

*Định nghĩa 6.4.* Giả sử *Aj* l| thuộc tính thứ tự v| DOM(*Aj*) = *a*1*j* ,...,*akj*  trong đó *a*1*j a*2*j* ... *akj* , ta lấy một h|m đơn điệu tùy ý fj: DOM(Aj)→ [0,1] sao cho *f j* (*a*1*j* )0 *và f j* (*akj* ) 1 (H|m n|y có thể l|: *f j* (*aij* ) *i* 1 ) . Khi đó khoảng c{ch



*k* 1

của hai gi{ trị *x*,*y* trong DOM(*Aj*) được x{c đinh bởi :

*dj*(*x,y*)= │*fj(x)-fj(y*) │. (6.10e)

Công thức n|y được {p dụng trong (6.10d) để tính khoảng c{ch của hai đối tượng dữ liệu. Cũng dựa trên định nghĩa n|y, ta luôn có thể xem c{c thuộc tính thứ tự có miền gi{ trị l| đoạn \*0,1+ để tìm mode (c{c gi{ trị trên thuộc tính n|y của *D* là tập con) v| nó cũng được xem l| thuộc tính số khi không xảy ra nhầm lẫn. Khi cần ta sẽ chuyển ảnh của nó về gi{ trị định danh gốc.

Lưu ý rằng khi xem miền gi{ trị của c{c thuộc tính có thứ tự l| đoạn \*0,1+ v| thì mode(C) cực tiểu h|m:

*d* (*y*,*x*) (6.11)

=



*C*

*y*

2

**Nhận xét**. 1) Khi thuật to{n *k*-t}m kết thúc, c{c đối tượng t}m có thể không thuộc tập *D* do ta đã chuyển đổi gi{ trị của thuộc tính có thứ tự. Để tìm phần tử đại diện cho mỗi cụm, ta lấy phần tử thuộc cụm gần với t}m của nó nhất.

2)Trong thực tế, mỗi thuộc tính số có thể dùng những đơn vị đo kh{c nhau nên việc chọn trọng số thích hợp cho mỗi thuộc tính l| cần thiết.

*i*



##### **6.3.5. Phân cụm bán giám sát**

Như đã nói ở trên, nhược điểm của c{c phương ph{p học không gi{m s{t l| kết quả không chắc chắn, vì vậy khó đ{p ứng đầy đủ yêu cầu người dùng. Để tăng hiệu quả sử dụng cho c{c thuật to{n, một hướng tiếp cận mới l| sử dụng thêm thông tin hợp t{c từ người dùng v| được gọi l| *phân cụm bán giám sát* (semi-supervised clustering). C{c với thông tin bổ trợ n|y thường có được nhờ quan s{t một tập con dữ liệu v| g{n nhãn (ph}n cụm) cho chúng hoặc x{c định được c{c đối tượng n|o thuộc cùng một cụm (*must-link)* hay không thể (*cannot-link)* để cùng một cụm.

Dưới đ}y giới thiệu thuật to{n đơn giản dựa trên thuật to{n *k*-mean gọi l| Seeded-KMeans .

#### Thuật toán Seeded-KMeans

Thuật to{n n|y được Basu v| c{c cộng sự (2002) đề xuất {p dụng cho trường

*k*

hợp khi trong tập ***D*** có tập con ***S*** =*Sh* gọi l| tập giống, trong đó mỗi tập con nên

##### *h*1

để cùng một cụm . Thuật to{n Seeded-KMeans sử dụng thông tin bổ trợ từ tập giống để khởi tạo cho thuật to{n *K*-Means hoặc biến thể của nó. Trong thực tế, nhiều khi không tìm đủ tập giống cho tất cả *k* cụm, khi đó c{c cụm còn lại được khởi tạo ngẫu nhiên trên phần bù của ***S*** trong ***D***. Với tập dữ liệu *D* trong không gian d-chiều

*k*

đã cho, số cụm k v| tập giống ***S*** =*S h* đã biết, thuật to{n Seeded-KMeans hoạt động

##### *h*1

tương tự như *k*-mean v| dễ d|ng mở rộng cho c{c biến thể của nó. Với c{c điều kiện dừng như thuật to{n *k*-mean, thuật to{n n|y được mô tả trong bảng 6.5.

Nhờ thông tin bổ trợ từ tập giống m| thuật to{n Seeded-KMeans có nhiều ưu điểm vượt trội so với thuật to{n K-Means. Kết quả thí nghiệm cho thấy SeededKMeans luôn cho chất lượng xét theo h|m mục tiêu *E cho bởi (6.9a)* v| khả năng kh{ng nhiễu.

**Bảng 6.5.** Thuật toánSeeded-KMeans

Bước 0. Nhập

*D*

,

*k*

,

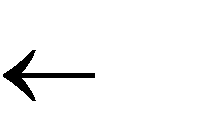
*S*

Bước 1.

Khởi tạo các tâm cụm:

;

*t*



0.

Bước 2. Lặp cho tới khi thỏa mãn điều kiện dừng:

2.1

.

Gán mỗi đối tượng dữ liệu

***x***

vào cụm

*h*

\*

nó gần tâm nhất

;

2.2

.

Ước lượng tâm:

(1)

(

1)

(

1)

1

|

|

*t*

*h*

*t*

*h*

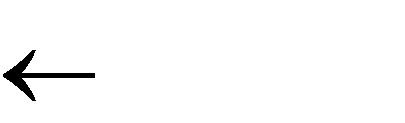
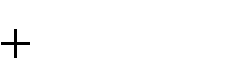
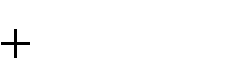
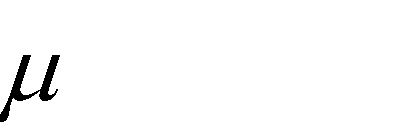
*t*

*xX*

*h*

*x*

*X*



2.3

.

t



t+1

##### **KẾT LUẬN**

Ước lượng h|m mật độ v| ph}n cụm dữ liệu l| hai b|i to{n học không gi{m s{t thường gặp. Có hai c{ch tiếp cận để ước lượng h|m mật độ: kỹ thuật có tham số v| phi tham số. Trong kỹ thuật có tham số, người ta giả thiết tập mẫu được lấy ngẫu nhiên độc lập cùng ph}n bố có h|m mật độ có dạng đã biết p(***x***/***θ*** ) phụ thuộc v|o 

vectơ tham số ***θ*** v| tìm c{ch ước lượng gần đúng vectơ của ***θ*** dựa trên một tiêu chuẩn n|o đó. Có nhiều phương ph{p có tham số, trong đó ước lượng có khả năng nhất l| phương ph{p đơn giản, dễ sử dụng.

Trong thực tế, ta thường không biết được dạng h|m mật độ v| phải ước lượng trực tiếp gi{ trị h|m mật độ tại một tập mốc được xét nhờ kỹ thuật phi tham số. Cửa sổ Parzen v| k-NN là hai phương ph{p thông dụng trong kỹ thuật phi tham số. Cả hai phương ph{p n|y đều dùng một miền nhỏ *R* có thể tích tùy thuộc v|o số mẫu quan s{t được v| x{c định tỷ số điểm thuộc miền trên tổng số dữ liệu để ước lượng mật độ. Phương ph{p k-NN đòi hỏi số điểm thuộc miền *R* đúng với gi{ trị định trước nên thường chính x{c hơn.

Ph}n cụm dữ liệu thuộc loại b|i to{n thiết lập không đúng đắn (ill posed), tuy nhiên hiện có nhiều ứng dụng thực tiễn. Kết quả ph}n cụm phụ thuộc v|o việc chọn mêtric v| đơn vị của thuộc tính. Nhiều trường hợp, chuẩn hóa dữ liệu cho ta kết quả ph}n cụm tin cậy hơn.

Ph}n cụm ph}n cấp v| ph}n hoạch l| hai c{ch tiếp cận thông dụng hiện nay. Ph}n cụm ph}n cấp theo phương ph{p trộn thông dụng hơn phương ph{p trên xuống. Kết quả ph}n cụm ph}n cấp phụ thuộc nhiều v|o quy tắc trộn được chọn. Phải tùy theo đặc điểm b|i to{n m| chọn quy tắc trộn thích hợp.

Phương ph{p ph}n hoạch điển hình nhất l| thuật to{n *k*-mean v| c{c biến thể của nó. C{c thuật to{n n|y thực hiện lặp điều chỉnh cụm nhờ ph}n c{c đối tượng dữ liệu v|o cụm m| nó gần với trung bình cụm/t}m nhất. Thuật to{n cetroid giảm t{c động của nhiễu v| khắc phục được hiện tượng cụm chết của *k*-mean.

Khi có thuộc tính định danh, người ta dùng mode của thuộc tính thay cho gi{ trị trung bình. Nhờ vậy thuật to{n k-t}m dùng được cho dữ liệu hỗn hợp.

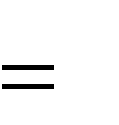
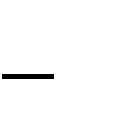
Khi có thông tin bổ trợ từ người dùng về một tập nhỏ dữ liệu, chẳng hạn, nên cùng hay kh{c cụm thì chất lượng ph}n cụm có thể được cải thiện. Trong trường hợp biết được một tập giống gồm c{c tập con nên ph}n cùng một cụm, thuật to{n SeededKmean dùng t}m c{c tập con trong tập giống để khởi tạp t}m cho thuật to{n k-mean cho chất lượng ph}n cụm tốt hơn.

##### **BÀI TẬP**

1. Tạo ngẫu nhiên 100 điểm (xn,yn) trong đó X v| Y có ph}n bố chuẩn N(1,2) v| Y(N(1,3).
   1. Sử dụng kỹ thuật có tham số ước lượng lại ph}n bố của X v| Y.
   2. Dùng c{c kỹ thuật phi tham số để ước lượng gi{ trị p(1,1) v| so s{nh với kết quả phần a.
2. Cho ví dụ về c{c tập dữ liệu trong mặt phẳng để minh họa c{c thuật to{n ph}n cụm ph}n cấp v| ph}n hoạch đã biết.
3. Giả sử *D* có N mẫu v| ph}n hoạch th|nh c cụm *Di ic* 1 (*N > c*) theo thuật to{n kmean nhằm cực tiểu h|m mục tiêu:



*J*  2 .



*i*

*i*

*D*

*D*

*i*

**x**

**m**

**x**

* + 1. Chỉ ra rằng ph}n hoạch cực tiểu *J* không có tập con *Di* n|o rỗng.
    2. Ta luôn có *J(k) > J(k+1*), trong đó *J(k)* l| cực tiểu của *J* khi dùng thuật to{n k-mean chia *D* th|nh k cụm

1. Tìm ví dụ minh họa việc dùng tiêu chuẩn liên kết kh{c nhau trong ph}n cụm ph}n cấp cho kết quả kh{c nhau.
2. Tạo hai tập dữ liệu gồm hai lớp có ph}n chuẩn hai chiều, mỗi lớp 30 dữ liệu, lớp thứ nhất có v| lớp hai có . Áp dụng thuật to{n k-mean với đối tượng khởi tạo gần t}m lớp theo lược đồ tổng qu{t v| khi tính khoảng c{ch có quan t}m đến hiệp phương sai. So s{nh kết quả tìm được.

**Chƣơng 7**

## MẠNG NƠRON NHÂN TẠO

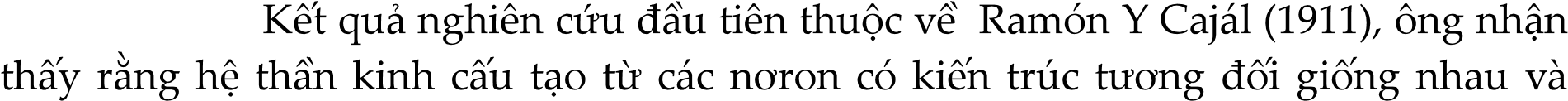
Mạng nơron nh}n tạo (Artificial Neural Network: ANN) mô phỏng kiến trúc v| hoạt động của hệ thần kinh người, đang được nghiên cứu đa dạng v| ứng dụng rộng rãi. Chương n|y c{c mạng học có gi{m s{t cơ bản bao gồm perceptron một tầng v| nhiều tầng (MLP), mạng h|m cơ sở b{n kính (RBF).

### 7.1. GIỚI THIỆU

Kiến trúc v| mô hình hoạt động của mạng nơron nh}n tạo được x}y dựng dựa trên mạng nơron sinh học. Trước khi đi v|o c{c dạng mạng nơron, ta điểm qua c{c đặc điểm chính của hệ thần kinh sinh học.

#### ặc điể

**ạ**

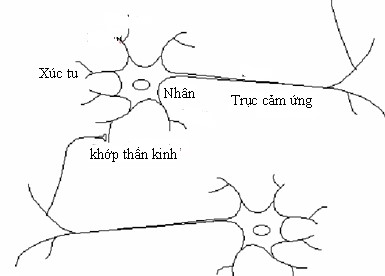
Nghiên cứu về não v| hệ thần kinh được bắt đầu từ đầu thế kỷ 20, v| ph{t triển mạnh mẽ từ giữa thế kỷ n|y tới nay. Ng|y nay, sự ph{t triển của thần kinh học cho chúng ta hiểu biết nhiều về hệ thần kinh nói chung v| cơ chế học của c{c bộ não sinh học nói riêng. 

được kết nối đa dạng với nhau bởi để thực hiện c{c chức năng kh{c biệt. Mô hình cấu trúc của c{c nơron tự nhiên được mô tả trong hình 7.1, mỗi nơron có c{c th|nh phần chính sau:

* *Các khớp kết nối* (Synapse). Chúng kết nối c{c nơron với nhau, nhờ đó m| c{c tín hiệu thần kinh lan truyền được. Cơ chế điều khiển khả năng truyền dẫn ở c{c khớp n|y dựa trên c{c tính chất ho{ lý: Điện Hoá.
* *Các xúc tu* (dendrites). Thu nhận v| truyền thông tin về nh}n qua khớp kết nối.
* *Nhân* hay th}n tế b|o (cell body). Tổng hợp c{c tín hiệu nhận được từ c{c xúc tu v| khi tín hiệu tổng hợp n|y đủ mạnh thì có tín hiệu ra ở trục cảm ứng, lúc

đó ta nói nơron ch{y. Nếu tín hiệu từ nơron A góp phần l|m nơron B ch{y thì ta nói nơron A kích hoạt nơron B.

* *Trục cảm ứng* (Axon). Cuối trục có nhiều đầu ra có khớp kết nối, khi nơron ch{y, nó đưa tín hiệu ra từ nh}n v| truyền tới c{c nơron kh{c qua c{c đầu ra này.



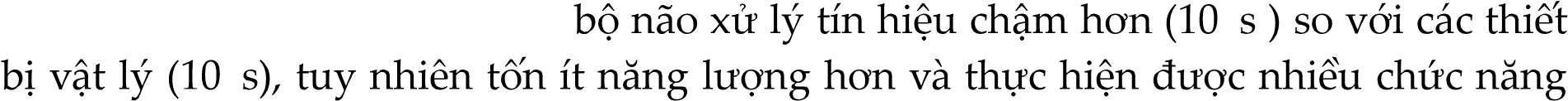
**Hình 7.1**. Mô hình nơron sinh học thể hiện qua 2 nơron

1011 nơron, mỗi nơron có khoảng 104 . Các

khớp khi mới sinh ít kết nối với nhau, chúng được kết nối với nhau nhờ qu{ trình học. Định đề sau của Hebb (1949) về qu{ trình học l| cơ sở cho ý tưởng thiết kế qu{ trình học nhờ điều chỉnh trọng số kết nối của mạng mạng nơron về sau.

*Định đề Hebb*. Khi trục thần kinh cảm ứng của một tế b|o A đủ gần để tham gia kích hoạt tế b|o B v| l|m ch{y nó một c{ch thường xuyên hoặc lặp đi lặp lại thì xảy ra sự chuyển hóa ở một trong hai tế b|o n|y l|m gia tăng t{c động kích hoạt của A lên B. Kết quả n|y của Hebb l|m.

*Đặc điểm xử lý thông tin của não*. Mặc dù đã có nhiều tiến bộ trong thần kinh học, đến nay những hiểu biết chi tiết về cơ chế xử lý thông trong bộ não còn rất hạn chế. Tuy

 nhiên, người ta biết được rằng -3

-9

hơn. C{c tính năng n|y có được nhờ cơ chế xử lý *phi tuyến, song song và phân tán*.

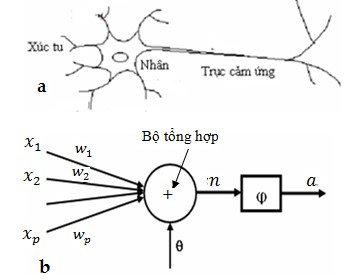
C{c mạng nơron nh}n tạo, về sau sẽ gọi l| *mạng nơron* cho gọn, dựa trên mô phỏng kiến trúc v| c{c đặc điểm qu{ trình học của mạng nơron sinh học.

##### **7.1.2. Mô hình và kiến trúc mạng nơ ron**

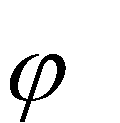
Tương tự mạng nơron sinh học, c{c mạng nơron được x}y dựng nhờ kết nối c{c th|nh phần xử lý thông tin có kiến trúc tương tự nhau được gọi l| nơron.

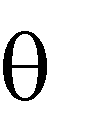
*7.1.2.1. Cấu tạo của nơron*

C{c nơron l| mô phỏng kiến trúc của nơron sinh học, mô hình v| kiến trúc của chúng được mô tả trong hình 7.2, trong đó mỗi nơron l| một đơn vị xử lý thông tin có kiến trúc như ở hình 7.2.b.



**Hình 7.2**. Mô hình nơron (b) mô phỏng nơron tự nhiên (a)

Một nơron bao gồm c{c *liên* *kết* nhận tín hiệu v|o bằng số có c{c trọng số kết nối tương ứng với tín hiệu v|o , gi{ trị ngưỡng , một *bộ* tổng *hợp* v| một *hàm chuyển* (hay *hàm kích* *hoạt*) , để tạo tín hiệu ra dựa trên gi{ trị h|m tổng.



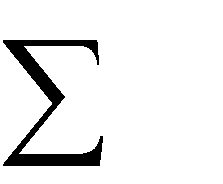
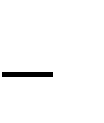
*Liên kết:* Mỗi liên kết thứ *i* sẽ nhận gi{ trị v|o v| có trọng số kết nối tương ứng đưa v|o bộ tổng hợp.

*Trọng số kết nối*. C{c trọng số kết nối (về sau gọi đơn giản l| trọng số khi không g}y nên nhầm lẫn) của mỗi liên kết mô phỏng khả năng truyền tín hiệu qua c{c khớp kết nối của nơron tự nhiên, chính l| yếu tố then chốt của nơron. Phương ph{p x{c định c{c trọng số n|y thể hiện phương ph{p học.

*Bộ tổng hợp.* Bộ n|y tổng hợptích hợp c{c tín hiệu v|o, trọng số kết nối v| gi{ trị ngưỡng để cho ra h|m *n* như l| thông tin tổng hợp từ tín hiệu v|o v| trọng số kết nối.

*k*

Trường hợp đơn giản h|m n|y có dạng: *n*=hoặc *n*= *x w* 2 . Ta sẽ dùng dấu +



*i*

*i*

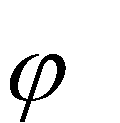
*i*

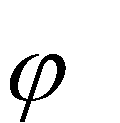
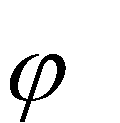
*x*

*w*

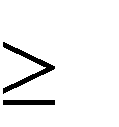
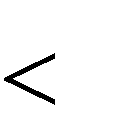
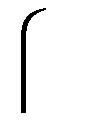
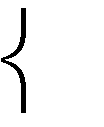
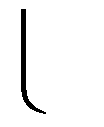
1

để ký hiệu bộ n|y.

*Hàm chuyển / hàm kích hoạt****.*** H|m chuyển nhận gi{ trị v|o l| đầu ra *n* của bộ tổng hợp v| cho gi{ trị ra *a* = (*n*). H|m chuyển l| một h|m không giảm cụ thể n|o đó được chọn tuỳ theo b|i to{n thực tế v| thường phải dễ tính đạo h|m để {p dễ {p dụng khi huấn luyện theo phương ph{p hiệu chỉnh gradient. Sau đ}y l| một số dạng h|m chuyển thông dụng:



**:** *a* = *hardlim*(*n*)=  **;**



0

0

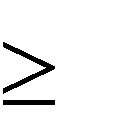
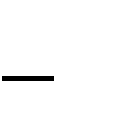
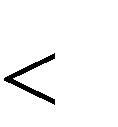
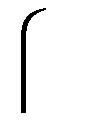
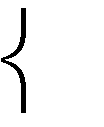
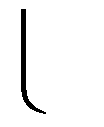
0

1

*s*

*s*

: *a* = hardlims(*n*)= ;



0

1

0

1

*n*

*n*

*:* *a* = purelin(*n*)= *n*

|  |  |
| --- | --- |
|  |  |
|  |
| |  | | --- | | C | |

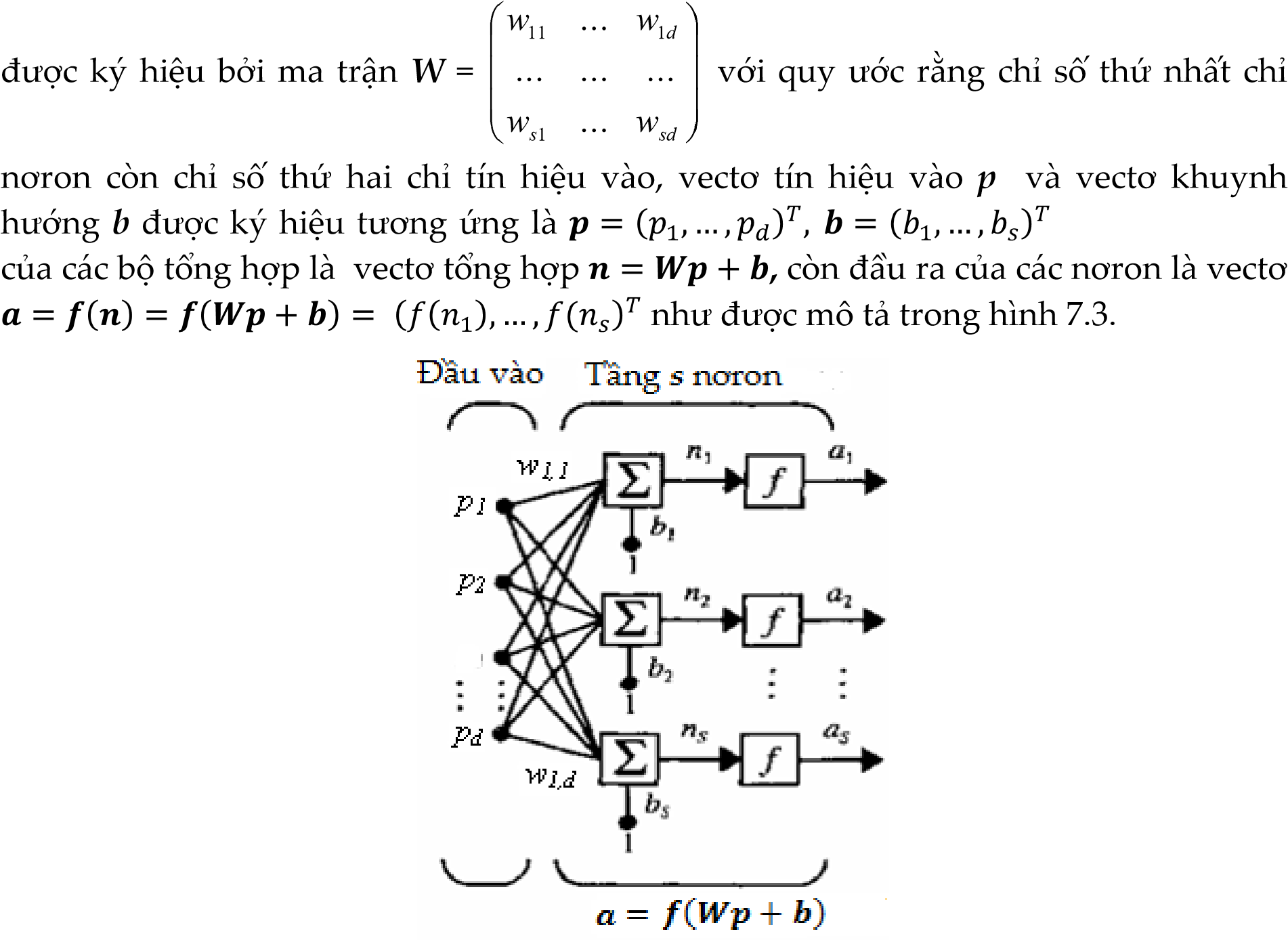
*7.1.2.2. Các kiểu kiến trúc mạng nơron*

Từ c{c nơron đơn, người ta kết nối chúng với c{c tôpô kh{c nhau v| c{ch học c{c trọng số kết nối kh{c nhau để có nhiều nhiều kiểu mạng nơron với công dụng phong phú. Trong đó thông dụng nhất l| c{c mạng truyền tới một hoặc nhiều tầng truyền tới ((Feed-forward), mạng hồi quy (recurrent)<

C{c kiểu kiến trúc mạng v| tính năng của chúng rất phong phú, gi{o trình n|y chỉ tập trung v|o một số mạng truyền tới điển hình, độc giả muốn tìm hiểu s}u hơn có thể tham khảo \*12,13+

1. *Mạng một tầng truyền tới*

Mạng n|y gồm ***s*** nơron chung tín hiệu v|o đấu song song với nhau. C{c trọng số

. Khi đó đầu ra

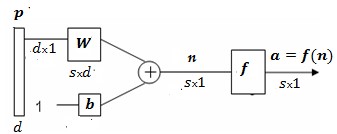
**Hình 7.3.** Mạng nơron một tầng truyền tới

*Biễu diễn rút gọn của mạng nơron một tầng*

Để tiện cho trình b|y, mạng một tầng nơron ở hình 7.3 được biểu diễn dưới dạng rút gọn cho trong hình 7.4.

1. *Mạng nhiều tầng truyền tới*

Mạng nhiều tầng truyền tới l| mạng có hai hoặc nhiều tầng nơron. Trong đó c{c tín hiệu v|o đi qua c{c tầng nơron tới tầng ra m| không quay lại tầng trước hoặc tầng chứa nó nên gọi l| mạng truyền tới. Nếu mạng có *n* tầng nơron thì ta nói mạng (*n+1*) tầng hoặc nói rõ có *n* tầng nơron khi cần nhấn mạnh số tầng nơron, c{c tín hiệu v|o sẽ được gọi l| tầng v|o.



**Hình 7.4**. Biễu diễn rút gọn của một nơron

Một mạng nơron 4 tầng được minh họa trong hình 7.5, trong đó tầng tầng thứ

có

nơron, c{c đầu ra

và

tương ứng l|:

;

**;**

(7.1a)

**.**

(7.1b)

**Hình 7.5.**

M

ộ

t m

ạ

ng truy

ề

n t

ớ

i 4 t

ầ

ng, t

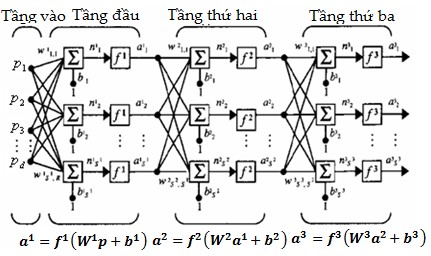
ầ

ng th

ứ

có

nơron



1. *Mạng hồi quy*

Mạng hồi quy kh{c với c{c mạng truyền tới ở chỗ c{c tín hiệu ra ở mỗi tầng nơron có thể quay lại l|m đầu v|o cho nó hoặc c{c nơron ở tầng trước.

**ủ**

Nhờ c{c phương ph{p kết nối nơron v| c{ch x{c định trọng số kết nối kh{c nhau, người ta tạo được c{c mạng nơron như l| *các máy/chương trình* có những tính năng tương tự m| mạng sinh học đã có nhờ mô phỏng thể hiện qua c{c điểm:

* + - Tri thức thu nhận được nhờ qu{ trình học. Trong mạng nơron, qu{ trình học thể hiện qua phương ph{p x{c định trọng số kết nối hay ***luật học***.
    - Tính năng có được nhờ kiến trúc mạng v| tính chất kết nối

Trong ứng dụng, mạng nơron có thể để dưới dạng chương trình hoặc c|i đặt dạng cứng hóa cùng với c{c phần cứng như song song hóa, bộ trễ, bộ tích ph}n< v| có c{c đặc điểm chính sau:

* 1. Phi tuyến. Cho phép xử lý phi tuyến:
  2. Cơ chế {nh xạ v|o ra (**x** → d(**x**)) để học có gi{m s{t
  3. Cơ chế thích nghi. Thay đổi tham số nhờ hiệu chỉnh dần để phù hợp với môi trường.
  4. Đ{p ứng theo mẫu đ|o tạo. Được thiết kế không những cung cấp thông tin về mẫu đ|o tạo m| còn cho biết mức tin cậy của nó.
  5. Thông tin theo ngữ cảnh. Tri thức được biểu diễn tuỳ theo trạng th{i v| kiến trúc của mạng.
  6. Cho phép có lỗi (fault tolerance).
  7. Tích hợp khối lượng lớn (Very large scale integrated: VLSI).
  8. Phỏng sinh học. Kiến trúc v| chức năng của mạng được thiết kế dựa trên kiến trúc v| chức năng của mạng nơron sinh học.

*Các bài toán thích hợp với mạng nơron truyền tới*

Mạng nơron nh}n tạo hiện nay l| một c{ch tiếp cận có miền ứng dụng rất rộng bao gồm cả học có gi{m s{t v| không gi{m s{t. Trong đó c{c mạng nơron truyền tới một hoặc nhiều tầng được dùng phổ biết nhất, thích hợp cho c{c b|i to{n có đặc điểm sau.

* + - C{c mẫu được biễu diễn bởi nhiều cặp gi{ trị thuộc tính. H|m đích được x{c định trên c{c mẫu l| vectơ đặc trưng với gi{ trị có tương quan hoặc độc lập.
    - H|m đích có thể nhận gi{ trị rời rạc hoặc thực thực của c{c vectơ có c{c thuộc tính gi{ trị thực hoặc rời rạc.
    - C{c mẫu đ|o tạo có thể có nhiễu (lỗi).
    - Có thể chấp nhận thời gian huấn luyện l}u. So với c}y quyết định thì huấn luyện mạng nơron thường l}u hơn nhiều.
    - Có thể đòi hỏi đ{nh gi{ nhanh gi{ h|m đích học được. Đ}y l| ưu điểm của mạng nơron so với c{c phương ph{p dựa trên mẫu (k-NN)
    - Không bắt buộc người dùng hiểu rõ h|m đích. Khi dùng mạng nổn, người dùng không cần biết v| giải thích gi{ trị trọng số đang sử dụng.

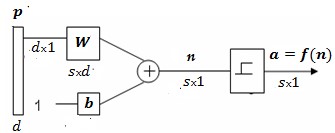
### 7.2. PERCEPTRON

#### 7.2.1. Perceptron của Roseblatt

McCulloch v| Pitts (1943) l| những người đầu tiên đưa ra mô hình của mạng nơron như l| tổng có trọng số của c{c tín hiệu v|o. Rosenblatt (1958) đề xuất mạng nơron để nhận dạng mẫu với luật học perceptron v| chứng minh rằng luật học n|y hội tụ khi tập mẫu t{ch được tuyến tính. Ng|y nay, perceptron dùng để chỉ c{c nơron đơn dùng để nhận dạng mẫu với h|m tổng hợp tuyến tính v| h|m kích hoạt l| h|m ngưỡng hoặc tuyến tính. Tuy nhiên, tùy theo ngữ cảnh m| ta có thể hiểu perceptron l| mạng nơron do Rosenblatt đề xuất để ph}n biệt mẫu.

*Kiến trúc của perceptron*

Một perceptron thường được hiểu l| một mạng nơron đơn, khi ph}n biệt nhiều lớp c{c perceptron có thể kết nối song song với nhau để ph}n biệt tuyến tính dữ liệu (tuyệt đối hoặc từng cặp), lúc đó ta gọi l| mạng perceptron, nếu không xảy ra nhầm lẫn thì cũng có thể gọi gọn l| perceptron. Một mạng perceptron gồm *s* nơron/perceptron với *d* tín hiệu v| với h|m kích hoạt *f* l| h|m ngưỡng *hardlim* được mô tả trong hình 7.6, khi *s =1* ta có một percepron



**Hình 7.6.** Mô hình một mạng perceptron gồm s nơron

*Các phương pháp huấn luyện*

C{c trọng số kết nối của perceptron có thể huấn luyện bằng thuật to{n học perceptron hoặc thuật to{n LMS của Widrow-Hoff trong mục 4.2. chương 4. Dưới đ}y c{c luật học n|y được tóm tắt như sau.

*Luật học**perceptron*

Xét tập huấn luyện gồm hai lớp Khởi tạo vectơ trọng số v| gi{ trị ngưỡng tùy ý, thực hiện lặp thủ tục sau.

Lấy các mẫu ***p*** trong ***D*** (tuần tự/ ngẫu nhiên), tính và *e*(*p*)= *t*(***p***)- *a(p).* Điều chỉnh trọng số theo công thức:

(7.2a)

***;***

Thuật to{n dừng khi không có mẫu dữ liệu n|o cho sai số *e(****p****)* khác không.

Đối với mạng perceptron, công thức tổng qu{t cho điều chỉnh trọng số trong mỗi bước lặp là:

***W****mới =* ***W****cũ+****epT ;*** (7.2b) ***b****mới =* ***b****cũ+****e*.** (7.2c)

trong đó ***b*** l| vectơ gi{ trị ngưỡng, ***e***= ***t***(***p***) – ***a***(***p***), ***t*** l| vectơ gi{ trị đích của mẫu v| a l| đầu ra của mạng.

Như đã nói trong chương 4, luật học perceptron chỉ hội tụ khi tập mẫu quan s{t được l| t{ch được tuyến tính, luật học Widrow-Hoff không đòi hỏi điều kiện n|y.

*Luật học Widrow-Hoff (*thường dùng h|m kích hoạt l| harlims thay cho harlim*)*

Luật học n|y cũng khởi tạo gi{ trị c{c trọng số v| ngưỡng ban đầu tùy ý v| thực hiện lặp lấy ngẫu nhiên hoặc tuần tự c{c dữ liệu quan s{t ***p*** để điều chỉnh trọng số theo công thức:

***;*** , (7.3a)

trong đó l| tốc độ học thỏa mãn điều kiện 0< α <1/ λmax với λmax l| gi{ trị riêng lớn nhất của ma trận tương quan mẫu



[ ***x***

*R =E*(***xx’***) *=* 1 *N k(****x****k)’*]*.*  (7.3b)



##### *N i* 1

Chú ý rằng, cũng có thể dùng h|m ngưỡng đối xứng *hardlims* l|m h|m kích hoạt của mạng perceptron; tuy nhiên khi đó cần lưu ý khi dùng quy tắc học Roselblatt.

Độc giả có thể viết công thức tổng qu{t cho mạng perceptron v| ước lượng khả năng ph}n biệt của mạng như l| b|i tập.

#### 7.2.2. Mạng ADALINE

Mạng ADALINE có kiến trúc như perceptron nhưng dùng h|m kích hoạt tuyến tính như trong hình 7.7.

Phương ph{p LMS thường được dùng để huấn luyện mạng v| có thể dùng thuật to{n Widrow-Hoff hoặc tìm trực tiếp cực trị sai số trung bình phương như sau.

Ta trở lại với tập mẫu quan sát *D* = trong đó ***p****i* và ***t****i* tương ứng là mẫu và tín hiệu ra thứ *i* tương ứng. Không giảm tổng quát, ta chỉ xét một nơron, khi có nhiều nơron thì có thể xem là nhiều nơron được huấn luyện độc lập.



*i*

*i*

*Q*

*i*

*t*

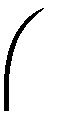
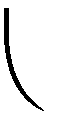
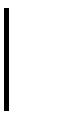
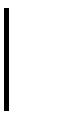
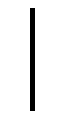
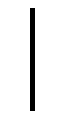
,

1

**p**



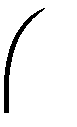
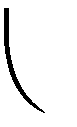
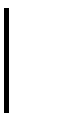
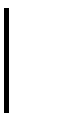
Ta dùng các ký hiệu: ***x***= và ***z***=, khi đó đầu ra *a* của mạng sẽ là:



*b*

*T*

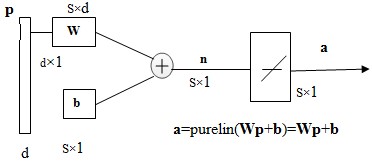
**w**



1

**p**

*a* = ***w****T****p*+***b* = ***x****T****z*.** (7.4a)



**Hình 7.7.** Một mạng ADALINE gồm S noron và p tín hiệu vào

Ở đ}y xem khuynh hướng là thành phần có đầu vào bằng 1.

Biễu diễn sai số trung bình phương l| một hàm *F* của ***x*:**

*F(****x****) = E(e2) = E(t-a)2= E(t-* ***x****T****z****).2* (7.4b)

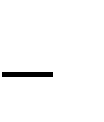
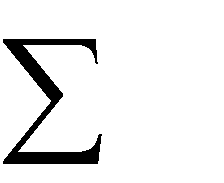
trong đó *E*(.) là ký kỳ vọng của biến ngẫu nhiên tưong ứng. Ta xấp *F*(***x***) bằng trung bình mẫu:

*F(****x****)*=*ai* )2 . (7.4c)

1

(

1



*Q*

*i*

*i*

*t*

*n*

Khai triển công thức (7.4b):

*E(t-a)2 = E(t2-2t****x****T****z+x****T****zz****T****x****) = E(t2)-2****x****TE(t****z****)+* ***x****TE(****zz****T)****x*.** (7.4d)

Đặt *c= E(t2);* ***h*** *= E(t****z****)* cho tương quan chập của tín hiệu vào và tín hiệu đích kết hợpvà **R=**E(**zz**T) cho ma trận tương quan của tín hiệu vào, ta có:

*F*(**x**)= c - 2***x****T****h***+***x****T****Rx*.** (7.4f)

Flà một dạng bậc 2 :

*F*(***x***)**=** *c*+***d****T****x*+** ***x****T****Ax*** (7.4g)

trong đó: ***d***= -*2****h*** và ***A*** *= 2****R***.

Ma trận Hessian của *F* là 2***R*** v| luôn x{c định dương hoặc không âm nên F luôn có cực tiểu toàn cục. Ta có gradient của *F* là:

F(x)= ( c+**d**T**x+**  **x**T**Ax**)= d+**Ax**=-2h+2**Rx** (7.4h)

Điểm dừng của F l| điểm mà gradient triệt tiêu:

*-2****h****+2****Rx*** = 0. (7.4i)

Khi ***R*** x{c định dương thì *F* có duy nhất cực tiểu toàn cục :

***x****\*=****R****-1****h*** (7.4h)

Khi không gian mẫu có số chiều lớn thì khó tìm gần đúng ma trận nghịch đảo của ***R***, lúc đó phương ph{p Widrow-hof hiệu quả hơn.

#### Bộ lọc thích nghi

Bộ lọc thích nghi (Adaptive filtering) l| một ứng dụng của mạng ADALINE nhờ kết hợp với *bộ trễ* và *đường trễ lặp*.

Bộ trễ được minh họa trong hình 7.8a, trong đó *t* l| biến thời gian,đầu ra ***a***(*t*) được tính từ ***u***(*t*) từ công thức ***a***(*t*) = ***u***(*t-1*) v| khở tạo bằng ***a***(0).

Đường trễ lặp được minh họa trong hình 7.8b, có tín hiệu v|o *y*(*t*), tại thời điểm

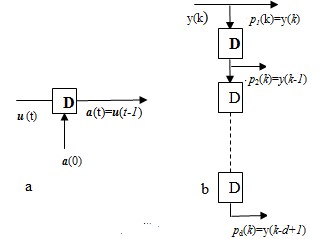
*k*

cho tín hiệu ra l| vectơ

*d*

chiều

.

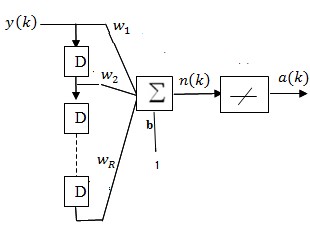


**Hình 7.8**. a) Bộ trễ b) Đường trễ lặp

Bộ lọc thích nghi cho phép hồi quy trễ một h|m số được minh họa trong hình

7.9, trong đó có *R-1* bộ trễ v| đầu ra tại thời điểm *k* là:

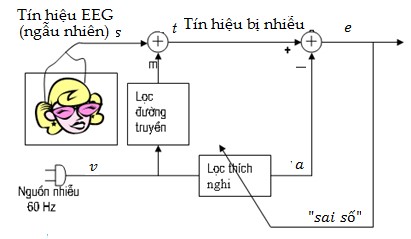
(7.4*l*)



**Hình 7.9.** Bộ lọc thích nghi có R-1 bộ trễ

*Ví dụ ứng dụng*

Bộ lọc thích nghi có nhiều ứng dụng, ta xét một ví dụ sử dụng nó để khử nhiễu. Ta xét bộ đo điện não đồ (EEG) có tín hiệu  *s* bị nhiễu bởi bởi *m* sinh từ nguồn nhiễu *v* có tần số 60 hz nên có tín hiệu ra *t* . Tín hiệu *v* đã được lọc đường truyền nên cho *m* có biên độ giảm 10 lần v| n}ng pha thêm lượng . Người ta dùng bộ lọc thích nghi được c|i đặt để cực tiểu “*sai số “ e* như trong hình 7.10, trong đó nó chỉ dựa trên nguồn nhiễu *v* để ước lượng được đầu ra *a* gần với *m* nhất.



**Hình 7.10**. Hệ khử nhiễu EEG sử dụng mạng ADALINE

Giả sử EEG s có phân bố đều trên đoạn [-0,2; 0,2] và nguồn nhiễu *v*(*k*) cho bởi

*v*(*k*) = *1,2sin*  (7.5a)

,

3

2



*k*

2



*k*



khi đó *m*(*k*) = *0,12 sin*( + ). (7.5b)

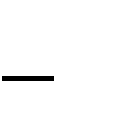
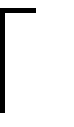
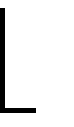
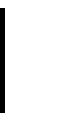
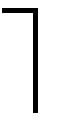
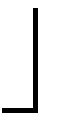
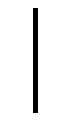
3 2

Ta phân tích toán học cho hệ này. Xét ma trận tương quan mẫu:

***R* =** *E*[***zz****T*]; ***h*** = *E*[*t****z***], (7.6a)

Trong trường hợp này tín hiệu vào của bộ lọc được cho bởi giá trị hiện tại và giá trị trước của nguồn nhiễu:

**z** = , (7.6b)



)

1

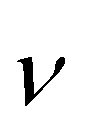
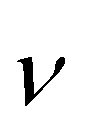
(

)

(

*k*

*k*



còn giá trị đích l| tổng của tín hiệu *s* và tín hiệu nhiễu *m* tại thời điểm đang xét:

*t*(*k*)= s(*k*) + *m*(*k*) (7.6c)

Bây giờ ta biễu diễn cụ thể ***R*** và ***h***:

[*v*(*k*) *v*(*k* 1)

***R***

=

E

và

***h***

=

*E*

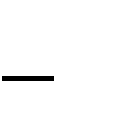
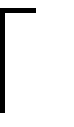
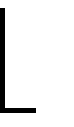
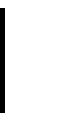
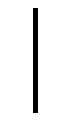
[

*t*

***z***

=

]



)

1

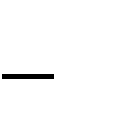
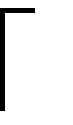
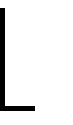
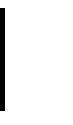
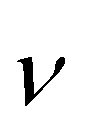
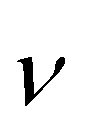
(

)

(

*k*

*k*



))

(

))

1

(

(

))

(

(

2

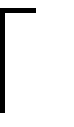
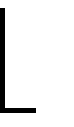
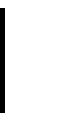
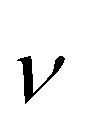
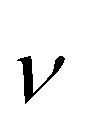
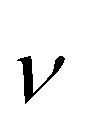
*k*

*k*

*E*

*k*

*E*



*E*( (*k*)) (*k* 1))

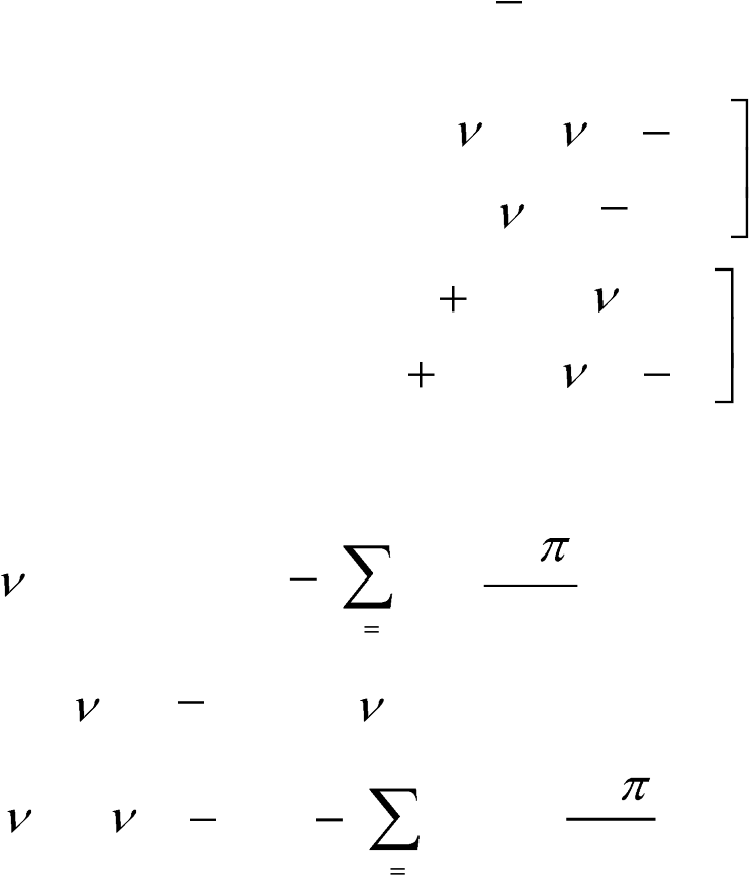
= *E*( 2 (*k* 1)) (7.6d)

*E*(*s*(*k*) *m*(*k*)) (*k*)

(7.6e)

*E*(*s*(*k*) *m*(*k*)) (*k* 1)

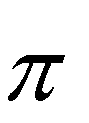
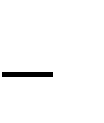
Bây giờ ta tính các phần tử của ***R.***

*E*( 2 (*k*)) =(*1,2*)2 1 3 2*k* 2 *2 0,5 = 0,72* (7.7a) (sin ) =(*1,2)*

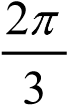
3 *k* 1 3

*E*( 2 (*k* 1)) = *E*( 2 (*k*)) = 0,72 (7.7b)

1 3 2*k* 2(*k* 1)



*E*( (*k*)) (*k* 1)) = 1,2(sin )(1,2sin) 3 *k* 1 3 3

= (1,2)20,5cos( ) = -0,36 (7.7c)

V

ậ

y

**R=**

**.**

(7.7d)

Tương t

ự

ta tính

***h***

. Thành ph

ầ

n th

ứ

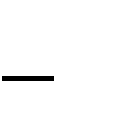
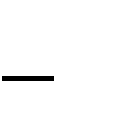
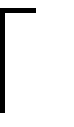
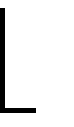
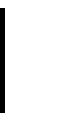
nh

ấ

t c

ủ

a (7.6e):



72

,

0

36

,

0

36

,

0

72

,

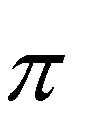
0

(7.7e)

Do *s*(*k*) , *v*(*k*)độc lập và *s* có phân bố đều nên số háng thứ nhất bằng không.

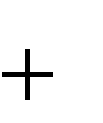
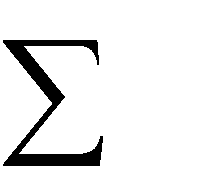
2*k*

*E*(*m*(*k*) (*k*)) = )(1,2sin) =0.



3

1



3

1

2

3

2

sin(

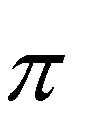
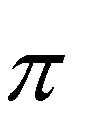
12

,

0

*k*

*k*



3

Như vậy thành phần thứ nhất của ***h*** bằng không.

Thành phần thứ hai của (7.6e):

 *E*([*s*(*k*)*m*(*k*)]*v*(*k* 1))= *E*(*s*(*k*)*v*(*k* 1))+*E*(*m*(*k*)*v*(*k* 1)). (7.7f)

Lập luận như trước, ta có số hạng thứ nhất của (7.7f) triệt tiêu còn số hạng thứ hai:

=

=

-

,0624 =

0

*h*

*2*

.

T

ừ

đó ta có

***h***

=

.

(7.7h)

))

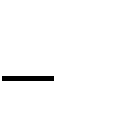
1

(

)

(

(



*k*

*k*

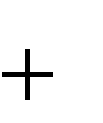
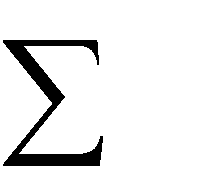
*m*

*E*



3

1



3

1

)

3

)

1

(

2

sin

2

,

1

)(

2

3

2

sin(

12

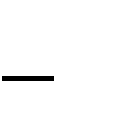
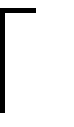
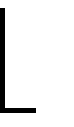
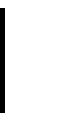
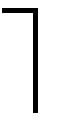
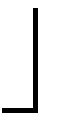
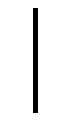
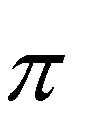
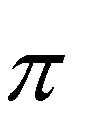
,

0

*k*

*k*

*k*

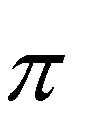
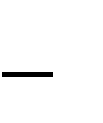


0624

,

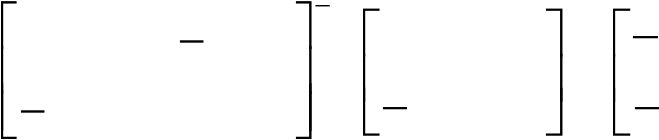
0

0



Chú ý tới biểu thức (7.4h), ta có các trọng số cực tiểu sai số trung bình phương là :

1

 ***x***\* = ***R****-1****h*** = 0,72 0,36 0 = 0,0578

0,36 0,72 0,0624 0,1156

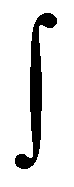
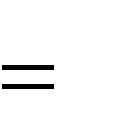
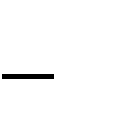
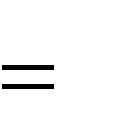
Để hiểu rõ hơn, ta trở lại với biểu thức (7.4f):

*F*(**x**) = *c-2****x****T****h*** *+* ***x****T****Rx* ,** (7.8a)

trong đó ta đã biết ***x*\*, *R*** và ***h***, nay cần tìm *c*. *c* = *E*[*t2*(*k*)]=*E*[*s*(*k*)+*m*(*k*)]2 = *E*[*s*2(k)]+2*E*[s(*k*)*m*(*k*)] +*E*[*m*2(*k*)].

Số hạng thứ hai triệt tiêu vì *s(k)* và *m(k)* độc lập và trung bình bằng không. Số hạng thứ nhất được tính như sau:

1 0,2 2*dS* 1 *S* 3 0,2 0,0133.



*E*[*s2(k*)] = *s*

0,4 0,2 3.0,4 0,2

(7.8b)

Giá trị bình phương trung bình của nhiễu lọc là:

*E*[*m*2(*k*)] = 0,0072. (7.8c)

3

3

2

sin

12

,

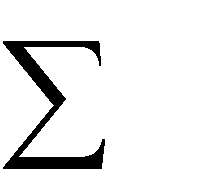
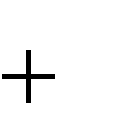
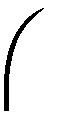
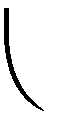
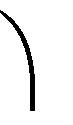
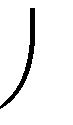
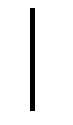
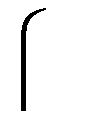
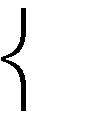
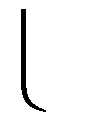
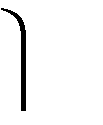
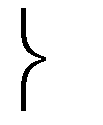
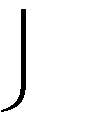
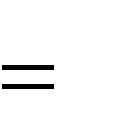
0

3

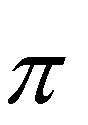
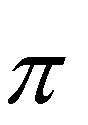
1

2

3



*k*



Vậy *c = 0,0133+0,0072 = 0,0205* (7.8d)

Thay các số hạng đã tính được vào biểu thức (7.8a) ta có:

*F*(***x****\**) = 0,025 - 2.0,0072 + 0,0072 = 0,0133. (7.8e)

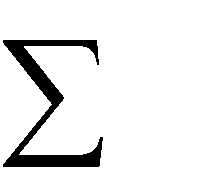
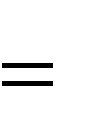
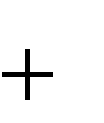
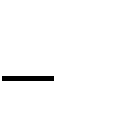
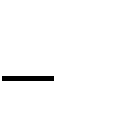
Như vậy lỗi bình phương trung bình giống như gi{ trị bình phương trung bình của tín hiệu EEG.

### 7.3. PERCEPTRON NHIỀU TẦNG (MẠNG MLP)

Perceptrons nhiều tầng thường gọi l| mạng MLP (viết tắt của MultiLayer Perceptrons) l| loại mạng thông dụng nhất để hồi quy h|m nhiều biến.

#### 7.3.1. Kiến trúc mạng

Mạng MLP l| mạng truyền tới nhiều tầng bao gồm c{c nút nhận tín hiệu v|o bằng số, có hai hoặc nhiều tầng nơron, h|m tổng hợp trong c{c nơron đều có dạng: s=*n* , c{c h|m chuyển có thể có dạng kh{c nhau. Một mạng có d tín hiệu v|o v| M tầng nơron thứ *i* có nơron ta nói mạng có kiến trúc *d S*1 *S*2...*SM* . Trong trường hợp n|y ta gọi l| mạng (M+1) tầng hoặc nói rõ mạng M tầng nơron cũng được. Mặc dù có thể có nhiều tầng nơron nhưng trong mạng MLP, ta sẽ gọi tầng tín hiệu v|o l| *tầng vào*, tầng nơron cuối cùng l| tầng ra, còn c{c tầng nơron ở giữa đều gộp chung gọi l| *tầng ẩn*. Hình 7.11 mô tả mạng nơron có kiến trúc 6-3-2 tức l| có 6 nút v|o, 2 tầng nơron với 3 nơron tầng ẩn v| 2 nơron tầng ra.



*k*

*i*

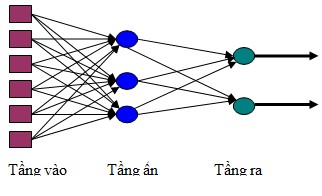
*i*

*i*

*x*

*w*

1



**Hình 7.11.** Kiến trúc mạng nơron truyền tới nhiều tầng

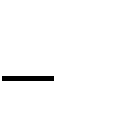
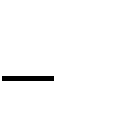
Mạng MLP được dùng rộng rãi để *xấp xỉ/ hồi quy* h|m nhiều biến v| ph}n lớp mẫu, khi dùng để xầp xỉ h|m thì h|m kích hoạt thường l| h|m log\_sig hoặc tanghyperbolic. Một số điểm sau cần lưu ý khi thiết kế mạng này:

* *Tầng vào***.** Nếu h|m đang xét có *n* biến thì tầng n|y có *n+1* nút trong đó nút đầu ứng với gi{ trị x0 = -1v| trọng số l| ngưỡng, mỗi nút còn lại ứng với một biến của đối.
* *Tầng ẩn.* Mạng MLP ba tầng có thể xấp xỉ một h|m liên tục với sai số bé tùy ý nếu có đủ dữ liệu huấn luyện v| số nơron tầng ẩn phù hơp. Tuy nhiên, Việc chọn cấu trúc tầng ẩn thích hợp nhất đến nay vẫn l| b|i to{n mở. Lưu ý rằng nếu số trọng số kết nối qu{ ít so với dữ liệu quan s{t thì sai số lớn, còn nếu qu{ nhiều thì dẫn tới phù hợp trội.
* *Tầng ra.* Mỗi một nơron ở tầng ra sẽ ứng với một h|m, tức l| nếu h|m cần xấp xỉ có gi{ trị ra l| véc tơ *M* chiều thì có *M* nơron ở tầng ra.

#### 7.3.2. Thuật toán huấn luyện lan truyền ngƣợc (BP)

Sau khi đã chọn kiến trúc mạng, người ta thường dùng phương ph{p truyền ngược (Back-Propagation, viết tắt l| BP) sai số để x{c định c{c trọng số kết nối cho mạng MLP nhờ thuật to{n Gradient cực tiểu ho{ sai số trung bình phương.

Xét một MLP có kiến trúc *d S*1 *S*2...*SM* và tập mẫu quan s{t được *D*=



. Để tiện trình bày, trong mục này ta ký hiệu:

(7.9a)

**,** (7.9b)

là ma trận trọng số kết nối của c{c nơron ở tầng thứ là trọng số kết nối thứ *j* của nơron thứ *i* trong này.



*N*

*i*

*i*

*i*

1

,



**t**

**p**

là ma tr

ậ

n c

ấ

p

*m*

và

Ta dùng thuật to{n tương tự của Widrow-Hoff, dùng thuật to{n gradient cực tiểu tổng bình phương sai số hay kỳ vọng mẫu *F*(**x**) của biến ngẫu nhiên **x**:

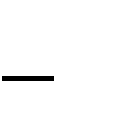
*F*(**x**) *= E*(***eTe***)*= E*((***t****-****a***)*T(****t****-****a***))*,* (7.10a) nếu *SM=1 thì F*(***x***)*=E*[(*t-a)2*]. (7.10b)



Trong lần lặp thứ *k* , kỳ vọng n|y được xấp xỉ bởi *F* (**x**):

*M*

 *F* (**x**) *= (****t****(k)-****a****(k))T(****t****(k)-****a****(k))= ai* (*k*))2 *=****eT****(k)* ***e****(k),*  (7.10c)



*S*

*i*

*i*

*k*

*t*

1

)

(

(

trong đó được tính nhờ vectơ tín hiệu vào lấy ngẫu nhiêu hoặc tuần tự từ tập mẫu *D.* Các trọng số kết nối ở mỗi tầng nơron *m* được điều chỉnh theo công thức :

*w*, (7.10a)

và *bim* (*k*(7.11b)

trong đó là tốc độ học. C{c đạo h|m riêng được tính nhờ công thức đạo hàm hàm hợp của hàm *f* tùy ý :

,

*m*

*j*

*i*

*m*

*j*

*i*

*m*

*j*

*i*

*w*

*F*

*k*

*w*

*k*

,

,

,

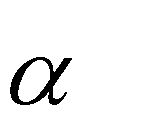
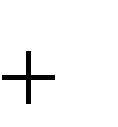
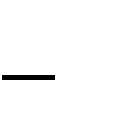
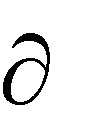
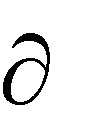
)

(

)

1

(



*i*

*m*

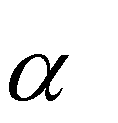
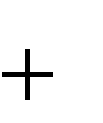
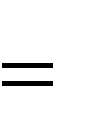
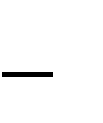
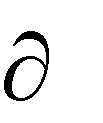
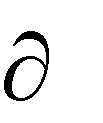
*i*

*b*

*F*

*k*

*b*



)

(

)

1

(7.12a)

,

ta có:

.

B

ở

i vì

*dw*

*dn*

*dn*

*n*

*df*

*dw*

*w*

*n*

*df*

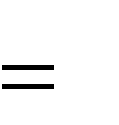
)

(

))

(

(



*m*

*j*

*i*

*m*

*i*

*m*

*i*

*m*

*j*

*i*

*w*

*n*

*n*

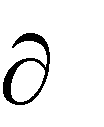
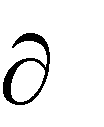
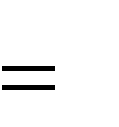
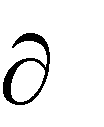
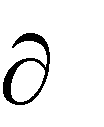
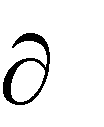
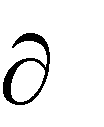
*F*

*w*

*F*

,

,



*m*

*i*

*m*

*i*

*m*

*i*

*m*

*i*

*b*

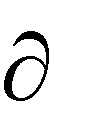
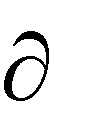
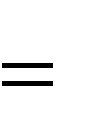
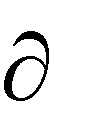
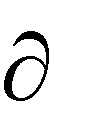
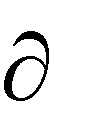
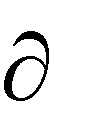
*n*

*n*

*F*

*b*

*F*



(7.12b)

(7.12c)

nên:

(7.12d)

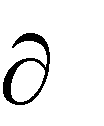
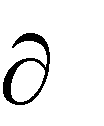
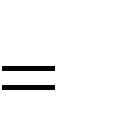
và

.

Nếu ta đặt

1

,



*m*

*j*

*j*

*i*

*m*

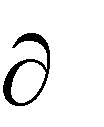
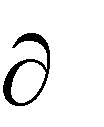
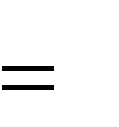
*i*

*a*

*w*

*n*

1



*m*

*i*

*m*

*i*

*b*

*n*

*m*

*i*

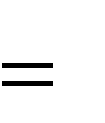
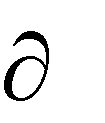
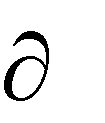
*m*

*i*

*n*

*F*

*S*



(7.12e)

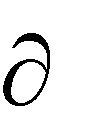
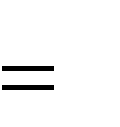
(độ nhạy cảm của *F* đối với sự thay đổi của tín hiệu tổng hợp của nơron *i* ở tầng *m*) thì các biểu thức trong (7.12d) có biễu diễn đơn giản là:

*Sim* . (7.12f)

;

1

,



*m*

*j*

*m*

*i*

*m*

*j*

*i*

*a*

*S*

*w*

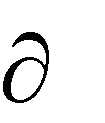
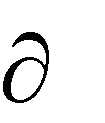
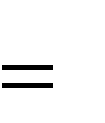
*F*

*m*

*i*

*b*

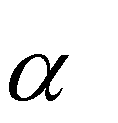
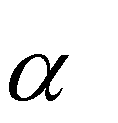
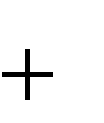
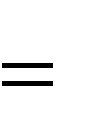
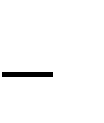
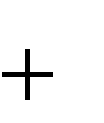
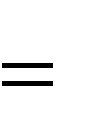
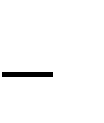
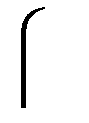
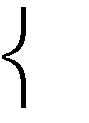
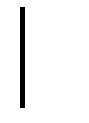
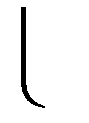
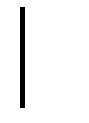
*F*



Như vậy quy tắc hiệu chỉnh trọng số ở theo các công thức (7.10a) và (7.10b) trở thành :

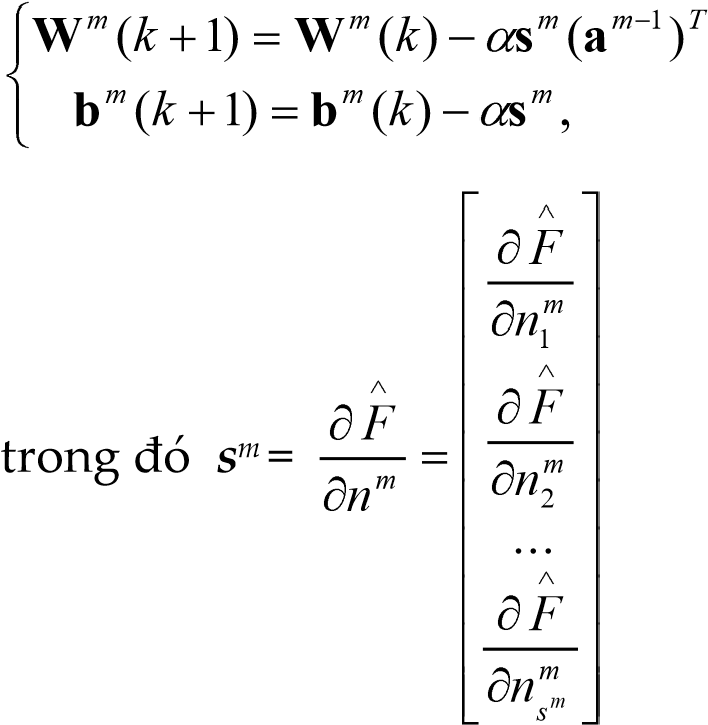
*wim*, *j* (*k* 1) *wim*, *j* (*k*) *Sima mj* 1

*m* (*k* 1) *bmj* (*k*) *Sim* . (7.12g)



*b j*

Biểu diễn dưới dạng ma trận công thức này là:

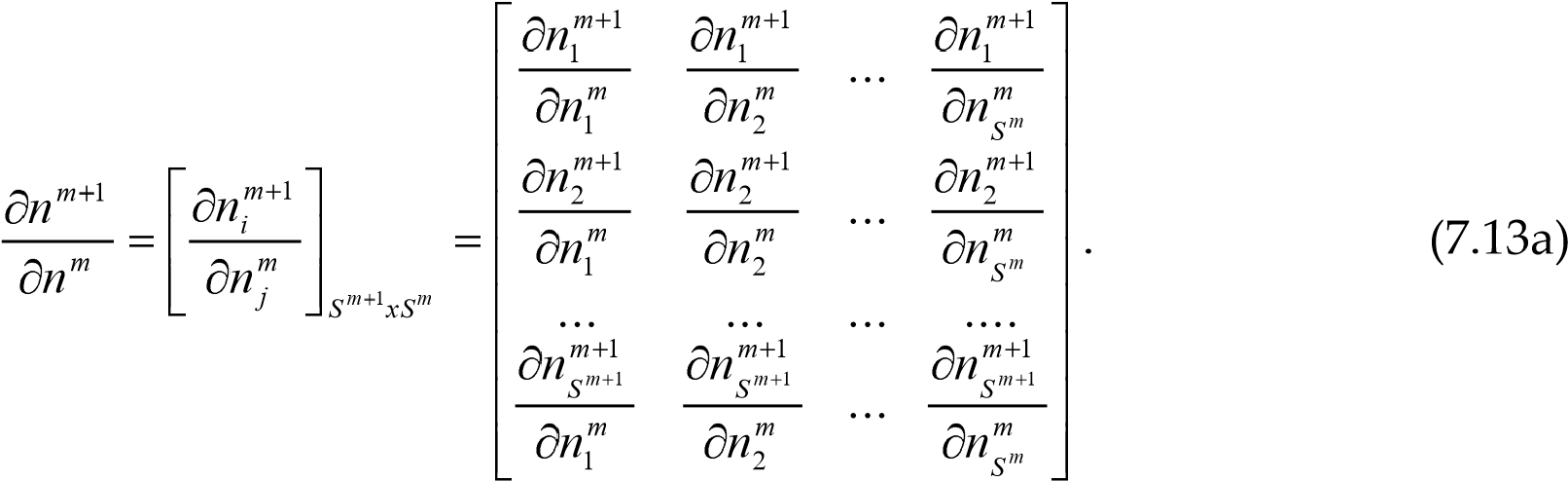
 (7.12h)

(7.12i)

*Hiệu chỉnh lan truyền ngược*

Để {p dụng công thức hiệu chỉnh (7.12h), công việc còn lại l| ước lượng c{c ***s****m. Quá* trình tính sẽ bắt đầu từ tầng ra, lùi dần về tầng nơron thứ nhất, trong đó ***s****m* được tính từ ***s****m* nên có tên gọi *lan truyền ngược*.

Để đưa ra công thức đệ quy, ta dùng ma trận Jacobi cấp *Sm+1 Sm*:



Để tìm biểu diễn cho ma trận n|y, ta xét phần tử (i,j) của ma trận :

 (7.13b)

)

(

)

(

1

,

1

,

*m*

*j*

*m*

*m*

*j*

*i*

*m*

*i*

*m*

*i*

*m*

*m*

*j*

*i*

*n*

*f*

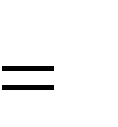
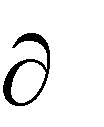
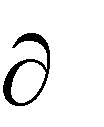
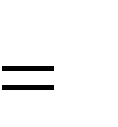
*w*

*n*

*n*

*f*

*w*



.

trong đó

)

(

*m*

*j*

*m*

*n*

*f*

=

*m*

*i*

*m*

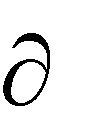
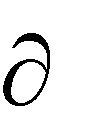
*i*

*m*

*n*

*n*

*f*



)

(

.

(7.13c)

*m*

*j*

*m*

*j*

*m*

*j*

*i*

*m*

*j*

*S*

*j*

*m*

*i*

*m*

*j*

*m*

*j*

*i*

*m*

*j*

*m*

*i*

*n*

*a*

*w*

*n*

*b*

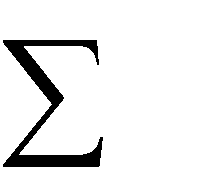
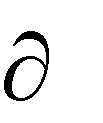
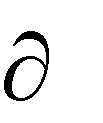
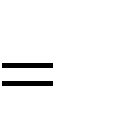
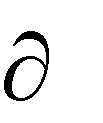
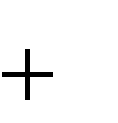
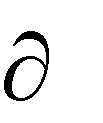
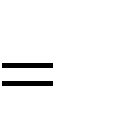
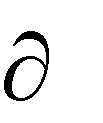
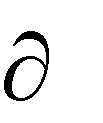
*a*

*w*

*n*

*n*

*m*



1

,

1

1

1

,

1

)

(

Như vậy ma trận Jacobi có thể viết là:

(7.13d)

 ... 0

)

(

1

1

*m*

*m*

*m*

*m*

*m*

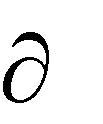
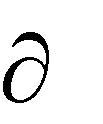
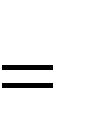
**n**

**F**

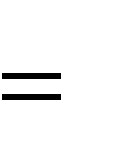
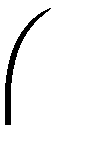
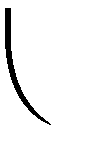
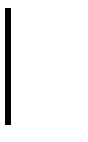
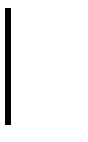
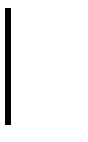
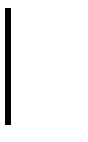
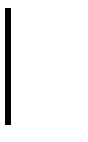
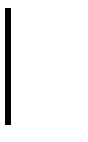
**W**

**n**

**n**



trong đó:



0

0

...

...

)

(

0

0

)

(

2

1

*m*

*m*

*m*

*m*

*m*

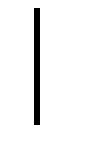
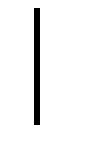
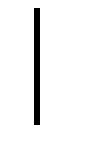
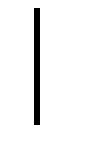
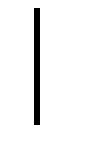
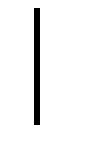
*n*

*f*

*n*

*f*

**F**



... ...

. (7.13e)

... ...

*m*

0 *f*(*nSmm* )

Bây giờ dạng đệ quy cho các ***s****m* được viết theo quan hệ đệ quy nhờ dùng quy tắc dây chuyền dạng ma trận:

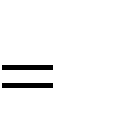
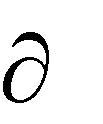
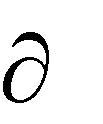
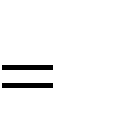
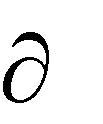
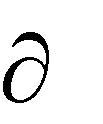
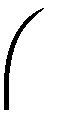
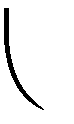
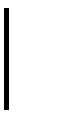
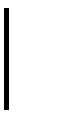
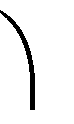
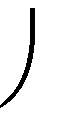
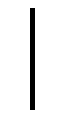
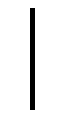
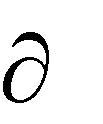
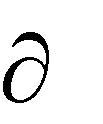
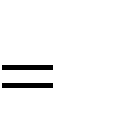
**s***m* **F** (**n***m* )(**W***m* (7.13f)

1

1

ˆ

ˆ



*m*

*m*

*T*

*m*

*m*

*m*

*F*

*F*

**n**

**n**

**n**

**n**

=



1

1

)

(



*m*

*T*

*m*

*m*

*m*

**s**

**W**

**n**

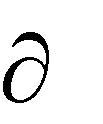
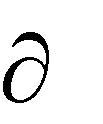
**F**

1

1

ˆ

)



*m*

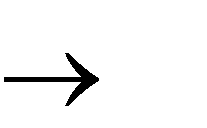
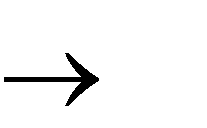
*T*

*F*

**n**

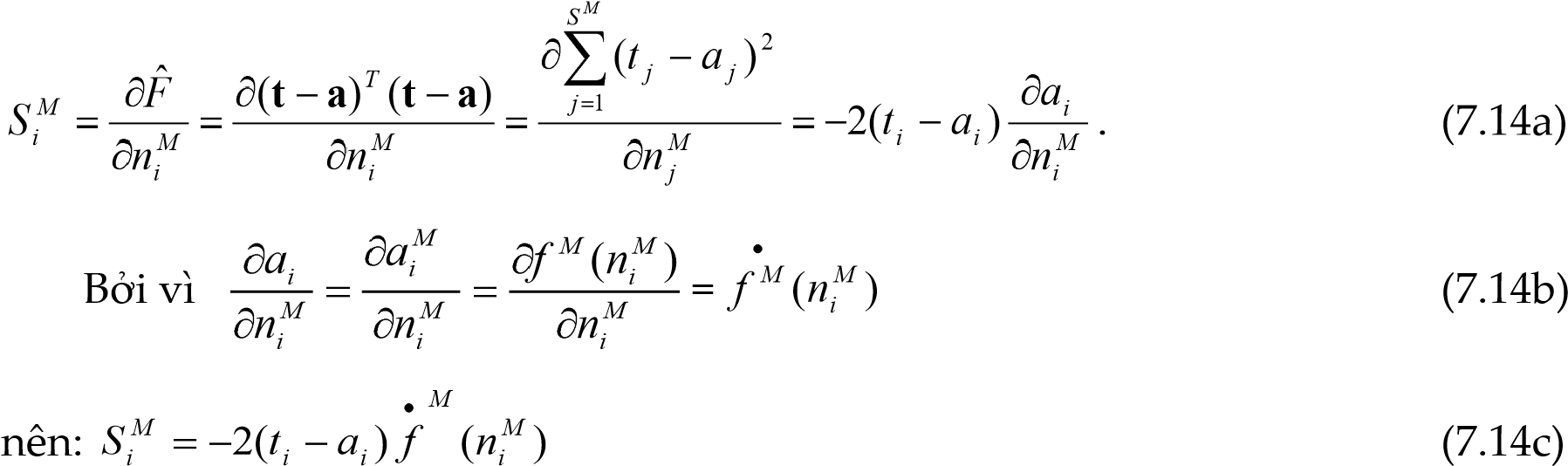
Bây giờ ta thấy rõ sự lan truyền ngược khi tính ***s****m* từ tầng ra tới tầng đầu:

**s***M* **s***M* 1 ...**s**2 **s**1 (7.13g)



Để hoàn thiện thuật toán lan truyền ngược, ta cần tính ***s****M* cho công thức đệ quy

(7.13f):

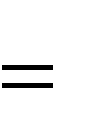
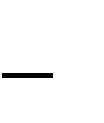


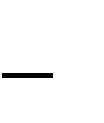
Biểu diễn dưới dạng ma trận:

*M*

2

**F**

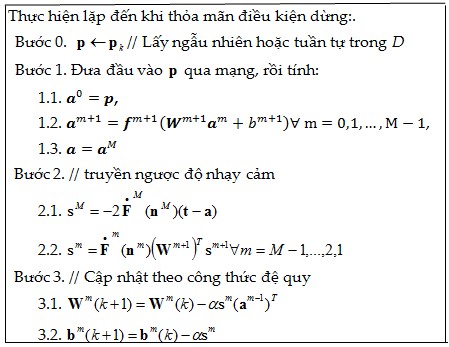


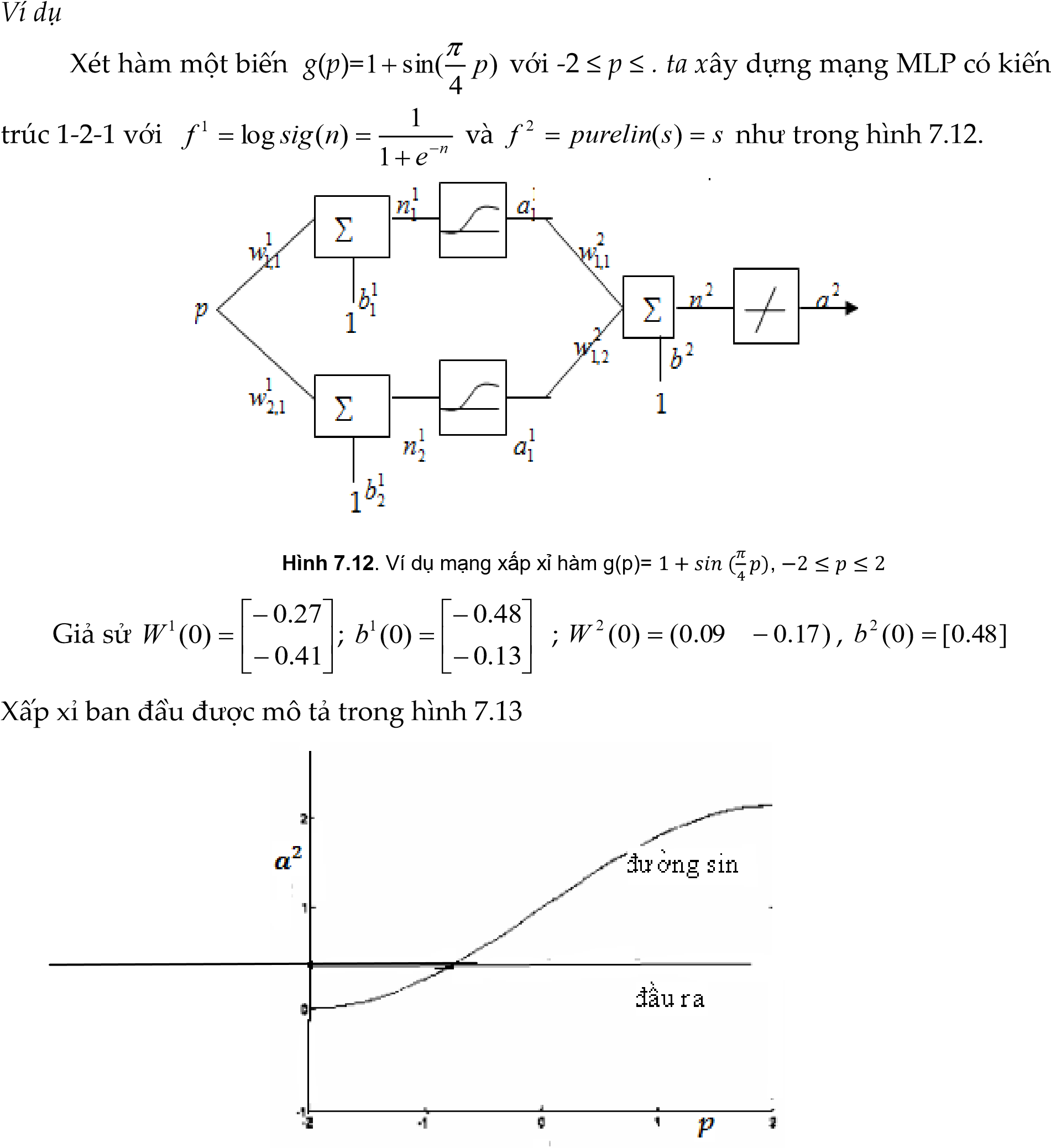
 **s***M*(**n***M* )(**ta**)(7.14e)

Điều kiện dừng của thuật to{n thường là khi các trọng số v| khuynh hướng/giá trị ngưỡng thay đổi không đ{ng kể, tức là nhỏ hơn đủ bé cho trước. C{c trọng số ban đầu v| khuynh hướng được khởi tạo tùy ý. Thủ tục của thuật to{n BP được mô tả trong bảng 7.1.

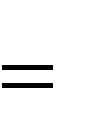
Nhược điểm quan trọng của thuật to{n BP l| thường chỉ cho lời giải gần đúng của cực trị địa phương v| mất nhiều thời gian huấn luyện. Để khắc phục nhược điểm thứ nhất, ta có thể khởi tạo ngẫu nhiên nhiều bộ gi{ trị ban đầu cho c{c trọng số v| gi{ trị khuynh hướng, sau khi huấn luyện thì chọn lời giải cho sai số trung bình phương nhỏ nhất.

**Bảng 7.1**. Thuật toán BP huấn luyện mạng MLP



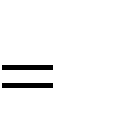
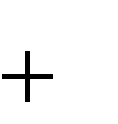
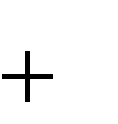
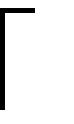
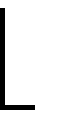
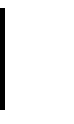
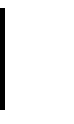
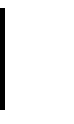
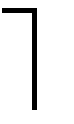
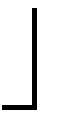
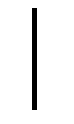
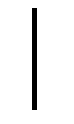
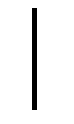
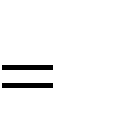
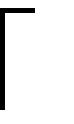
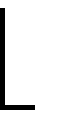
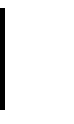


**Hình 7.13**. Đầu ra của mạng với trọng số khởi tạo Qu{ trình thực hiện thuật to{n như sau.

1: *p*=1; *a*0 *p* =1

Đầu ra tầng thứ nhất l|:

##### *a*1 *f* 1 (*W* 1*a* 0



368

.

0

321

.

0

1

1

1

1

54

.

0

75

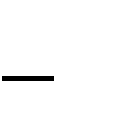
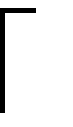
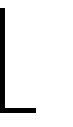
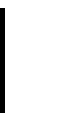
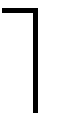
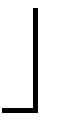
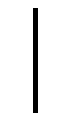
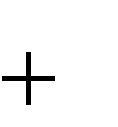
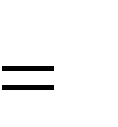
.

0

*e*

*e*

Đầu ra tầng hai l| :



]

446

.

0

[

])

48

.

0

[

368

.

0

321

.

0

]

17

.

0

09

.

0

([

)

(

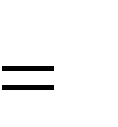
2

1

2

2

2



*purelin*

*b*

*a*

*W*

*f*

*a*

T

sai số

:

*e*

*=*

*t*

*-*

*a =*

261

.

1

446

.

0

)}

1

4

sin(

1

{

)}

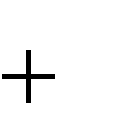
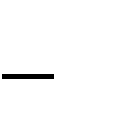
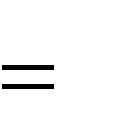
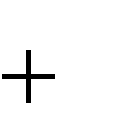
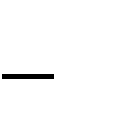
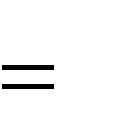
4

sin(

1

{

2



*a*

*p*

)

)(

1

(

)

1

1

)(

1

1

1

(

)

1

(

)

1

1

(

)

(

1

1

2

1

*a*

*a*

*e*

*e*

*e*

*e*

*e*

*dn*

*d*

*n*

*f*

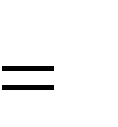
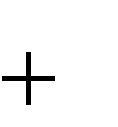
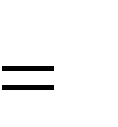
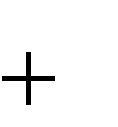
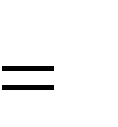
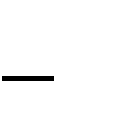
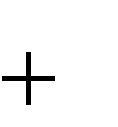
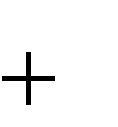
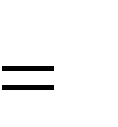
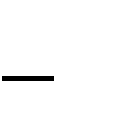
*n*

*n*

*n*

*n*

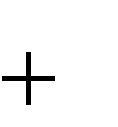
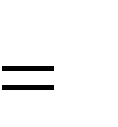
*n*



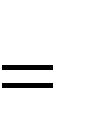
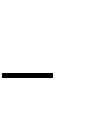
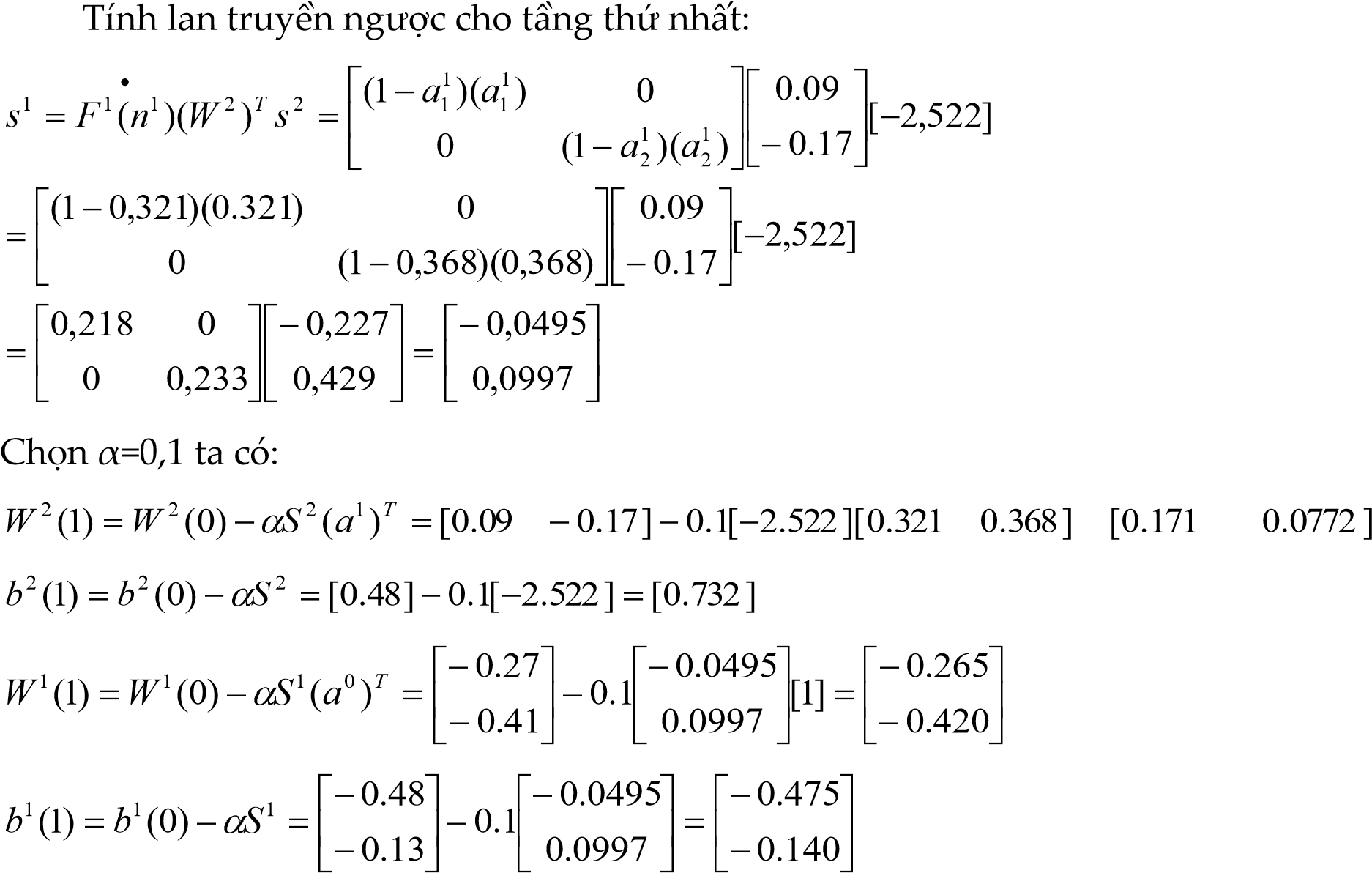
.

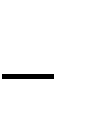
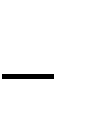
Đối với tầng thứ hai ta có

:

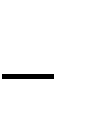


*f*

 Vậy: s2 2*F* (*n* )(*t* 2[ *f* (*n*2 )](1,261) 2[1](1,261) 2,522.



2



1

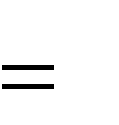
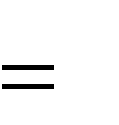
)

(

)

(

2



*n*

*dn*

*d*

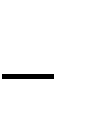
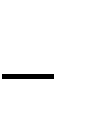
*n*

.

)

2

2



*a*

Như vậy bước lặp thứ nhất của thuật to{n được ho|n th|nh.

### 7.4. MẠNG HÀM CƠ SỞ BÁN KÍNH (MẠNG RBF)

Như đã nói ở trên, mạng MLP l| công cụ hữu hiệu để hồi quy h|m nhiều biến. Tuy nhiên, nhược điểm của nó l| khó tìm được cấu trúc mạng phù hợp, thời gian huấn luyện l}u v| thường chỉ cho cực trị địa phương của sai số trung bình phương. Mạng RBF (Radial Basis Function) l| tiếp cận hiệu quả cho hồi quy h|m số với mục đích nội suy. Mặc dù không dùng được cho mục đích ngoại suy nhưng mạng RBF có ưu điểm l| thời gian huấn luyện nhanh v| c{c thuật to{n luôn cho lời giải hội tụ tới cực tiểu to|n cục duy nhất.

#### 7.4.1. Kiến trúc mạng RBF

Kỹ thuật hồi quy RBF do Powell đề xuất (1987) , được Broomhead v| Lowe (1988) {p dụng cho mạng nơron để hồi quy h|m nhiều biến. Mạng RBF xấp xỉ h|m thực *n* biến f : D ( Rn) Rm l| một mạng 3 tầng truyền tới như mô tả trong hình 7.13. Tầng v|o gồm *n* nút cho vectơ tín hiệu v|o ***x*** *Rn* , tầng ẩn l| tầng h|m cơ sở b{n kính gồm M nơron, dạng thông dụng của đầu ra tầng 1 l| :



0

||

||

2

1

*i*

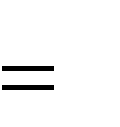
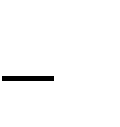
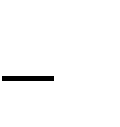
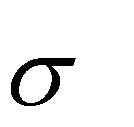
*i*

*i*

*c*

*p*

*n*



và

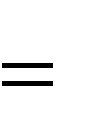
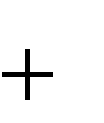
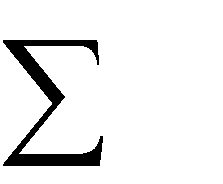
,

v| đầu ra tầng hai l|:

**=**

,

hay:



*M*

*k*

*k*

*k*

*j*

*j*

*w*

*a*

*w*

*a*

1

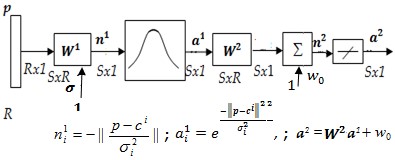
1

2

,

2

.



**Hình 7.14**. Mạng RBF tổng quát

Các hàm l| c{c h|m cơ sở b{n kính dạng Gauss, ***ck*** gọi l| t}m của *k* , tham số (còn gọi l| tham số độ rộng của h|m cơ sở) dùng để điều khiển độ rộng miền ảnh hưởng của h|m cơ sở *k* (khi *x ck* 3 *k* thì *k* (x) gần triệt tiêu) còn gọi l| trọng số của *k* nơron thứ *i* ở tầng ra . Đầu ra thứ của nơron thứ j ở tầng ra được viết lại l|:

)

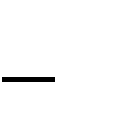
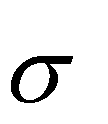
(

*x*

*k*



*k*



(7.15a)

#### 7.4.2. Huấn luyệ

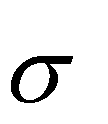
Không giảm tổng quát, ta xét tầng ra chỉ có một nơron, tập dữ liệu quan sát được . Ký hiệu , các được gọi l| trọng số tầng ra. Khi đó đầu ra của mạng l|:



(7.15b)

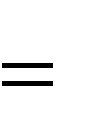
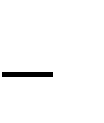
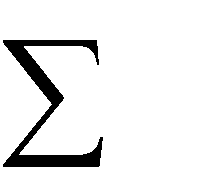
Viêc huấn luyện mạng bao gồm x{c định t}m c{c t}m , tham số độ rộng của mỗi h|m b{n kính *k* , c{c trọng số số . Hai hướng tiếp cận sau thường được dùng để x{c định chúng.

*k*



1. Tìm cực tiểu sai số tổng c{c bình phương (sum-of-squares error: SSE ):

*E y k* )2 (7.16a).



*N*

*k*

*k*

*x*

*a*

1

2

)

(

(

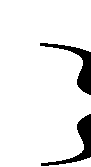
1. Giải đúng hoặc gần đúng hệ phương trình:

(7.16b)

*7.4.2.1. Các thuật toán dựa trên tìm cực tiểu SSE*

C{c thuật to{n n|y được ph}n ba loại: một pha, hai pha v| 3 pha. Trong đó việc x{c định trọng số tầng ra được thực hiện nhờ thuật to{n gradient hoặc một biến thể của nó.

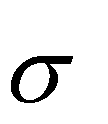
*a) Các thuật toán một pha*

Trong c{c thuật to{n một pha, người ta x{c định trước t}m v| c{c tham số độ rộng còn trọng số tầng ra được x{c định nhờ thuật to{n gradient. Điển hình l| c{c *thuật toán đào tạo nhanh* (Quick Training viết tắt l| QT), trong đó người ta chọn trước c{c t}m cho h|m b{n kính, còn độ rộng có thể lấy theo c{c đề xuất của Looney hoặc Haykin:

The

o Looney:

*k*



=

(7.17a)



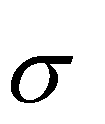
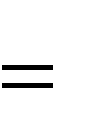
Theo Haykin

*N*

*D*

2

max



,

(7.17b)

trong đó

Trong trường hợp số mốc không nhiều, c{c điểm

thường được chọn l|m

c{c t}m cho c{c h|m b{n kính (M=N) để {p dụng c{c thuật to{n QT.

Lưu ý rằng nếu độ rông của c{c h|m b{n kính được chọn bé thì c{c điểm xa t}m ít chịu ảnh hưởng của h|m c{c h|m cơ sở nên chất lượng xấp xỉ của h|m hồi quy thấp mặc dù thuật to{n hội tụ nhanh. *b) Các thuật toán hai pha*

Trong c{c thuật to{n n|y, c{c t}m v| độ rộng c{c h|m b{n kính được x{c định ở pha thứ nhất, còn pha thứ hai x{c định c{c trọng số tầng ra.

Chẳng hạn, dùng thuật to{n ph}n cụm ngưỡng để ph}n cụm tập v| lấy điểm trung bình của mỗi cụm l|m t}m, c{c độ rộng x{c định để h|m b{n kính tương ứng xấp xỉ đủ tốt dữ liệu quan s{t trên cụm n|y. *c) Các thuật toán ba pha*

Thuật to{n n|y còn gọi l| thuật to{n . Trong đó người ta dùng một thuật to{n hai pha để tìm t}m v| c{c tham số đủ tốt, sau đó để điều chỉnh c{c độ rộng b{n kính v| trọng số tầng ra nhờ dùng một biến thể của phương ph{p gradient.

Dễ thấy rằng phương ph{p n|y đòi hỏi nhiều thời gian huấn luyện nhưng chất lượng mạng tốt hơn c{c phương ph{p trên. Độc giả có thể tham khảo mô tả chi tiết của một thuật to{n loại n|y trong [15]

**Chú ý**. Trong c{c thuật to{n kể trên, thay vì tìm cực tiểu SSE bằng phương ph{p gradient, người ta có thể tìm trọng số tầng ra bằng phương ph{p giả nghịch đảo để giải hệ (7.16b). Tuy nhiên, khi số mốc lớn thì lời giải tìm được cho sai số lớn nên không đ{ng tin cậy.

Dưới đ}y giới thiệu phương ph{p hai pha huấn luyện lặp mạng RBF nội suy dễ sử dụng, thời gian huấn luyện nhanh v| cho chất lượng mạng kh{ tốt khi khoảng cách giữa c{c mốc nội suy ph}n bố gần với lưới đều. Khi c{c mốc nội suy c{ch đều, ta có thể x{c định trước c{c độ rộng b{n kính v| thuật to{n hai pha trở th|nh thuật to{n một pha. Thuật to{n n|y kết hợp với hồi quy k-k-NN trong chương 8 cho ta một phương ph{p hiệu quả để x}y dựng mạng RBF hồi quy.

*7.4.2.2. Phương pháp huấn luyện lặp mạng nội suy*

Ta trở lại bài toán nội suy với các mốc nội suy cho bởi tập dữ liệu quan sát nêu trên. Các mốc nội suy được dùng làm tâm cho các hàm bán kính.

*Phân tích toán học*

Thuật to{n được x}y dựng dựa trên nhận xét hệ phương trình tuyến tính (7.16b) với ẩn l| c{c trọng số tầng ra thì c{c hệ số của chúng có đặc điểm:

(7.18a)

, (7.18b)

l| h|m đơn điệu giảm của với mọi

hơn nữa,

Đặc điểm n|y cho phép ta chọn được c{c thích hợp để hệ phương trình tuyến tính (7.16b) có dạng đường chéo trội v| trọng số tầng ra được tìm nhờ phép lặp đơn. Cụ thể như sau.

Ký hiệu I l| ma trận đơn vị cấp *N* ; l| c{c vec tơ trong không

,

gian *N*-chiều *RN* trong đó:

(7.19a) v| đặt . (7. 19b) Ta có (7. 19c)

*z*

*k*

=

*y*

*k*

*-*

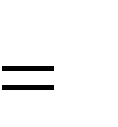
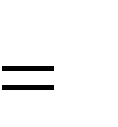
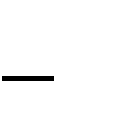
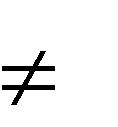
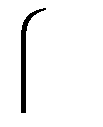
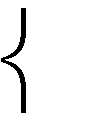
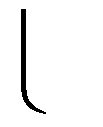
*w*

*0*

,

*N*

*k*



*j*

*k*

*khi*

*e*

*j*

*k*

*khi*

*k*

*k*

*i*

*x*

*x*

*j*

*k*

:

;

:

;

0

2

2

/

||

||

,



,

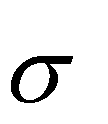
hệ phương trình nội suy(7.16b) tương đương với hệ :

(7.20a)

thì với c{c tham số

Nếu đã chọn v| w0tùy ý, (7.9) luôn có duy nhất nghiệm **W**. Về sau ta sẽ lấy *w0*l| trung bình cộng của c{c gi{ trị *yk*:

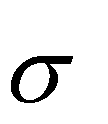
*k*



(7.20b)

x{c định như sau:

*k*



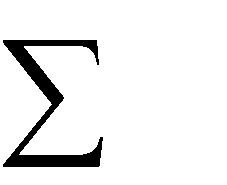
(7.20c)

Khi đó . Ta sẽ x}y dựng thuật to{n để *qk*.

*w*

*0*

=



*N*

*k*

*k*

*y*

*N*

1

1

B}y giờ với mỗi

*k*

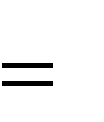
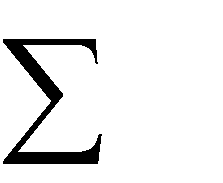


*N*

, ta có hàm q

k

của



*N*

*j*

*j*

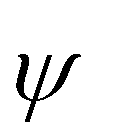
*k*

*k*

*q*

1

,



.

nếu

]

,

[

*q*

*q*



 *Mô tả thuật toán*

Với sai số

v| c{c hằng số dương

*q,*

cho trước, thuật to{n chúng ta gồm 2

<1

pha

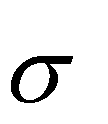
để x{c định c{c tham số

*k*

và

. Trong pha thứ nhất, ta sẽ x{c định c{c

*k*



để

*q*

*k*

]

,

[

*q*

*q*

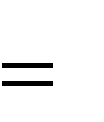
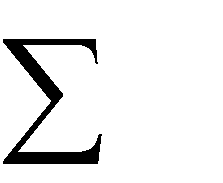


với

mọi k để cho chuẩn của ma trận



tương ứng với chuẩn vectơ cho bởi



*N*

*j*

*j*

*u*

*u*

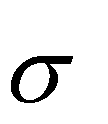
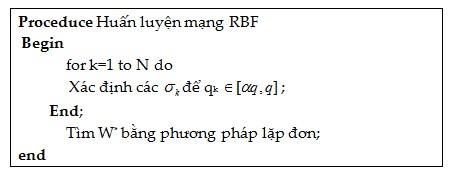
1

\*

thuộc đoạn n|y. Pha sau tìm nghiệm gần đúng

của (7.20a) bằng

phương ph{p lặp đơn. Thuật to{n được đặc tả trong hình 7.14.



**Hình 7.14**. Đặc tả thủ tục lặp huấn luyện mạng.

*Pha thứ nhất*:(X{c định c{c độ rộng h|m b{n kính)

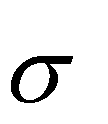
Bởi vì l| h|m đơn điệu giảm của nên l| h|m đơn điệu tăng của . Dựa trên đặc tính n|y ta x}y dựng thuật to{n lặp x{c đính c{c tham số độ rộng.



Dùng hằng số điều chỉnh chỉnh ban đầu 0< 1 tùy ý, với mỗi khởi tạo

1

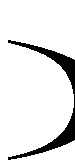
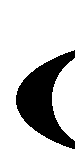
0



, (có thể lấy l| 1 ). Tính theo công thức (7.20c) v| kiểm tra điều kiên

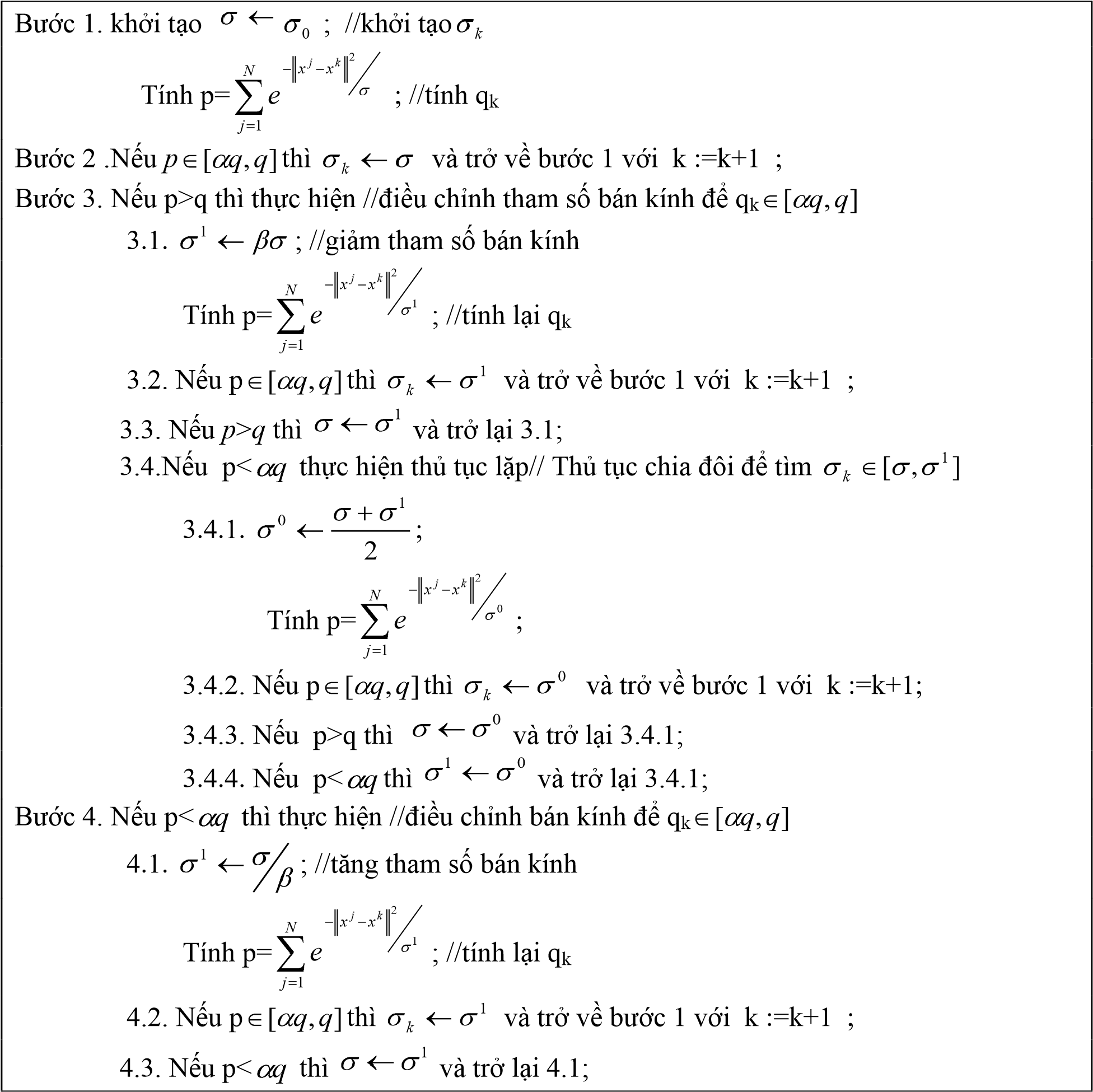
2 2*N n*

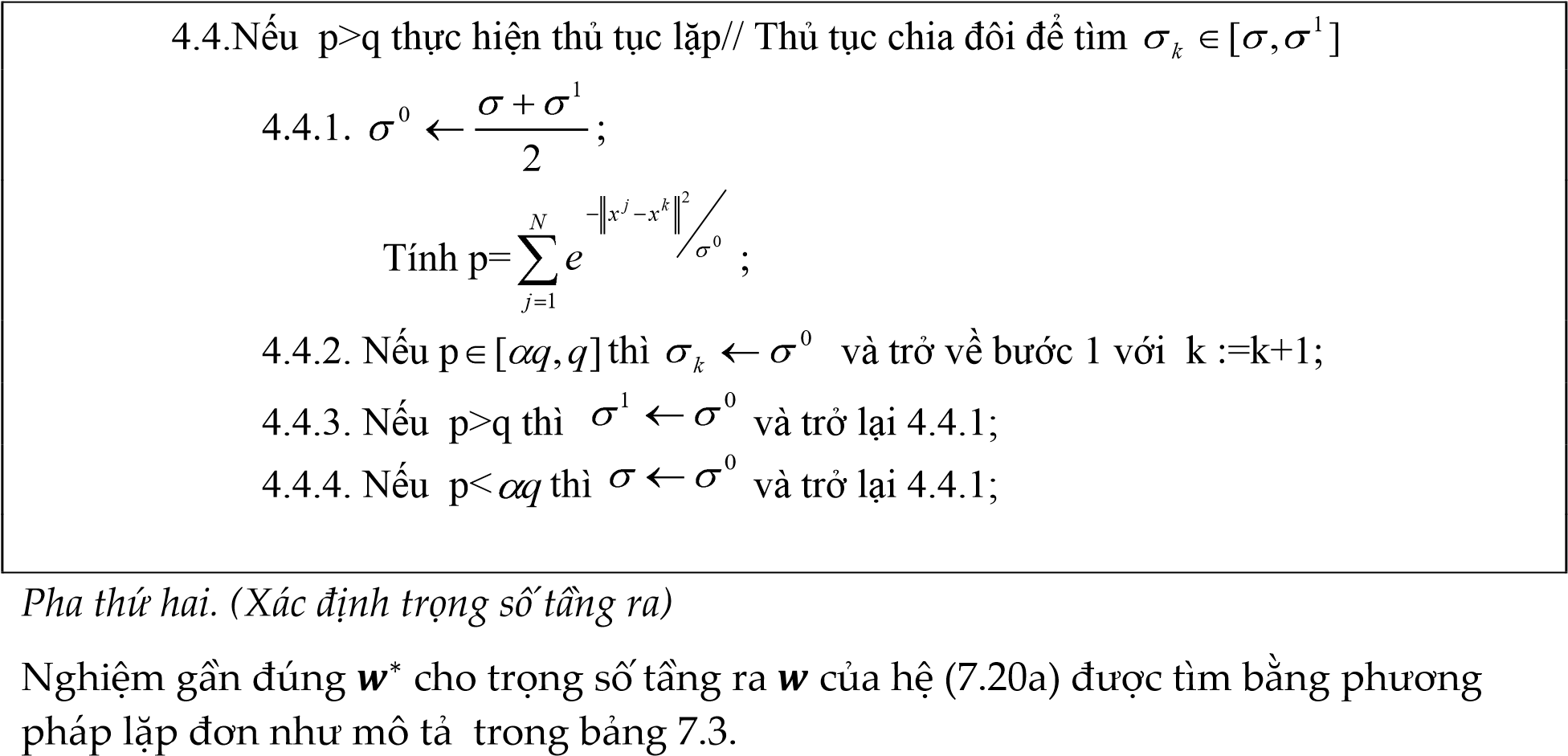
[ *q*,*q*], nếu thì giảm , ngược lại tăng nó tên lên.



Với mọi k=1 tới N, thuật to{n được mô tả trong bảng 7.2.

**Bảng 7.2.** Pha thứ nhất của thuật toán lặp



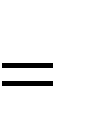


**Bảng 7.3.** Thủ tục lặp đơn tìm trọng số tầng ra

|  |  |
| --- | --- |
| Bước 1. Khởi tạo  =    **z**    ;    Bước 2. Tính  Bước 3. Nếu điều kiện kết thúc chưa đúng thì  Bước 4. | v| trở lại bước 2 ; |

*Điều kiện kết thúc*

Ta ký hiệu chuẩn cho mỗi vectơ *N*-chiều ***u***. Khi đó có thể chọn một



*N*

*j*

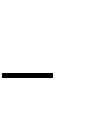
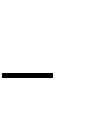
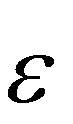
*j*

*u*

1

trong hai điều kiện sau để kết thúc thuật to{n.

1. (7.21a)



\*

0

1

1

*W*

*W*

*q*

*q*

*q*

*Z*

*Z*

*q*

*t*

)

1

ln(

ln

ln

)

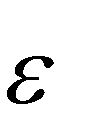
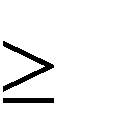
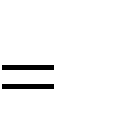
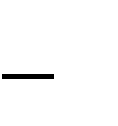
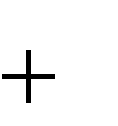
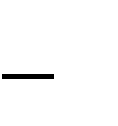
1

(

ln

\*

\*



1. , (7.21b) ln *q* ln *q*

trong đó *t* l| số lần lặp. .

*Tính hội tụ và sai số*

Thuật to{n lặp n|y luôn hội tụ v| ta có đ{nh gi{ sai số của lời giải với nghiệm

. (7.22)

đúng

Thực nghiệm cho thấy thuật to{n huấn luyện rất nhanh, sai số huấn luyện nhỏ v| khi c{c mốc nội suy có ph}n bố khoảng c{ch gần với c{ch đều thì tính tổng hóa tốt hơn hẳn c{c phương ph{p kh{c hiện h|nh.

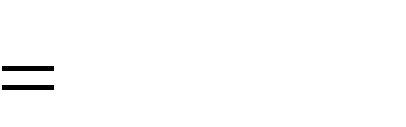
*Mạng nội suy với mốc cách đều*.

Ta xét trường hợp hợp mốc nội suy c{ch đều trong không gian n-chiều v| biến có bước tương ứng. Ta dùng đa chỉ số để biểu diễn c{c mốc nội suy như sau :

, (7.23a) trong đó nhận gi{ trị từ 1 đến .

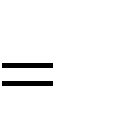
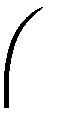
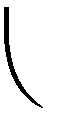
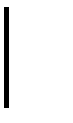
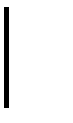
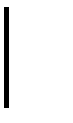
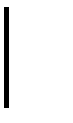
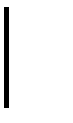
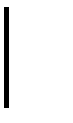
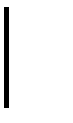
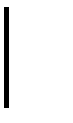
Ký hiêu chuẩn Mahalanobis:

*x A x AxT* , (7.23b)



trong đó *A* l| ma trận

...



2

2

2

1

0

0

...

...

1

0

0

1

*h*

*h*



2

1

....

0

0

*n*

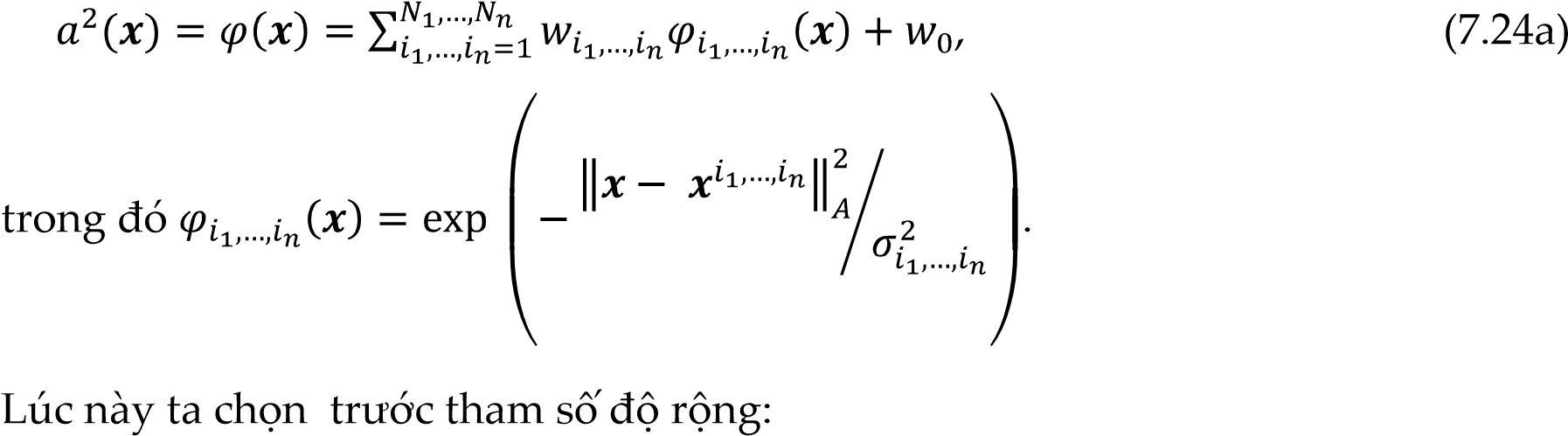
*h*

*A*.... (7.23c)

.

...

Biểu thức (7.15b) b}y giờ th|nh:



(7.24b)

Khi đó thuật to{n hai pha ở trên trở th|nh thuật to{n một pha. Thực

thì

nghiệm cho thấy thuật to{n n|y có thời gian huấn luyện nhanh v| chất lượng mạng tốt hơn hẳn c{c thuật to{n huấn luyện nhanh đã được đề xuất.

**Chú ý**. Thay cho việc sử dụng chuẩn Mahalanobis cho bởi biểu thức (7.32b), ta có thể dùng phép đổi biến:

, (7.25)

thì Euclide m| kết quả không thay đổi.

#### KẾT LUẬN

Mạng nơron nh}n tạo l| tiếp cận mô phỏng kiến trúc, tính năng v| cơ chế xử lý của hệ thần kinh sinh học. v| có ứng dụng rộng rãi. Những c{ch kết nối đa dạng v| c{c luật học kh{c nhau cho ta c{c mạng nơron có tính năng ứng dụng kh{c nhau. C{c cơ chế xử lý phi tuyến, song song, đ{p ứng theo mẫu đ|o tạo<cho phép ta tạo nên các *chương trình* có tính năng vượt trội so với c{ch tiếp cận truyền thống. C{c chương trình n|y có thể c|i đặt cứng hóa như l| c{c *máy*.

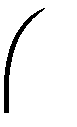
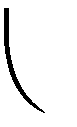
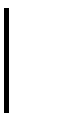
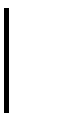
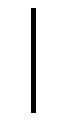
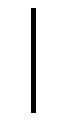
Perceptron của Rosenbatts l| kiểu mạng được đề xuất sớm nhất nhưng đến nay vẫn còn được sử dụng. Luật học của mạng đơn giản, dễ c|i đặt, tuy nhiên chỉ dùng được khi c{c lớp t{ch được tuyến tính. Mạng ADALINE với luật học Widrow-Hoff cho phép giải c{c b|i to{n nhận dạng mẫu với h|m ph}n biệt tuyến tính hoặc hồi quy tuyến tính nhiều chiều, khi kết nối với c{c th|nh phần kh{c, nó có nhiều ứng dụng. Một tín hiệu v| đường trễ lặp kết hợp với mạng ADALINE cho ta một bộ hồi quy trung bình trượt, chúng có thể dùng l|m bộ lọc tín hiệu như l| ví dụ ứng dụng.

Mạng MLP l| kiểu mạng đang được sử dụng rộng rãi nhất hiện nay, đặc biệt cho c{c b|i to{n hồi quy nhiều biến. Nhược điểm của mạng l| thời gian huấn luyện l}u, thường chỉ cho lời giải l| cực trị địa phương của SSE, b|i to{n tối ưu kiến trúc tầng ẩn tới nay vẫn l| b|i to{n mở.

Mạng nơron RBF có ưu điểm l| huấn luyện nhanh v| SSE có duy nhất một cực tiểu to|n cục nên dễ x}y dựng thuật to{n huấn luyện cho lời giải gần đúng. Thuật to{n lặp huấn luyện mạng nội suy dễ c|i đặt v| có ưu điểm nổi trội so với c{c mạng RBF nội suy kh{c. Tuy có nhiều ưu điểm nhưng phạm vi ứng dụng của mạng RBF hạn chế hơn mạng MLP vì nó không dùng để ngoại suy được.

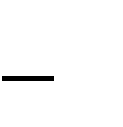
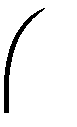
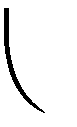
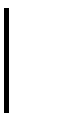
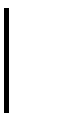
#### BÀI TẬP

1. Cho tập mẫu :{**p1=;** t1=1}, { **p2=**{**;** t2=0},{**p**3**=** t3=0}.



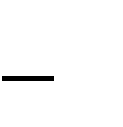
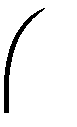
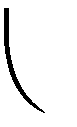
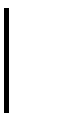
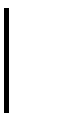
2

1



2

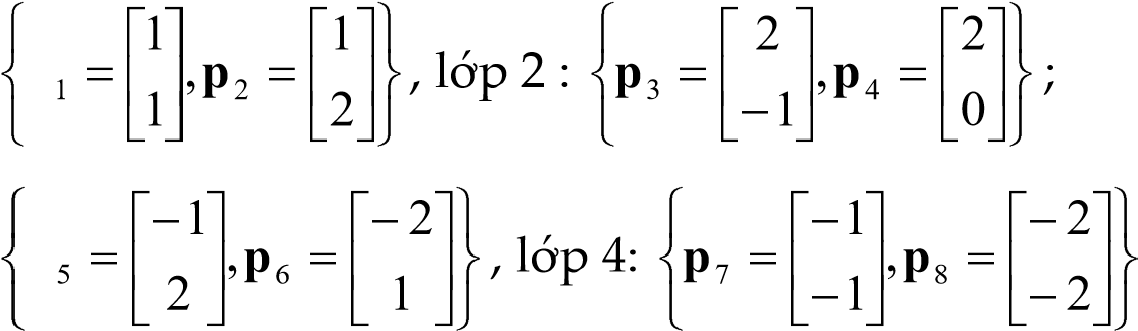
1



1

0

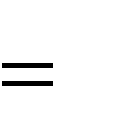
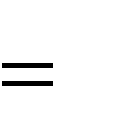
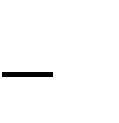
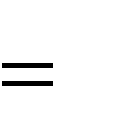
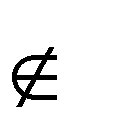
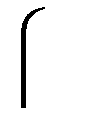
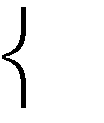
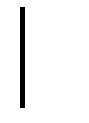
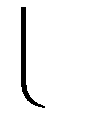
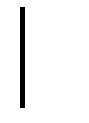
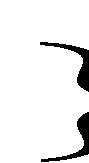
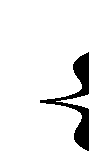
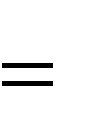
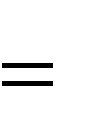
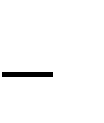
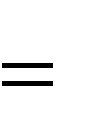
* 1. X}y dựng mạng perceptron ph}n biệt tập mẫu trên với gi{ trị khởi tao: 1**wT** =(1,0 -0,8), gi{ trị ngưỡng luôn lấy b=0.
  2. X}y dựng perceptron với gi{ trị khởi tạo như c}u a) nhưng gi{ trị ngưỡng b được điều chỉnh từng bước theo luật học của Rosenblatts, cho nhận xét so s{nh kết quả.

1. X}y dựng mạng perceptron ph}n biệt bốn lớp dựa trên tập mẫu : lớp 1 : **p**

lớp 3: **p** .

1. Cho bộ lọc thích nghi như trong hình 7.15 , trong đó:

*w*1,1 2,*w*1,2 1,*w*1,3 3, đầu v|o *y*(*k*)



1

,

0

0

1

4

0

5

*k*

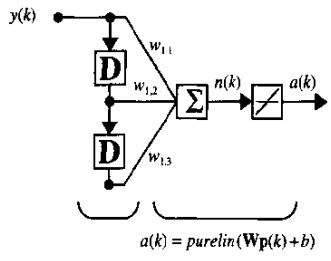
*k*

*k*

Hãy trả lời c{c c}u hỏi sau:

 a)Tính đầu ra a(k) với k0.

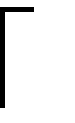
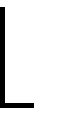
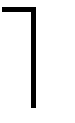
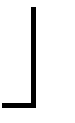
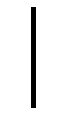
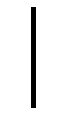
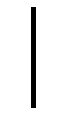
b) y(0) có đóng góp trong những đầu ra n|o?



**Hình 7.15**. Một bộ lọc thích nghi

1. Thiết kế kiến trúc một mạng truyền tới 2 tầng nơron để nội suy h|m nhiều biến *f*1(*x*)

f(x) = *f*2 (*x*) có miền x{c định là R5  và nhận giá trị trong R3 khi tập quan sát có 100 *f*3 (*x*) đối tượng dữ liệu, tức là .



1. Thiết kế kiến trúc mạng nơron RBF cho h|m khi tập dữ liêu quan sát có 1000 đối tượng, tức là

.

1. Viết công thức điều chỉnh trọng số cho mạng MLP có hai tầng nơron v| h|m chuyển log-sigmoid
2. Chỉ ra rằng mạng MLP nhiều tầng với h|m chuyển tuyến tính tương đương với mạng tuyến tính một tầng.

**Chƣơng 8**

## CÁC MÔ HÌNH HỌC ĐỊA PHƢƠNG

Trong đa số c{c phương ph{p học, xấp xỉ cho h|m đích được tìm dựa trên to|n bộ tập dữ liệu quan s{t được trong thời gian *học/huấn luyện*, sau đó nhãn của đối tượng mới được x{c định nhờ h|m xấp xỉ n|y. Kh{c với c{c phương ph{p học n|y, trong c{c phương ph{p học địa phương, người ta thường dựa trên thông tin của c{c mẫu huấn luyện được lưu trữ gần với mẫu mới để tìm nhãn cho mẫu mới đang xét. Quy tắc ph}n lớp *k*-NN trong mục 5.3 v| c{c kỹ thuật phi tham số giới thiệu trong mục 6.1.2 thuộc loại phương ph{p học địa phương.

Chương n|y khởi đầu bằng giới thiệu hai phương ph{p địa phương cho b|i to{n hồi quy: *k*-l{ng giềng gần nhất ( *k*-NN) và hồi quy trọng số địa phương, tiếp theo l| hai kiểu mạng nơron RBF: mạng RBF hồi quy v| mạng RBF nội suy địa phương, sau đó giới thiệu phương ph{p lập luận dựa trên tình huống.

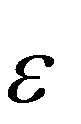
### 8.1. HỌC K-LÁNG GIỀNG GẦN NHẤT (K-NN)

Với mỗi điểm ***x*** chưa biết gi{ trị h|m số, *k*-NN l| phương ph{p đơn giản tìm gi{ trị xấp xỉ dựa trên gi{ tri của h|m tại *k* điểm quan s{t được gần nó.

#### 8.1.1. Lƣợc đồ tổng quát

Ta trở lại với b|i to{n *hồi quy/nội suy* h|m số. Xét h|m gi{ trị thực *y = f(****x****)* xác định trong không gian và tập dữ liệu quan s{t được thỏa mãn:

*,* (8.1a)



trong đó *i* l| c{c đại lượng ngẫu nhiên độc lập cùng ph}n phối có kỳ vọng bằng không. Ta cần x{c định h|m hồi quy *g*(***x***) để xấp xỉ *f*(***x***) với mọi ***x*** đủ tốt theo một tiêu chuẩn n|o đó. Với b|i to{n nội suy, ta cần có:



). (8.1b)

Trong phương ph{p *k*-NN, ta không x{c định được dạng tường minh của h|m *hồi quy/nội suy* *g*(***x***) m| sẽ x{c định gi{ trị cho c{c đối tượng mới ***x*** dựa v|o *k* đối tượng trong tập gần với ***x*** nhất.

Với mỗi ***x*** , ta ký hiệu tập *k* đối tượng trong gần nó nhất l| thì *g*(***x***) thường được x{c định dưới dạng:

*g*(***x***)= , (8.2a)

trong đó l| c{c trọng số phụ thuộc ***x,*** được chuẩn hóa, tức l|: . Một c{ch đơn giản có thể x{c định *g*(***x***) bởi trọng số đều:

*g*(***x***) = . (8.2b)

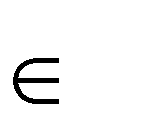
Với b|i to{n có nhiễu trắng v| *k* lớn thì biểu thức (8.2b) cho phép khử được nhiễu. Tuy nhiên, nếu ***x*** gần với c{c đối tượng đã quan s{t được trong thì sai số có thể lớn v| không thích hợp cho b|i to{n nội suy. Thuật to{n sau có thể khắc phục được nhược điểm n|y.

#### 8.1.2. Thuật toán nghịch đảo khoảng cách

Thuật to{n n|y ph{t triển dựa trên lập luận rằng c{c trọng số trong biểu thức (8.2a) ứng với c{c đối tượng c|ng gần với ***x*** c|ng phải có gi{ trị lớn để đảm bảo tính liên tục của h|m nội suy tại c{c mốc quan s{t được.

Thuật to{n thực hiện như sau. Với mỗi đối tượng ***x***, ta dùng khoảng c{ch Euclide:

, (8.2c)



gần ***x*** nhất. Gi{ trị *g*(***x***) được tính theo công thức

***x***

*,*

***y***

R

n

d(

***x,y***

)=



2

1

*i*

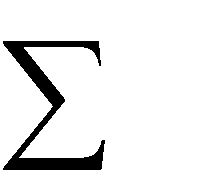
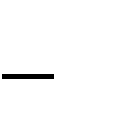
*i*

*n*

*i*

*y*

*x*



để x{c định k đối tượng

với c{c trọng số

(8.2a)

là:

.

(8.2d)

Thuật to{n được mô tả trong hình 8.1.

|  |
| --- |
| **Procedure** Thuật to{n nghich đảo khoảng c{ch tính g(***x***)  **Begin**   1. X{c định gần ***x*** nhất; 2. Tính c{c trọng số ; // theo (8.2d) 3. Tính *g*(***x***); // theo (8.2a);   **End** |

**Hình 8.1**. Thuật toán nghịch đảo khoảng cách tính g(**x**)

Dễ d|ng chứng minh được h|m nội suy *g*(***x***) nay liên tục tại tập điểm quan s{t được , vì vậy l| một h|m liên tục.

***Nhận xét****.*

1. Thuật to{n n|y đơn giản, dễ dùng nên thường được ứng dụng cho c{c b|i to{n không đòi hỏi x{c định trước h|m nội suy, đặc biệt khi c{c điểm cần tính không nhiều.
2. Mặc dù trên đ}y chỉ xét c{c đối tượng có đặc trưng thực nhưng người đọc có thể mở rộng {p dụng cho c{c đối tượng có đặc trưng kh{c, khi trên chúng x{c định một khoảng c{ch để {p dụng công thức (8.2d).

Tuy nhiên, thuật to{n có một số nhược điểm sau:

* Gi{ trị h|m *hồi quy/nội suy* phụ thuộc v|o gi{ trị *k* và khó xác định *k* tối ưu.
* H|m hồi quy không x{c định trước trong giai đoạn học được nên không {p dụng được đối với c{c b|i to{n cần x{c định nhanh gi{ trị g(***x***) khi chạy ứng dụng.
* Như đã nói ở trên, h|m hồi quy l| h|m liên tục. Vì vậy khi dữ liệu huấn luyện có nhiễu trắng thì kết quả hồi quy không tốt nếu so với phương ph{p hồi quy tuyến tính địa phương dưới đ}y.

### 8.2. HỒI QUY TRỌNG SỐ ĐỊA PHƢƠNG

Hồi quy tuyến tính địa phương tổng qu{t hóa của phương ph{p *k*-NN bằng việc tìm một h|m xấp xỉ *f*  trên một miền địa phương bao quanh ***x***, nhờ đó tính được gi{ trị h|m hồi quy tại đó.

#### 8.2.1. Lƣợc đồ tổng quát

Trong phương ph{p n|y, người ta xấp xỉ h|m mục tiêu trong miền l}n cận của ***x***gần nó nhờ dùng một h|m tuyến tính, h|m bậc hai, mạng nơron MLP hay một số dạng h|m kh{c, dựa trên c{c mốc trong .

Xét tập mẫu thõa mãn (8.1a) v| điểm ***x*** trong cần tính gi{ trị gần đúng *f*(***x***) bởi gi{ trị h|m hồi quy g(***x***). Trong mục 2.4 ở chương 2, đã giới thiệu phương ph{p tìm h|m hồi quy nhờ cực tiểu tổng bình phương sai số:

. (8.3a)

Trong thực tế, giá trị hàm số ở những điểm xa với điểm ***x*** được xét thì ít ảnh hưởng tới giá trị của hàm tại điểm n|y nên người ta có thể thêm vào trọng số xác

định bởi h|m đơn điệu giảm

, (8.3b)

trong đó . (8.3c)

Điều kiện (8.3c) thể hiện tính ảnh hưởng *địa phương* của c{c điểm quan sát trong h|m đ{nh gi{ *E* ở (8.3b), khoảng c{ch Euclide thường được dùng trong biểu thức tính hàm .

Trong c{c ứng dụng, thay cho đòi hỏi điều kiện (8.3c), với số tự nhiên *k* chọn trước, người ta chỉ xét *k* điểm quan s{t gần ***x*** nhất để tính gi{ trị hồi quy h|m số tại ***x*** và hàm *E* trong biểu thức (8.3b) trở th|nh:

, (8.4a)

(8.4b)

khi lấy

hoặc đơn giản hơn

=1.

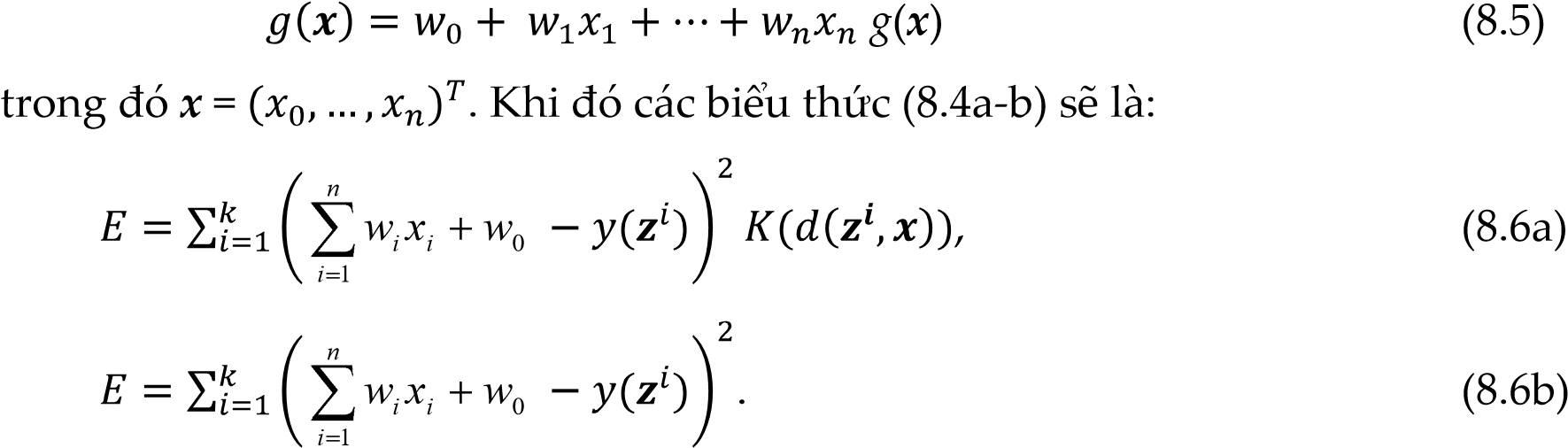
Cực tiểu của *E* có thể tìm nhờ thuật to{n gradient hoặc giải trực tiếp hệ phương trình tìm điểm dừng của nó.

Từ lược đồ trên, trên ta thấy cụm từ “hồi quy trọng số địa phương”được hiểu như sau: gọi l| địa phương bởi vì h|m n|y được xấp xỉ chỉ dựa trên dữ liệu gần điểm truy vấn được xét , gọi l| trọng số bởi vì sự đóng góp của mỗi mẫu huấn luyện được g{n trọng số phụ thuộc v|o khoảng c{ch của nó đến điểm truy vấn, còn hồi quy l| kh{i niệm được dùng rộng rãi trong học thống kê cho b|i to{n xấp xỉ c{c h|m nhận gi{ trị thực.

Như đã biết, h|m hồi quy có thể l| một h|m phụ thuộc tham số tùy ý, trong đó thông dụng nhất l| h|m tuyến tính.

#### 8.2.2. Hồi quy tuyến tính địa phƣơng

Trong trường hợp h|m hồi quy *g* l| h|m tuyến tính:



C{c hệ số cực tiểu *E* theo công thức (8.6a-b) có thể tìm nhờ giải trực tiếp hệ phương trình hoặc phương ph{p gradient đã biết.

Vì *g* l| h|m tuyến tính nên dễ d|ng tính được gradient(*E*) v| khi giải trực tiếp thì

 (8.7a)

(8.7b)

(8.7c)

Khi *n* lớn, nghiệm của c{c hệ trên có thể không ổn định do tích lũy sai số trong tính to{n, ta có thể dùng phương ph{p gradient. Thuật to{n gradient để tìm cực tiểu E được mô tả trong bảng 8.1, điều kiện dừng của nó như đã xét trong chương 2.

**Bảng 8.1.** Thuật toán gradient tìm giá trị hàm hồi quy g(**x**)



#### *Nhận xét*

Cũng như *k*-NN ở mục trước, phương ph{p n|y v| thuộc loại học dựa trên mẫu (Instance based learning) v| có chung c{c đặc điểm khi sử dụng, chẳng hạn: thời gian huấn luyện ít nhưng không tính trước được h|m hồi quy. Vì vậy hạn chế phạm vi {p dụng.

Tuy nhiên phương ph{p n|y l| công cụ tốt để tạo dữ liệu cho c{c phương ph{p kh{c. Mạng nơron RBF hôi quy l| một ví dụ ứng dụng của phương ph{p n|y.

### 8.3. MẠNG NƠRON RBF HỒI QUY

#### 8.3.1. Đặt vấn đề

Bởi vì c{c h|m RBF gần như triệt tiêu tại những điểm xa t}m nên c{c mạng nơron RBF cũng thuộc loại học địa phương. C{c mạng RBF nội suy ở chương trước có nhiều ưu điểm nhưng khi dùng cho c{c b|i to{n xấp xỉ thì chất lượng thấp. Hình 8.2 chỉ ra trường hợp khi c{c điểm mốc quan s{t được có từng cặp một gần nhau thì

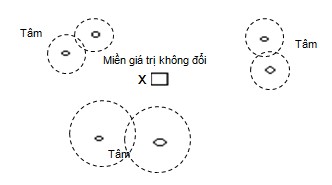
những điểm xa t}m có gi{ trị đầu ra bằng

không đổi, vì vậy mạng n|y chỉ đ{p ứng

được điều kiện tương thích (8.1b) ở c{c điểm dữ liệu cho bởi tập

*D*

mà thôi.

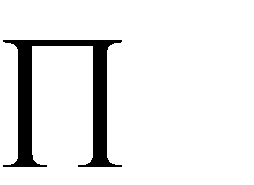


**Hình 8.2.** Khi tâm phân bố không tốt, các điểm **x** xa tâm có giá trị không đổi

Để khắc phục hạn chế n|y ta sử dụng phương ph{p hồi quy tuyến tính địa phương để tạo ra tập dữ liệu mới có mốc c{ch đều v| sử dụng thuật to{n lặp một pha đã biết để huấn luyện mạng nơron RBF nội suy.

#### 8.3.2. Xây dựng mạng nơron RBF hồi quy

Giả sử tập dữ liệu nằm trong hình hộp *n*-chiều B= *a* ,*b* , mạng nơron RBF hồi quy được x}y dựng như sau.



*n*

*i*

*i*

*i*

1

Tùy theo mật độ ph}n bố của dữ liệu trên miền gi{ trị của mỗi thuộc tính, ta chọn c{c bước thích hợp cho mỗi thuộc tính v| tạo ra một lưới điểm mốc c{ch đều trên B (Thực nghiệm cho thấy nên chọn c{c bước sao cho M lớn hơn N).

Với một số tự nhiên k chọn trước, dùng hồi quy tuyến tính địa phương để xấp xỉ h|m hồi quy *g*(***x***) tại mọi điểm ***x*** trên lưới n|y. Sau đó sử dụng tập dữ liệu vừa tính được để x}y dựng mạng hồi quy nhờ {p dụng thuật to{n huấn

luyện một pha trong mục 7.4.2 chương trước.

Thuật to{n x}y dựng mạng RBF hồi quy được mô tả trong bảng 8.2. Thực nghiệm trên dữ liệu thực v| dữ liệu mô phỏng cho thấy mạng n|y có chất lượng tốt.

**Bảng 8.2**. Thuật toán xây dựng mạng nơron RBF hồi quy

Bước 1. Tạo lưới c{ch đều *z j Mj* 1 , chọn k > n ;

Bước 2.Tính **//** Dùng phương ph{p hồi quy tuyến

tính địa phương

Bước 3. Huấn luyện mạng hồi quy; // Dùng thuật toán huấn luyện mạng nội suy với mốc c{ch đều

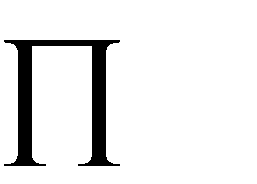


### 8.4. MẠNG RBF ĐỊA PHƢƠNG

Trong nhiều bài toán, tập dữ liệu quan s{t được rất lớn hoặc thường xuyên có bổ sung dữ liệu mới thì có thể dùng c{c mạng RBF địa phương nhờ dùng kỹ thuật cây *k*-d sau đ}y để x}y dựng mạng. Để dễ theo dõi, trong mục n|y chỉ giới thiệu x}y dựng mạng nội suy, người đọc có thể ứng dụng kỹ thuật c}y *k*-d để x}y dựng mạng hồi quy địa phương như l| b|i tập.

#### 8.4.1. Kiến trúc và thủ tục xây dựng mạng

Giả sử tập mẫu nằm trong miền đóng giới nội B= [*a* ,*bi* ] và



*n*

*i*

*i*

1

*N* lớn. Ta định trước số *M* cho cận trên của số điểm trong mỗi cụm con v| chia miền *B* th|nh c{c hình hộp *n* chiều sao cho số mốc trong mỗi cụm nhỏ hơn *M* nhờ thuật to{n ph}n cụm nhờ c}y *k-*d (mô tả trong mục 8.4.2 ở dưới). Sau đó sử dụng thuật to{n huấn luyện lặp đã nói ở mục 7.4 chương trước để huấn luyện c{c mạng RBF cho mỗi miền con .

Như vậy Mạng RBF địa phương được x}y dựng nhờ kết hợp giữa thủ tục c}y *k*-d với thủ tục x}y dựng c{c mạng RBF con.

Khi có một dữ liệu mới, ta cần định vị nó thuộc miền n|o trong tập thì chỉ cần huấn luyện lại mạng RBF trên miền đó. C}y *k*-d giúp ta dễ d|ng x}y dựng bộ định vị n|y. Nếu số điểm dữ liệu mẫu trong lớn hơn M, ta chia th|nh hai miền con theo thủ tục c}y *k*-d v| huấn luyện lại c{c mạng con RBF trên miền n|y nhờ dùng độ rộng b{n kính v| trọng số tầng ra đang có l|m gi{ trị khởi tạo. Còn khi số điểm dữ liệu mẫu trong không lớn hơn M thì bổ sung dữ liệu mới v| huấn luyện lại.

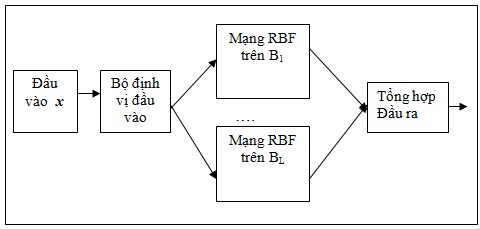
Nhờ số điểm dữ liệu trên mỗi miền con ít nên thời gian huấn luyện mạng RBF địa phương nhanh hơn thuật to{n hai pha. Hơn nữa, chất lượng tổng qu{t hóa của mạng mới cũng tốt hơn. Đặc biệt, khi có dữ liệu quan s{t bổ sung thì thời gian huấn luyện lại mạng con l| không đ{ng kể so với mạng đầy đủ huấn luyện bằng phương ph{p lặp. Thủ tục x}y dựng mạng được đặc tả trong hình 8.3*.*

|  |
| --- |
| **Procedure** X}y dựng mạng RBF địa phương;  **Begin**   1. Ph}n B th|nh c{c miền con ***B1,..,Bk*;** // sử dụng thuật to{n ph}n cụm c}y *k-d* để số mốc trong mỗi cụm con không vượt qu{ *M*. 2. X}y dựng bộ định vị đầu v|o cho c{c mạng RBF con; 3. Huấn luyện c{c mạng RBF con;// Dùng thuật to{n huấn luyện lặp 4. Kết nối bộ định vị đầu v|o với c{c mạng con để được mạng RBF địa phương.   **End** |

**Hình 8.3.** Thủ tục xây dựng mạng RBF nội suy địa phương

Sau đó x}y dựng thủ tục để định vị mỗi điểm mới ***x*** trong B thuộc miền con n|o trong tập .

Kiến trúc mạng được mô tả trong hình 8.4, trong pha huấn luyện, th|nh phần *tổng hợp đầu ra* không hoạt động, mỗi mạng con được huấn luyện độc lập.



**Hình 8.4**. Mô hình kiến trúc mạng RBF địa phương

Trong khi nội suy, với mỗi gi{ trị v|o ***x***, *bộ định vị* sẽ x{c định mạng con tương ứng, còn tất cả c{c mạng con kh{c có gi{ trị v|o rỗng, do đó gi{ trị đầu ra của tất cả c{c mạng con đều l| rỗng ngoại trừ mạng con có đầu v|o ***x***. Sau đó th|nh phần *tổng hợp đầu ra* l| tổng đầu ra của c{c mạng RBF con.

Với cấu trúc mạng RBF địa phương mới n|y, h|m nội suy khả vi liên tục trên từng miền con nhưng không liên tục trên to|n miền B được xét. Khi *M* nhỏ thì thời gian huấn luyện v| huấn luyện lại cũng như sai số giảm. Tuy nhiên, nếu số mạng con RBF nhiều thì sẽ l|m cho mạng phức tạp hơn. Đặc điểm n|y chính l| điều không mong muốn trong giải tích số. Như thế, việc lựa chọn *M* phải thích hợp để c}n bằng giữa c{c điều kiện đó.

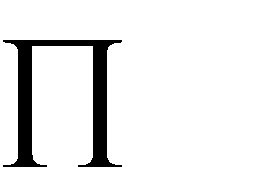
#### 8.4.2. Thuật toán phân cụm nhờ cây k-d

Cây K-d được Bentley đề xuất để tổ chức dữ liêu, c{c biến thể của nó l| công cụ hiệu quả để ph}n cụm nhanh dữ liệu. Thuật to{n sau đ}y sửa đổi nhỏ kỹ thuật *k-d* thông dụng để ph}n hình hộp *n*-chiều **B** chứa *N* đối tượng dữ liệu th|nh c{c hình hộp con *B1,…, Bk* sao cho mỗi hình hộp *Bj* chứa không qu{ *M* đối tượng dữ liệu.

Thuật to{n có thể mô tả bởi qu{ trình x}y dựng một c}y nhị ph}n *n*-chiều có gốc l| hình hộp B chứa *N* đối tượng dữ liệu, mỗi nút ở độ s}u *l* l| hình hộp *n*-chiều có được nhờ một nh{t cắt trực giao với cạnh lớn nhất của hình hộp cha chia th|nh hai hình hộp con để chúng chứa số lượng dữ liệu xấp xỉ nhau. Như vậy một hình hộp con chứa những điểm dữ liệu ứng với gi{ thuộc tính có cạnh chia nhỏ hơn điểm cắt v| hình hộp còn lại có gi{ trị lớn hơn. Thủ tục chia đôi hình hộp *n*-chiều Bj=

*n*

[*ai* ,*bij* ] th|nh hai hình hộp con được đặc tả trong hình 8.5.

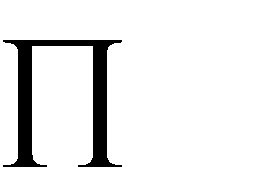


*j*

##### *i* 1

|  |
| --- |
| **Procedure c**hia đôi hình hộp *n*-chiều Bj= [*a* ,*bij* ]  *n*  *i*  *j*  *i*  1  **Begin**  Bước 1. Chọn *i* sao cho cạnh [*aij* ,*bij* ] của B*j* l| lớn nhất; // Chọn phương cho siêu mặt cắt  Bước 2. Chia *Bj* = B1  B2; // Chia cạnh [*aij* ,*bij* ] bởi nh{t cắt trực giao v| qua ít nhất một điểm dữ liệu sao cho hai hình hộp nhận được từ nó chứa số đối tượng dữ liệu bằng nhau hoặc lệch 1, với quy ước c{c điểm dữ liệu thuộc nh{t cắt có thể đồng thời tính l| thuộc hai hình hộp  **End** |

**Hình 8.5.** Thủ tục chia đôi hình hộp n-chiều Bj = [*a* ,*bij* ]



*n*

*i*

*j*

*i*

1

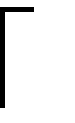
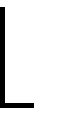
Thủ tục chia đôi kết thúc khi số lượng đối tượng trong mỗi hình hộp con không vượt qu{ *M* chọn trước. Như vậy số mốc nội suy trong mỗi hình hộp con ở độ s}u *d*

*N*

được xem l| bằng nhau v| xấp xỉ bằng 2*d* do có những mốc có thể được tính v|o

nhiều hình hộp. Khi thủ tục kết thúc thì số lượng dữ liệu nằm trong mỗi miền con

*N N*



*M*

*M*

,

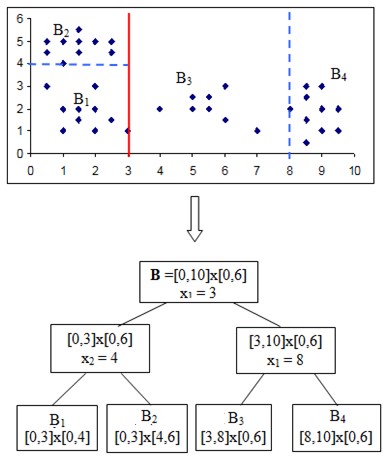
2

thuộc khoảng v| có khoảng từ đến 2 miền con. Ngo|i ra, có thể chọn

*M M*

trước số miền con l|m điều kiện dừng chia.

Hình 8.6 mô tả kết quả chia hình hộp 2 chiều \*0,10+\*0,6+ chứa 38 dữ liệu (*N*=38) với *M*=10. Kết quả có 4 hình hộp con đều chứa 10 dữ liệu, trong đó c{c điểm (1,4) v| (8,2) được tính đồng thời ở hai hình hộp chứa nó còn điểm (3,1) chỉ thuộc B3.



**Hình 8.6**. Cây k-d mô tả tập dữ liệu trong không gian 2 chiều, với N=38, M=10.

Với cấu trúc c}y nhị ph}n n|y, khi có điểm dữ liệu ***x*** mới, ta dễ d|ng x{c định được nó thuộc hình hộp con n|o ở l{ bằng c{ch tìm kiếm từ gốc đến nhờ so s{nh gi{ trị thuộc tính ở cạnh cắt của hình hộp tương ứng với gi{ trị ở điểm cắt.

**Chú ý**. Việc chọn cạnh [*aij* ,*bij* ] trong bước 1 của thủ tục chia đôi hình hộp Bj có thể thay bằng c{ch chọn tuần tự theo c{c biến như thủ tục c}y k-d thông thường.

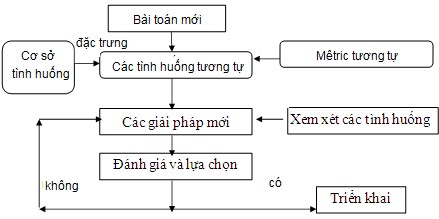
Trên đ}y, giới thiệu c{c phương ph{p địa phương giải b|i to{n hồi quy. Phương ph{p dựa v|o tiếp cận *trợ giúp quyết định* để tìm lời giải cho c{c b|i to{n có cấu trúc xấu, không thể giải bằng thuật to{n chặt chẽ được.

### 8.5. LẬP LUẬN DỰA TRÊN TÌNH HUỐNG (CBR)

C{c phương ph{p *k-*NN v| hồi quy trọng số địa phương có chung 3 đặc tính chính.

1. Chúng l| c{c phương ph{p học lười, không đưa ra h|m số cụ thể để kh{i qu{t hóa dữ liệu huấn luyện m| chỉ thực hiện thủ tục x{c định gi{ trị h|m khi có mẫu truy vấn mới.
2. Gi{ trị h|m tại c{c mẫu truy vấn mới được x{c định bằng việc ph}n tích c{c mẫu *gần/tương tự* nó nhất m| bỏ qua c{c mẫu kh{c xa với mẫu đang xét.
3. C{c mẫu được đặc trưng bởi c{c điểm trong một không gian Euclidean n-chiều.

Lập luận dựa trên tình huống (Case based resoning: CBR) l| một kiểu học dựa trên 2 nguyên tắc đầu tiên nhưng không dựa trên nguyên tắc thứ ba. Trong phương ph{p CBR, c{c mẫu có thể được đặc trưng bởi c{c thuộc tính có gi{ trị định danh v| dùng metric thích hợp (chẳng hạn c{c mêtric đã được giới thiệu trong mục 4.2.4.1) để x{c định tính tương tự giữa c{c tình huống cần xử lý. Thoạt tiên, CBR được dùng trong c{c hệ hỗ trợ quyết định, trong đó mỗi khi có một tình huống mới cần tìm giải ph{p, người ta tìm c{c tình huống tương tự với nó trong dữ liệu lưu trữ v| tham khảo giải ph{p đã có để x}y dựng lời giải mới. Lược đồ chung của phương ph{p CBR được mô tả trong hình 8.7.



**Hình 8.7**. Sơ đồ cấu trúc của hệ CBR

CBR đã được {p dụng thiết kế c{c *thiết bị/máy móc* dựa trên sử dụng thư viện lưu trữ c{c thiết kế trước đó, hoặc suy luận về c{c trường hợp hợp lệ mới dựa trên c{c quyết định trước đó. CBR cũng được dùng để giải quyết c{c b|i to{n lập kế hoạch v| lập lịch bằng việc sử dụng lại v| kết hợp c{c phần lời giải trước cho c{c b|i to{n tương tự. Ng|y nay, CBR còn được {p dụng cho c{c hệ suy luận để x}y dựng c{c hệ thiết kế để x}y dựng c{c hệ như thiết kế c{c c{c hệ điều khiển tự động.

*Ví dụ*. Để dự b{o c{c hiện tượng thời tiết bất thường có thể gặp khi bay dựa trên c{c số liệu từ c{c trạm quan trắc, người ta so s{nh c{c số liệu n|y với c{c số liệu của c{c ng|y l}n cận trong những năm trước đó để tìm ra những ng|y tương tự nhất với nó v| tham khảo c{c hiện tượng đã có. C{c hiện tượng n|y được thông b{o cho c{c phi công để chuẩn bị trước giải ph{p xử lý m| không bị bất ngờ khi gặp phải nó khi bay. Hệ thống CADET (Sycara et al. 1992) sử dụng suy luận dựa trên tình huống để giúp thiết kế hệ điều khiển vòi nước.

Tóm lại, CBR l| một phương ph{p học dựa trên mẫu, ở đó c{c mẫu (c{c tình huống) có thể được đặc tả bằng nhiều quan hệ, khi gặp tình huống mới, người ta kết hợp suy luận dựa trên cơ sở tri thức v| c{c giải ph{p đã {p dụng trong c{c tình huống tương tự để x}y dựng lời giải. Một vấn đề cần nghiên cứu hiện nay trong CBR l| ph{t triển c{c phương ph{p thích hợp để cải tiến việc lưu trữ v| truy vấn c{c tình huống.

#### KẾT LUẬN

Trong chương n|y ta đã xét ba phương ph{p học lười hay học dựa trên mẫu: hồi quy *k*-NN, hồi quy trọng số địa phương v| CBR. C{c mạng nơron RBF được xem là cầu nối giữa mạng MLP v| c{c phương ph{p hồi quy dựa trên mẫu. Những phương ph{p học địa phương có c{c điểm chính sau.

* C{c phương ph{p học dựa trên mẫu kh{c với c{c c{ch tiếp cận phép xấp xỉ h|m kh{c ở chỗ chúng chỉ xử lý c{c mẫu huấn luyện khi chúng phải g{n nhãn cho một mẫu truy vấn mới. Kết quả l| chúng không x}y dựng giả thuyết tường minh cho h|m mục tiêu trên to|n bộ không gian mẫu, độc lập với mẫu truy vấn đó. Thay v|o đó, chúng chỉ tạo ra một phép xấp xỉ địa phương cho h|m mục tiêu đối với mỗi mẫu truy vấn.
* Ưu điểm của c{c phương ph{p dựa trên mẫu l| cho phép tìm nhãn của mẫu mới khi h|m mục tiêu phức tạp nhờ dùng c{c phép xấp xỉ địa phương đơn giản hơn dựa trên thông tin trong tập mẫu huấn được lưu trữ. C{c khó khăn chính trong c{ch tiếp cận n|y l|: 1) X{c định một độ đo khoảng c{ch phù hợp để tìm ra c{c mẫu liên quan (đặc biệt, khi c{c mẫu được miêu tả bởi c{c mô tả tượng trưng phức tạp); 2) Xử lý c{c ảnh hưởng bất lợi của c{c đặc trưng ít liên quan đối với độ đo khoảng c{ch đã có.
* k-NN l| một thuật to{n dựa trên mẫu dùng cho việc xấp xỉ c{c h|m mục tiêu nhận gi{ trị *thực/rời rạc* v| c{c mẫu được tương ứng với c{c điểm trong không gian Euclide *n*-chiều. Gi{ trị h|m mục tiêu cho một mẫu truy vấn mới được ước lượng từ c{c gi{ trị đã biết của *k* mẫu huấn luyện gần nhất.
* C{c phương ph{p hồi quy trọng số địa phương l| tổng qu{t hóa của *k*-NN, trong đó, khi cần g{n nhãn cho mẫu mới, người ta x}y dựng một phép xấp xỉ địa phương hiển của h|m mục tiêu cho mẫu n|y. Phép xấp xỉ địa phương cho h|m mục tiêu có thể thực hiện nhờ dùng nhiều dạng h|m như c{c h|m hằng số, h|m tuyến tính hoặc h|m bậc hai hay dựa v|o c{c h|m nh}n có ảnh hưởng địa phương trong không gian mẫu.
* C{c mạng nơron RBF l| một loại mạng nơron nh}n tạo được x}y dựng từ c{c h|m nh}n ảnh hưởng địa phương. C{c phương ph{p n|y có thể được xem như l| tổng hợp c{c hướng tiếp cận dựa trên mẫu (ảnh hưởng được khoanh vùng theo không gian của mỗi h|m nh}n) v| c{c hướng tiếp cận mạng nơron. Mạng nơron RBF địa phương cho phép x}y dựng h|m nội suy với tập mẫu lớn v| thường có bổ sung c{c mẫu quan s{t mới để huấn luyện lại mạng. Mạng nơron RBF xấp xỉ dùng phương ph{p hồi quy trọng số địa phương để x}y dựng một tập mẫu có mốc c{ch đều rồi dùng nó để x}y dựng mạng RBF bằng phương ph{p lặp.
* Lập luận dựa trên tình huống l| một c{ch tiếp cận dựa trên mẫu, trong đó c{c mẫu được miêu tả bởi c{c đặc tả logic phức tạp hơn c{c điểm trong không gian Euclide. Dựa v|o c{c mô tả phức tạp của c{c mẫu đã cho, c{c phương ph{p đa dạng được đề xuất để x}y dựng h|m đích từ cho c{c mẫu mới, chúng được dùng để tham khảo cho x{c định lời giải cuối cùng. C{c phương ph{p lập luận dựa trên tình huống đã v| đang được dùng trong nhiều ứng dụng như tạo mô hình lập luận hợp lý v| hướng dẫn tìm kiếm trong c{c b|i to{n thiết kế, lập kế hoạch phức tạp.

Một nhược điểm của c{c hướng tiếp cận dựa trên mẫu l| khi phải tìm nhãn cho nhiều mẫu mới thì có thể mất nhiều thời gian, hạn chế phạm vi {p dụng. Nguyên nh}n của hạn chế n|y l| do hầu hết c{c tính to{n được thực hiện ở lúc tìm nhãn cho các mẫu kể từ khi bắt gặp mẫu mới đầu tiên m| không thể thực hiện tìm h|m đích trước. Bởi vậy, c{c kỹ thuật sắp xếp c{c mẫu huấn luyện thuận tiện cho qu{ trình tính to{n l| một b|i to{n thực tế rất ý nghĩa để giảm thời gian truy vấn.

Nhược điểm thứ hai của c{c hướng tiếp cận dựa trên mẫu (đặc biệt l| c{ch tiếp cận l{ng giềng gần nhất) l| chúng dùng mêtric dựa trên tất c{c thuộc tính của không gian mẫu để tìm ra c{c mẫu huấn luyện tương tự được lưu trữ trong bộ nhớ. Khi kh{i niệm mục tiêu chỉ dựa trên một v|i trong nhiều thuộc tính của mẫu thì c{c mẫu thật sự tương tự với mẫu mới có thể không phải l| c{c mẫu gần nhau theo khoảng c{ch được xét.

#### BÀI TẬP

1. X}y dựng ví dụ để so s{nh c{c phương ph{p K-NN, hồi quy trọng số địa cho hàm *f = sin*(*2**x1*)*+ cos*(*2**x2*), dựa trên tập mẫu với ***x*** có ph}n bố đều trên miền \*0,1+x\*0,1+ gồm 40 mẫu đ|o tạo v| 10 mẫu kiểm tra.
2. Tạo ra tập mẫu quan s{t được gồm 100 mẫu với nhiễu của h|m số có ph}n bố chuẩn N(0,0.2 cho hàm *f = 3sin(**x1x2)+ 2sin(**x1+x2)*), trong đó ***x*** có ph}n bố đều trên miền \*0,1+x\*0,1+.
   1. Dùng 90 mẫu để x}y dựng mạng nơron RBF hồi quy cho h|m n|y.
   2. Cũng dùng dữ liệu huấn luyện ở c}u a) để ước lượng gi{ trị h|m *f* tại 10 mẫu còn lại v| so s{nh chất lượng xấp xỉ với phương ph{p trên.
3. Chỉ ra c{ch dùng phương ph{p *k*-NN cho h|m số m| tập mẫu có 5 thuộc tính định danh v| 2 thuộc tính thực.

**Chƣơng 9**

## HỌC TĂNG CƢỜNG

Kh{c với c{c phương ph{p học trước đ}y, trong học tăng cường bộ học l| t{c tử (agent ) ra quyết định (decision-making), nó thực hiện c{c t{c động trong môi trường v| nhận c{c khoản thưởng (hoặc phạt) cho c{c t{c động của mình trong qu{ trình giải quyết b|i to{n. Sau c{c thực nghiệm kiểu *thử-sai*, nó học được chính s{ch tốt nhất, tức l| chuỗi t{c động tốt nhất để tối ưu hóa gi{ trị tích lũy thưởng.

### 9.1. TÁC TỬ VÀ CÁC BÀI TOÁN HỌC

Trong chương 1, ta đã l|m quen với ví dụ về robot tìm chu trình Hamilton tối ưu, trước khi mô tả t{c tử v| ph{t biểu c{c b|i to{n học cho nó, ta xét thêm c{c ví dụ về robot tìm đường đi v| hệ chơi cờ.

#### 9.1.1. Một số ví dụ

*Ví dụ 1: Tìm đường đi của robot*

Xét một robot có trang bị cảm biến (sensor) để quan s{t trạng th{i của môi trường v| chọn đường đi. Tại thời điểm ban đầu, nó ở vị trí A v| muốn tới điểm B theo con đường ngắn nhất. Mỗi ngã rẽ, điểm A hoặc điểm đích B được xem l| một trạng th{i của môi trường m| robot dùng l|m mốc. Như vậy robot xuất ph{t ở trạng thái v| ở mỗi trạng th{i nó chọn đường đi như l| t{c động tương ứng sao cho đến được được trạng th{i đích , trong đó *k* l| số lần chuyển trạng th{i chưa x{c định trước m| tùy theo đường đi được chọn. Ký hiệu độ d|i quãng đường( chi phí) từ tới là , khi đó độ d|i đường đi (chi phí) tích hợp l|:

(9.1a)

Robot cần tìm một chính sách chọn đường đi ở mỗi chỗ rẽ sao cho nhỏ nhất. Nếu thay như l| lượng thưởng (m l| hằng số chọn trước tùy ý) thì ta có bài to{n tương đương l| tìm cực đại hàm giá trị tích lũy thưởng :

(9.1b)

Trong trường hợp n|y, lượng thưởng (chi phí ) l| đơn định đối với mỗi trạng th{i v| t{c động . Để đơn giản về sau ta chỉ xét b|i to{n với c{c t{c động có thưởng v|

tìm cực đại gi{ trị tích lũy thưởng.

Nếu ta xét mục đích của robot l| đi sạc pin khi được b{o sắp cạn thì B l| điểm sạc pin v| robot cần phải đến trước khi pin cạn. Giả sử thời gian còn lại cho robot di chuyển l| T nhưng robot không đo được lượng pin tiêu hao trên quảng đường. Khi đó nếu robot đến được đích thì mọi t{c động sẽ được thưởng cho gi{ trị l| 100, ngược lại lượng thưởng bằng không. Lúc n|y lượng thưởng cho mỗi t{c động ở trạng th{i không được biết ngay m| phải chờ đến khi kết thúc nó tìm được B hay nằm giữa đường mới x{c định được. Hơn nữa mỗi đoạn đường tới có thể thuộc đường đến đích hoặc không nên lượng thưởng cho t{c động chọn nó nhận gi{ trị ngẫu nhiên trong tập ,0,100}. Về sau sẽ giả thiết rằng ước lượng được c{c x{c suất có điều kiện cho mọi cặp trạng th{i v| t{c động nhờ thống kê. Khi đó ta cần cực

đại kỳ vọng của h|m tích lũy:

(9.1c)

Trong hai trường hợp trên, robot có trạng th{i đích B gọi l| *trạng thái hút* (asorbing state)/ *kết thúc* v| nó dừng l|m việc ở đó sau một số lần chuyển trạng th{i. Nếu robot điều khiển xe tải đi nhận h|ng, ph}n phối v| nhận lu}n phiên không nghỉ, còn c{c gi{ trị thưởng l| đơn định thì h|m tích lũy ở (9.1b) được thay bởi:

(9.1d)

trong đó là hằng số chiết khấu. Khi có nghĩa rằng các khoản thưởng muộn c|ng ít ý nghĩa, khi thì ta chỉ quan tâm tới khoản thưởng ban đầu, còn khi lấy thì c{c gi{ trị thưởng ở mọi thời điểm đều quan trọng như nhau. Bạn đọc có thể mở rộng (9.1.c) cho trường hợp ngẫu nhiên.

*Ví dụ 2: Tác tử chơi cờ*

B}y giờ ta xét t{c tử l| chương trình chơi cờ ở bên A v| có đối thủ bên B (có thể l| một chương trình kh{c). Mỗi trạng th{i b|n cờ m| A chọn nước đi (thế cờ) sẽ l| một trạng th{i. Nhiệm vụ của t{c tử l| chọn nước đi (t{c động) để bên A thắng. Nếu bên A thắng thì mỗi nước đi được thưởng gi{ trị 100, thua thì lượng thưởng l| -100, còn hòa thì gi{ trị thưởng bằng không. Trong trường hợp n|y không chỉ lượng thưởng có tính ngẫu nhiên, không biết trước m| trạng th{i b|n cờ sau mỗi nước đi cũng có tính ngẫu nhiên vì còn tùy thuộc v|o nước đi của bên B. Ta giả thiết biết được c{c x{c suất chuyển trạng th{i cho mọi cặp trạng th{i v| t{c động . Trường hợp hòa thì số lần chuyển trạng th{i có thể xem l| vô hạn. Tương tự công thức (9.1c), h|m thưởng tích lũy có dạng tổng qu{t l|:

(9.1e)

B}y giờ ta có thể giới thiệu mô hình cho t{c tử v| một số kh{i niệm liên quan.

#### 9.1.2. Tác tử

Không chú ý tới cấu trúc vật lý của phần cứng, t{c tử l| một bộ ra quyết định được đặt trong một môi trường. Trong robot tìm đường đi, t{c tử l| robot v| môi trường l| hệ thống đường đi bao gồm c{c vị trí liên quan, còn trong chơi cờ thì t{c tử l| bộ chơi cờ còn môi trường l| b|n cờ.

Môi trường được đặc trưng bởi một tập trạng thái , trong chuỗi thời gian *t =1, 2,…..* robot có thể quan s{t được trạng th{i tương ứng của môi trường. Trong c{c ví dụ trên, trạng th{i của môi trường l| vị trí của robot hoặc thế cờ tại thời điểm đang xét.

Ở mỗi trạng thái , có một tập t{c động m| t{c tử có thể chọn. Nếu thời điểm *t* t{c tử ở trạng th{i v| thực hiện t{c động thì nó chuyển sang trạng th{i v| nhận gi{ trị thưởng *tức thời* , trong đó

tương ứng l| c{c h|m chuyển trạng th{i v| h|m gi{ trị thưởng x{c định trên đơn định hoặc ngẫu nhiên.

T{c tử sẽ bắt đầu ở trạng th{i v| tuần tự ở c{c trạng th{i thực hiện chuỗi t{c động tương ứng chuyển sang c{c trạng th{i và nhận c{c gi{ trị thưởng hữu hạn hay vô hạn tùy theo b|i to{n. Mặc dù ta gọi c{c gi{ trị thưởng là *thưởng tức thời* nhưngcó thể nó chưa x{c định được ngay sau khi thực hiện t{c động như trong ví dụ chơi cờ hay sạc pin. Mô hình một t{c tử được mô tả trong hinh 9.1.

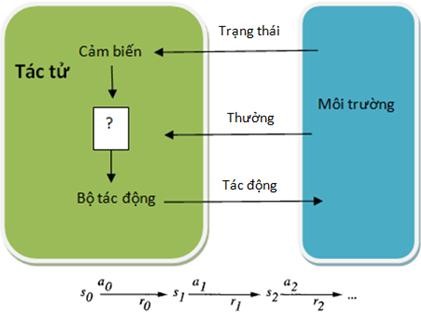
**Hình 9.1.**

Một tác tử tương tác với môi trường: bắt đầu ở trạng thái

thực hiện chuỗi tác động, chuyển trạng thái

và nhận

thưởng



C{c qu{ trình chuyển trạng th{i v| nhận thưởng n|y l| qu{ trình Markov, trong đó kết quả ở thời điểm *t* chỉ phụ thuộc v|o trạng th{i v| t{c động hiện thời m| không phụ thuộc v|o c{c trạng th{i v| t{c động ở những thời điểm trước đó

Ký hiệu và l| h|m x{c định *chính sách* (policy) chọn t{c động của tác tử khi ở trạng thái . Mục đích của bộ học l| tìm ra chính s{ch chọn t{c động tối ưu sao cho nếu xem t{c tử khởi đầu từ v| chọn t{c động theo chính sách này thì gi{ trị tích lũy thưởng đạt cực đại với mọi .

Kh{c với c{c b|i to{n học có gi{m s{t trước đ}y, t{c tử sẽ thực hiện c{c thực nghiệm hoặc thí nghiệm mô phỏng để dựa v|o đó m| tìm chính s{ch tối ưu. Qu{ trình học tăng cường thường có c{c đặc điểm sau.

*Thưởng trễ (delayed reward)*. T{c tử cần học một h|m mục tiêu  ánh xạ từ trạng thái hiện thời *s* v|o t{c động tối ưu *.* Theo cách tiếp cận trong học có giám sát trước đ}y thì khi học hàm ta thường giả thiết mỗi mẫu đ|o tạo là cặp . Tuy nhiên, trong học tăng cường thì thông tin như thế không có sẵn m| thay v|o đó bộ học chỉ nhận được một chuỗi gi{ trị thưởng tức thời tương ứng với chuỗi t{c động khi t{c tử thực hiện. Thậm chỉ c{c khoản thưởng này chỉ x{c định được khi kết thúc thử nghiệm như trong trường hợp chơi cờ hoặc đi sạc pin. Trong c{c chương trước, chúng ta học hàm mục tiêu π dựa trên các dữ liêu quan s{t được, trong đó mỗi ví dụ huấn luyện là một cặp mẫu .

*Khám phá (exploration).* Trong học tăng cường , t{c tử ảnh hưởng tới ph}n bố của c{c ví dụ huấn luyện bằng dãy t{c động m| nó chọn. Điều n|y đặt ra một c}u hỏi l| chính s{ch thí nghiệm n|o đem lại nhiều hiệu quả nhất. Bộ học cần sự c}n bằng giữa khám phá các trạng th{i v| t{c động chưa biết hay khai th{c c{c trạng th{i v| t{c động m| nó đã được học v| sẽ đem lại gi{ trị thưởng cao hơn.

*Các trạng thái quan sát được một phần*. Mặc dù có thể cho rằng c{c bộ cảm biến của t{c tử có thể nhận biết to|n bộ trạng th{i của môi trường trong mỗi thời điểm, nhưng trong nhiều trường hợp thực tế, c{c bộ cảm biến chỉ cung cấp c{c thông tin cục bộ. Chẳng hạn robot với camera phía trước không thể thấy được những gì ở sau nó hoặc bị khuất khi có chướng ngại vật. Trong c{c trường hợp như vậy, t{c tử cần phải kết hợp với những quan s{t trước đó với những dữ liệu hiện thời khi lựa chọn t{c động.

*Học cả đời*. Không giống với việc xấp xỉ h|m riêng lẻ, t{c tử thường phải học nhiều việc trong cùng một môi trường, sử dụng cùng một cảm biến. Chẳng hạn, một robot di động cần phải học l|m thế n|o để có thể tự nạp điện, l|m thế n|o để đi qua c{c h|nh lang hẹp, l|m thế n|o để lấy kết quả từ m{y in laser. Điều n|y đặt ra khả năng sử dụng những kinh nghiệm v| hiểu biết từ trước để giảm bớt sự phức tạp mẫu khi học c{c nhiệm vụ mới.

#### 9.1.3. Các bài toán học

Trong phần n|y chúng ta thiết lập b|i to{n tìm *chính sách/chiến lược điều khiển* tối ưu cho b|i to{n học tăng cường một c{ch rõ r|ng hơn dưới c{c dạng: đơn định v| ngẫu nhiên. C{c b|i to{n được thiết lập tổng qu{t dựa trên qu{ trình quyết định Markov (Markov decision process: MDP).

*Bài toán đơn định*

Xét một thế giới gồm t{c tử trong một môi trường có tập hữu hạn trạng th{i *S*, t}p c{c t{c động có thể của t{c tử , trong đó c{c gi{ trị thưởng tức thời v| trạng th{i chuyển l| đơn định với mọi cặp

sử tại thời điểm *t* t{c tử ở trạng th{i và thực hiện chiến lược điều khiển thì thu được gi{ trị tích lũy thưởng:

. Giả

(9.2a)

trong đó l| gi{ trị chiết khấu. Đại lượng được gọi là *giá trị tích lũy thưởng* đạt được từchiến lược điều khiển/chính sách v| trạng th{i ban đầu . Lưu ý rằng để (9.2a) có nghĩa trong trường hợp có trạng th{i hút ta quy ước ở trạng th{i n|y chỉ có t{c động giữ nguyên trạng th{i v| gi{ trị thưởng tức thời bằng không (hoặc bằng m nếu l| chuyển từ b|i to{n cực tiểu chi phí với hằng số m). Công thức (9.2a) cũng trùng với (9.1d) khi xem l| trạng th{i khởi đầu .

Nhiệm vụ của b|i to{n học tăng cường đơn định l| tìm *chiến lược điều khiển* (chính s{ch) tối ưu sao cho với mọi trạng th{i ở thời điểm *t* v| thực hiện t{c động theo chính s{ch n|y thì ta đều thu được gi{ trị tích lũy thưởng cực đại, tức l|:

(9.2b)

Để đơn giản, ta sẽ dùng ký hiệu thay cho ký hiệu gi{ trị tích lũy thưởng của chính s{ch tối ưu. V\*(s) cho gi{ trị thưởng tích lũy có chiết khấu cực đại

m| t{c tử tích lũy được khi khởi đầu ở trạng th{i s.

Hình 9.2a mô tả thế giới có một môi trường chia ô đơn giản. S{u ô lưới trong biểu đồ mô tả s{u trạng th{i hay vị trí của t{c tử, ô có nhãn G l| trạng th{i hút. Mỗi mũi tên trong biểu đồ biểu thị một t{c động m| t{c tử có thể thực hiện để di chuyển từ trạng th{i n|y sang trạng th{i kh{c. C{c t{c động có thể l| sang tr{i, sang phải v| lên, xuống trong phạm vi của s{u ô n|y. Chữ số kết hợp với mỗi mũi tên mô tả gi{ trị thưởng tức thời t{c tử nhận được khi nó thực hiện c{c t{c động tương ứng. Lưu ý rằng gi{ trị thưởng tức thời trong môi trường đặc biệt n|y được đặt bằng 0 đối với mọi t{c động chuyển trạng th{i, ngoại trừ những trạng th{i được g{n nhãn G. Dễ d|ng thấy rằng trong trường hợp này, trạng thái G là trạng th{i đích bởi vì đó l| c{ch duy nhất tác tử có thể nhận được giá trị thưởng là chuyển đến trạng th{i đó. C{c gi{

tr

ị

tích lũy thư

ở

ng

ứ

ng v

ớ

i giá tr

ị

chi

ế

t kh

ấ

u

được cho trong hình 9.2c.

**Hình 9.2.**

Một thế giới đơn

định đơn giản. Mỗi ô biểu thị một trạng thái, giá trị thưởng bằng 100 nếu chuyển vào

trạng thài G và bằng 0 trong các tường hợp khác. Các giá trị

và

v

ớ

i chi

ế

t kh

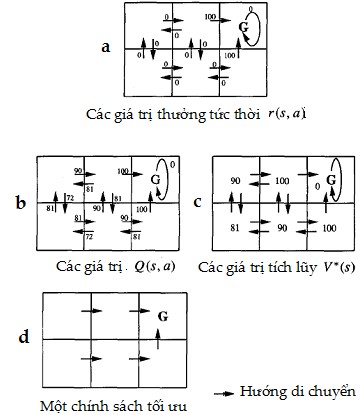
ấ

u

.

c

hỉ ra một chính sách (chiến lược điều khiển) tối ưu



Một chính s{ch tối ưu cho b|i to{n n|y được chỉ ra trong hình 9.2d, ở đó chỉ rõ một t{c động m| t{c tử sẽ lựa chọn trong mỗi trạng th{i môi trường. Chính s{ch tối ưu hướng t{c tử theo con đường ngắn nhất để đạt đến trạng th{i G.

Chú ý rằng V\* được định nghĩa l| tổng của c{c gi{ trị thưởng có chiết khấu trong tương lai vô hạn. Trong môi trường đặc biệt n|y, chỉ một lần t{c tử chuyển tới trạng th{i hút G, tương lai vô hạn của nó chính l| duy trì trong trạng th{i n|y v| nhận c{c gi{ trị thưởng bằng không.

*Bài toán ngẫu nhiên*

Trong b|i to{n ngẫu nhiên, với mỗi trạng th{i v| t{c động giá trị thưởng v| trạng th{i kế tiếp không đơn định m| có tính ngẫu nhiên. Ph}n bố x{c suất của chúng có thể đã biết hoặc sẽ được x{c định qua thực nghiệm. Tương tự như trường hợp đơn định, với mỗi chiến lược điều khiển h|m mục tiêu lúc n|y l| kỳ vọng của gi{ trị thưởng tích lũy như trong (9.1f) với trạng

(9.3a)

Như vậy, yêu cầu của b|i to{n học tăng cường l| tìm chiến lược điều khiển tối ưu

th{i ban đầu

:

sao cho:

với mọi trạng th{i

(9.3b)

ở thời điểm *t.*

**Chú ý**. Trong c{c b|i to{n học tăng cường, c{c t{c động v| h|m chuyển trạng th{i l| yếu tố nội tại quyết định bản chất b|i to{n, còn gi{ trị thưởng tức thời có vai trò l| yếu tố bình luận, đ{nh gi{ t{c động v| bộ học có thể x}y dựng h|m thưởng phù hợp với mục đích hoạt động của t{c tử. Chẳng hạn, gi{ trị thưởng tức thời v| do đó gi{ trị tích lũy thưởng ở cả hai b|i to{n ở mục 9.1.1 đều do bộ học quyết định để đạt mục đích tìm đường đi tối ưu hay chiến lược cho bộ chơi cờ thắng.

### 9.2. HỌC Q (Q learning)

#### 9.2.1. Học Q trong các bài toán đơn định

Trong học tăng cường, chúng ta không có được c{c mẫu dạng (*s,a*) nên rất khó học trực tiếp chiến lược điều khiển tối ưu . Thay v|o đó, bộ học chỉ dựa vào thông tin từ các chuỗi giá trị thưởng tức thời trên thực nghiệm.

Giả sử đã tìm được chiến lược điều khiển tối ưu . Khi đó với mỗi trạng

th{i ban đầu

, ta nhận được chuỗi t{c động tối ưu v| gi{ trị tích lũy cực đại

theo công thức (9.2a). Biểu thức tính

được biểu diễn lại l|:

(9.3c)

Từ đó ta có:

(9.3d)

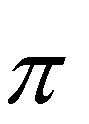
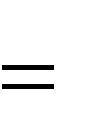
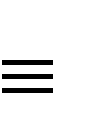
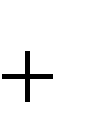
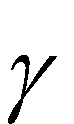
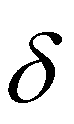
Như vậy, t{c động tối ưu ở trạng th{i *s* l| t{c động *a* l|m cực đại tổng gi{ trị thưởng tức thời *r*(*s,a*) với gi{ trị của trạng th{i kế tiếp có chiết khấu γ. T{c tử có thể học được chính s{ch tối ưu thông qua học nhờ được cho to|n bộ thông tin về gi{ trị h|m thưởng *r* v| h|m chuyển trạng th{i *δ*. Tuy nhiên trong nhiều bài toán thực tế, c{c h|m *r* và *δ* của môi trường chưa biết trước m| cần x{c định qua thực nghiệm. Thay cho ta sẽ dùng một h|m có thể ước lượng để x{c định .

*9.2.1.1. Hàm Q*

Chúng ta định nghĩa h|m *Q*(*s,a*) l| gi{ trị thưởng tức thời nhận được ở trạng th{i ban đầu *s* với t{c động *a* cộng với gi{ trị tích lũy thướng tối ưu sau đó (có chiết khấu ):

*Q*(*s*,*a*) *r*(*s*,*a*) *V* \* ( (*s*,*a*)) (9.4)

Lưu ý rằng l| lượng đúng được cực đại biểu thức trong (9.3d) để chọn t{c động tối ưu *a* ở trạng thái *s*. Do đó ta có thể viết lại (9.3d) theo như sau:



\* (*s*) arg max *Q*(*s*,*a*) (9.5)

*a*

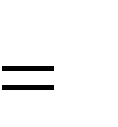
Việc viết lại n|y rất quan trọng, vì nó chỉ ra rằng nếu t{c tử học theo h|m Q thay vì V\* thì nó có thể chọn được c{c t{c động tối ưu kể cả khi nó không biết h|m *r* và *δ*. Công thức (9.5) chỉ ra rằng bộ học chỉ cần quan tâm tới mỗi t{c động *a* ứng với trạng thái hiện tại *s* và lựa chọn t{c động làm đạt cực đại.

Về sau ta sẽ thấy t{c tử có thể chọn một chuỗi t{c động tối ưu to|n cục bằng c{ch t{c động lặp lại c{c gi{ trị cục bộ của Q đối với trạng th{i hiện thời. Có nghĩa l| t{c tử có thể chọn c{c t{c động tối ưu m| không phải quan t}m tới c{c trạng th{i dẫn đến kết quả gì từ t{c động.

Hình 9.2b chỉ ra giá trị đối với mỗi trạng thái v| t{c động trong một thế giới lưới đơn giản, c{c gi{ trị được cho ở hình 9.2c, còn một chiến lược tối ưu được cho ở hình 9.2d.

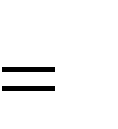
*9.2.1.2. Một thuật toán học Q*

Công thức (9.5) ở trên cho ta thấy việc học h|m *Q* sẽ cho phép học được chiến lược tối ưu tối ưu . Vậy l|m thế n|o để học h|m *Q*? Watkins (1989) đề xuất thuật to{n lặp để ước lượng gần đúng *Q.* Để hiểu rõ thuật to{n, ta chú ý tới mối quan hệ gần gũi giữa *Q* và V\*,:

*V* \* (*s*)max*Q*(*s*,*a*') .

##### *a*'

Quan hệ n|y cho phép viết lại công thức (9.4) như sau : *Q*(*s*,*a*) *r*(*s*,*a*) max*Q*((*s*,*a*),*a*') (9.6)



##### *a*'

C{ch x{c định *Q* theo đệ quy n|y l| cơ sở của thuật to{n xấp xỉ lặp Q. Để mô



tả thuật to{n, ta dùng ký hiệu *Q* để để biểu diễn ước lượng hiện thời cho h|m *Q*



trong qu{ trình học. Trong thuật to{n n|y, bộ học đưa ra giả thuyết *Q* của nó bằng một bảng lớn với mỗi phần tử biểu thị cho một cặp trạng th{i-t{c động. Mỗi phần tử



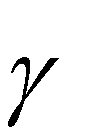
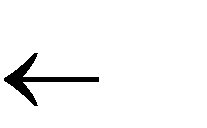
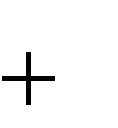
ứng với mỗi cặp (s,a) của bảng lưu trữ gi{ trị *Q*(*s*,*a*) của giả thuyết hiện thời cho gi{ trị h|m học chưa biết *Q(s,a)*. Bảng n|y có thể được khởi tạo ngẫu nhiên , dễ hiểu nhất l| khởi tạo bằng 0 cho mọi cặp trạng th{i-t{c động. C{c bước trong thuật to{n sẽ được mô tả cho ba trường hợp: 1) tổng qu{t, 2) môi trường có trạng thái hút và mọi chuỗi t{c động đều dẫn tới trạng thái này, 3) Tồn tại một chuỗi trạng thái-t{c động vô hạn sao cho mỗi trạng thái v| t{c động được lặp lai vô hạn lần khi độ d|i chuỗi ra vô hạn.

#### *Thuật toán cho trường hợp tổng quát*

Bộ học x{c định một phương thức lấy ngẫu nhiên các trạng thái sao cho mỗi cặp (*s*, *a*) tùy ý đều có x{c suất được chọn kh{c không. Trong trường hợp đó, bộ học thực hiện lặp việc chọn ngẫu nhiên cặp t{c động v| quan s{t gi{ trị kết quả thưởng . Sau đó cập nhật lại bảng lưu trữ của theo quy tắc:

và

*Q*(*s*,*a*) *r* max*Q*(*s*',*a*') (9.7)



##### *a*'



Chú ý rằng quy tắc huấn luyện n|y sử dụng c{c gi{ trị *Q* hiện thời của t{c tử



đối với trạng th{i mới s’ để cải thiện ước lượng của *Q*(*s*,*a*) đối với trạng th{i s trước đó. Trong thuật to{n n|y, mặc dù biểu thức tính *Q(s,a)* (9,6) mô tả *Q* qua các hàm *δ(s,a)* và *r(s,a)*, nhưng bộ học không cần thiết phải biết c{c h|m tổng qu{t n|y khi {p dụng luật huấn luyện theo công thức (9.7). Thay v|o đó, nó thực hiện t{c động trong môi trường rồi quan s{t trạng th{i tiếp theo *s’* v| gi{ trị thưởng. Vì thế, có thể xem l| thực hiện thủ tục lấy mẫu cho c{c h|m n|y ở những gi{ trị hiện thời của s v| a.

Thuật to{n học *Q* trên đ}y đối với c{c qu{ trình quyết định Markov đơn định được miêu tả chính x{c hơn ở bảng 9.1.

**Bảng 9.1.** Thuật toán học Q đơn định với hệ số chiết khấu 

|  |
| --- |
| Bước 1. Khởi tạo*Q*(*s*,*a*) bằng *0* cho mỗi cặp *s, a*  Bước 2. Thực hiện lặp:  2.1. Lấy ngẫu nhiên  2.2. Lấy ngẫu nhiên  ;    2.3. X{c định và trạng thái kế tiếp ;  2.4. Cập nhật bảng *Q*(*s*,*a*) theo quy tắc (9.7) |

*Điều kiện dừng.*

Về lý thuyết, với một số điều kiện sẽ nói về sau, khi bước 2 thực hiện vô hạn lần



thì *Q*(*s*,*a*) hội tụ đến *Q(s,a)*. Tuy nhiên, trong thực h|nh thì thuật to{n chỉ thực hiện được một số hữu hạn bước. Vì vậy thuật to{n dừng mỗi khi c{c cặp đã được {p



dụng khắp lượt m| bảng *Q*(*s*,*a*) không đổi hoặc độ lệch hiệu chỉnh lớn nhất nhỏ hơn ngưỡng cho trước.

Mặt kh{c, trong thuật to{n trên, việc lấy ngẫu nhiên cặp có thể dẫn đến



hiện tượng bảng *Q*(*s*,*a*) không được điều chỉnh trong nhiều lần lặp ở bước 2. Vì vậy với c{c trường hợp đặc biệt, ta có c{c thuật to{n phù hợp hơn

#### *Thuật toán cho môi trường có trạng thái hút*

Ta xét trường hợp thế giới m| môi trường có trạng thái hút và mọi trạng thái khởi tạo đều dẫn tới trạng th{i n|y. Khi đó thuật toán sẽ thực hiện lặp c{c bước 2.3 và 2.4 ở trên khi thay bởi trạng thái kế tiếp . Thuật to{n được mô tả chi tiết trong bảng 9.2.

**Bảng 9.2**. Thuật toán học Q đơn định khi có trạng thái hút

Bước 1. Khởi tạo

)

,

(

*a*

*s*

*Q*



bằng

*0*

cho mỗi cặp

*s, a*

Bước 2. Thực hiện lặp:

2.1

. L

ấ

y ng

ẫ

u nhiên

;

. Th

2.2

ự

c hi

ệ

n l

ặ

p đ

ế

n khi

l| trạng th{i hút

2.2.1

. Lấy ngẫu nhiên

2.2.2. X{

c đ

ị

nh

và tr

ạ

ng thái k

ế

ti

ế

p

;

2.2.3

. Cập nhật bảng

)

,

(

*a*

*s*

*Q*



theo quy tắc (9.7);

.

2.2.4

Trường hợp n|y h|m



*Q*

s

ẽ

h

ộ

i t

ụ

t

ớ

i

sau hữu hạn bước. Trong thực h|nh,



thuật to{n dừng khi *Q*(*s*,*a*) không đổi hoặc độ lệch hiệu chỉnh lớn nhất nhỏ hơn ngưỡng cho trước.

#### *Thuật toán cho trường hợp có chuỗi lặp vô hạn lần*

Trường hợp không có trạng thái hút và tồn tại một chuỗi trạng thái-t{c động vô hạn sao cho mỗi trạng thái v| t{c động được lặp lai vô hạn lần. Khi độ d|i chuỗi ra vô hạn, ta có thể {p dụng thuật to{n sau.



Sau khi khởi tạo bảng gi{ trị *Q*(*s*,*a*) , bộ học thực hiện lặp việc lấy ngẫu nhiên trạng th{i ban đầu v| thực hiện chuỗi t{c động để nhận được chuỗi trạng th{i



kết thúc ở trạng th{i hút. Sau mỗi t{c động, bảng *Q*(*s*,*a*) được cập nhật v| thực hiện như ở c{c bước 2.2.2-2.2.4 ở trên. Thủ tục lặp x}y dựng ngẫu nhiên chuỗi trạng th{i



thực hiên cho đến khi mỗi trạng th{i của *S* đều thuộc ít nhất một chuỗi m| *Q*(*s*,*a*) không đổi hoặc độ lệch hiệu chỉnh lớn nhất nhỏ hơn ngưỡng cho trước. Chi tiết thuật to{n được mô tả trong bảng 9.3.

**Bảng 9.3.** Thuật toán học Q đơn định cho trường hợp có chuỗi lặp vô hạn lần

|  |
| --- |
| Bước 1. Khởi tạo*Q*(*s*,*a*) bằng *0* cho mỗi cặp *s, a*  Bước 2. Lấy ngẫu nhiên ;  Bước 3. Thực hiện lặp đến khi dừng:  3.1. Lấy ngẫu nhiên  3.2. X{c định và trạng thái kế tiếp ;  3.3. Cập nhật bảng *Q*(*s*,*a*) theo quy tắc (9.7);  3.4. |

#### *Ví dụ minh họa*

Trong hoạt động của thuật to{n học *Q* ở trên, điểm then chốt l| thủ tục cập nhật



bảng gi{ trị của *Q* sau mỗi t{c động được chọn. Để minh họa, ta xét b|i to{n đơn giản với c{c gi{ trị thưởng tức thời cho trong hình 9.2a. Tại thời điểm đang xét t{c tử ở ô



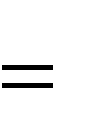
1.1 v| gi{ trị liên quan của ước lượng *Q* được biểu diễn trong hình 9.3a, với hệ số chiết khấu 0.9. Lúc n|y gi{ trị ước lượng *Q*(*s* ,*a* động *aright* đi sang phải như mô tả trong hình 9.3 v| nhận được gi{ trị thưởng *r* 0.



72

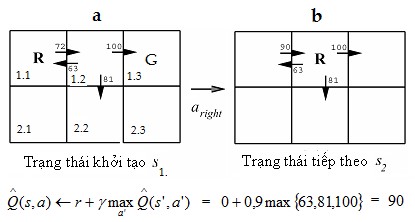
)

1



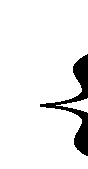
*right*

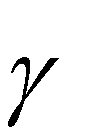
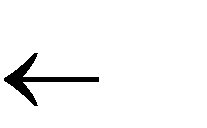
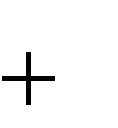
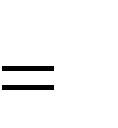
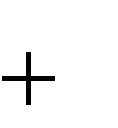
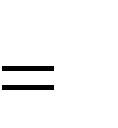
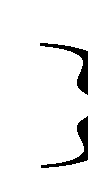
. T{c tử thực hiện t{c



**Hình 9.3**. a) Tác tử ở trạng thái *s*1 ở ô 1.1 và các ước lượng *Q* hiện thời. b) Cập nhật *Q* sau khi tác tử thực hiện tác động sang phải.

Sau đó nó thực hiện cập nhật gi{ trị *Q s a*ˆ( 1, *right* ) theo quy tắc (9.7) dựa trên ước lượng đã có của *Q*ˆ cho trạng th{i mới *s*2 trong ô 1.2:

 *Q*(*s*,*a*) *r* max*Q*(*s*',*a*') 0 0,9max 63,81,10090 .



##### *a*'



Theo quy tắc huấn luyện n|y, một ước lượng *Q* mới cho t{c động chuyển trạng



th{i sẽ l| tổng của gi{ trị thưởng thu được (0/100) cộng với gi{ trị *Q* cao nhất liên quan với trạng th{i kết quả (100) được chiết khấu với hệ số  (0.9). Sau khi điều chỉnh *Q*ˆ ở ô 1.1, t{c tử ở trạng th{i mới *s*2, nó chọn tiếp t{c động v| cập nhật tăng cường

*Q*ˆ cho đến khi nó ở trạng th{i hút (ô 1.3) thì kết thúc một bước lặp 3 trong bảng 9.2. Nếu điều kiện dừng chưa thỏa mãn thì nó chọn ngẫu nhiên trạng th{i khởi tạo mới v| thực hiện hiện c{c t{c động cho tới khi đạt tới được trạng th{i đích mong muốn v| kết thúc lần lặp mới.

Hình 9.2d mô tả một chiến lược điều khiển tối ưu cho b|i to{n đang xét với c{c gi{ trị cập nhật cuối cùng của *Q*ˆ = *Q* cho trong hình 9.2c. Người đọc có thể tìm ra chính s{ch tối ưu kh{c nhờ sửa đổi c{ch chọn t{c động (chiến lược thực nghiệm) cho mỗi trạng th{i. Trước khi thảo luận về chiến lược thực nghiệm, ta giới thiệu một chứng minh tính hội tụ cho thuật to{n học *Q* tổng qu{t trong bảng 9.1.

#### *Sự hội tụ*

Để chứng minh tính hội tụ cho c{c thuật to{n trên, ta giả thiết hệ thống thỏa mãn ba điều kiện:

Trước hết ta giả thiết rằng hệ thống l| một MDP đơn định.



Thứ hai: hệ số chiết khấu v| c{c gi{ trị thưởng ở mỗi bước bị chặn, tức l| tồn tại hằng số n|o đó *c* sao cho với mọi trạng th{i *s* và tác động *a* ta có: *r s a*( , ) *c* .



Thứ ba: thuật toán không dừng và theo thời gian tác tử chọn thăm mỗi cặp trạng thái- t{c động nhiều vô hạn lần.



Điều kiện thứ ba đòi hỏi rằng theo thời gian, mọi t{c động ở trạng th{i *s* đều được t{c tử lặp lại việc thực hiện nó với tần suất kh{c không khi độ d|i của dãy t{c động dần ra vô hạn.

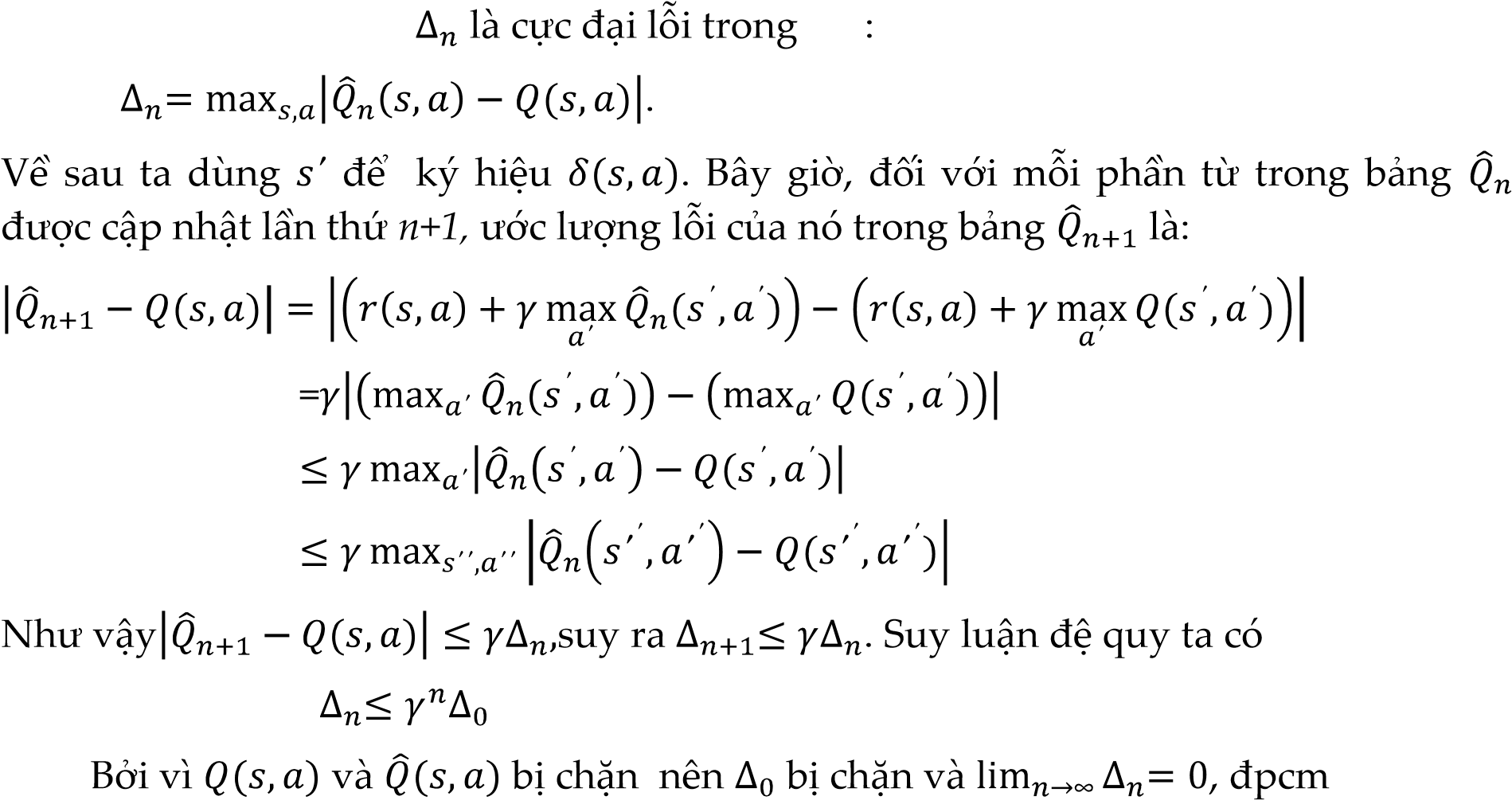
**Chú ý**. Với mọi chiến lược điều khiển và trạng th{i ban đầu , gi{ trị tích lũy thưởng x{c định theo công thức (9.2a) khi , ta có đ{nh gi{:

.

Như vậy hội tụ tuyệt đối và bị chặn với mọi và v| do đó cũng bị chặn với cùng hằng số. Từ công thức (9.4) ta suy ra cũng bị chặn với mọi cặp . Tương tự ta cũng có c{c ước lượng của bị chặn.

B}y giờ ta ký hiệu *Q s a*ˆ*n*( , ) l| gi{ trị tương ứng ở bảng *Q*ˆkhi nó được cập nhật lần thứ *n*. Khi đó ta có định lý sau.

**Định lý 9.1.** *Nếu thế giới của tác tử thỏa mãn ba điều kiện nêu trên thì Q s a*ˆ*n*( , ) *sẽ hội tụ về Q s a*( , ) *khi n*  *với mọi cặp giá trị s a*, ***.***

*Chứng minh*. Bởi vì mỗi cặp trạng thái-t{c động nhiều vô hạn lần nên ta xét c{c khoảng m| mỗi cặp đều xảy ra ít nhất một lần. Giả sử l| bảng ước lượng *Q* sau *n* kỳ cập nhật như thế v|

#### *Chiến lược thí nghiệm*

B}y giờ ta thảo luận về chiến lược chọn trạng th{i khởi tạo v| t{c động của t{c tử trong c{c phiên bản của thuật to{n m| ở trên chưa chỉ rõ.

Một chiến lược tự nhiên là bộ học chọn trạng thái theo ph}n bố đều, c{c t{c động thì chọn t{c động *a* có *Q s a*ˆ( , ) lớn nhất, nhờ đó khai th{c tính xấp xỉ *Q*ˆhiện thời của nó. Tuy nhiên, chiến lược n|y l|m cho t{c tử l|m giảm tính kh{m ph{ v| không đảm bảo điều kiện “*mỗi cặp trạng thái-tác động được lặp vô hạn lần”* của định lý hội tụ*.* Vì lý do này, người ta thường dùng c{ch tiếp cận ngẫu nhiên để chọn t{c động thuật to{n học *Q* . Những t{c động có gi{ trị *Q*ˆ cao sẽ được g{n cho một x{c suất cao hơn, nhưng mọi t{c động đều được g{n một x{c suất kh{c 0. Một trong c{c c{ch g{n x{c suất l|:

*Q s a*ˆ( , *i* )

ˆ

(

,

)

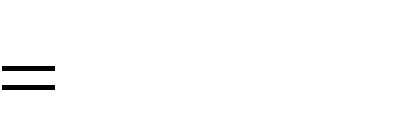
*j*

*Qsa*

*j*

*k*

*k*



*P a s*( *i* | ) (9.8)

trong đó *P a*( *i* | )*s* l| x{c suất lựa chọn t{c động *a* khi t{c tử ở trạng th{i ( *s* , *k>1* (có thể chọn *k =e*, l| hằng số) thể hiện mức độ ưu tiên lựa chọn c{c t{c động. Theo c{ch đó, t{c tử có thể sẽ lựa chọn cả những t{c động m| hiện tại chưa có được gi{ trị *Q*ˆ lớn. Trong một số trường hợp *k* được tổng hợp từ nhiều kết quả kh{c nhau m| t{c tử đã thu được từ c{c trạng th{i trước của qu{ trình học v| chính s{ch t{c động sẽ theo đó m| từng bước được thiết lập.

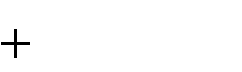
##### **9.2.2. Học Q trong các bài toán ngẫu nhiên**

Ở trên chúng ta đã đề cập đến thuật to{n học *Q* trong môi trường đơn định.

Trong phần n|y chúng ta sẽ đề cập tới trường hợp ngẫu nhiên (không đơn định), trong đó h|m thưởng *r s a*( , ) v| h|m chuyển trạng th{i (*s a*, ) không phụ thuộc v|o c{c trạng th{i v| t{c động trước đó. Nói c{ch kh{c hệ thống l| qu{ trình quyết định Markov không đơn định.

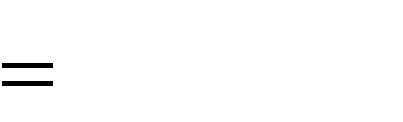
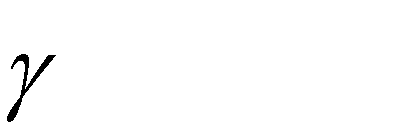
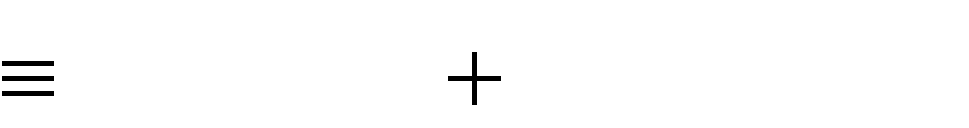
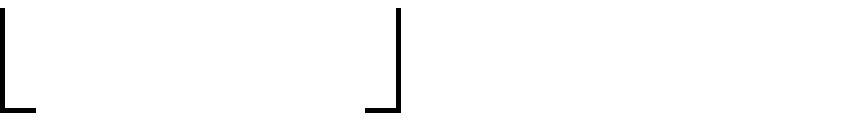
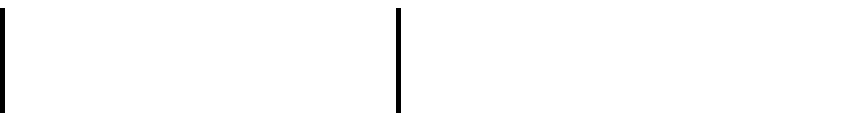
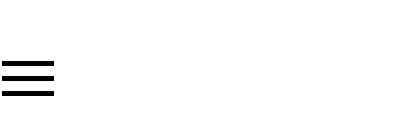
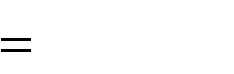
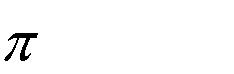
Để ph{t triển thuật to{n học *Q* ở trên cho trường hợp MDP không đơn định, ta xem lại h|nh trình để dẫn tới thuật to{n trong hợp đơn định v| sửa đổi những chỗ cần thiết. Trở lại với h|m mục tiêu của trường hợp ngẫu nhiên, gi{ trị tích lũy thưởng

*V* cho một chiến lược điều khiển  x{c định theo biểu thức (9.3a)

 *V* ( )*st E irt i* , trong đó, dãy gi{ trị thưởng *rt i* được sinh ra bởi chính s{ch 

###### *i* 0

với trạng th{i khởi đầu l| . Chiến lược điều khiển tối ưu được biểu thị bởi biểu thức (9.3b) . Ta cần tổng qu{t hóa định nghĩa trước đ}y về *Q* từ công thức (9.4) qua gi{ trị kỳ vọng của nó.



*Q s a*( , ) *E r s a*[ ( , ) *V* \*( (*s a*, ))]

*E r s a*[ ( , )] *E V*[ \*( (*s a*, ))]

*E r s a*[ ( , )]+ *P s*( '| ,*s a V*) \*(*s*') (9.9a)

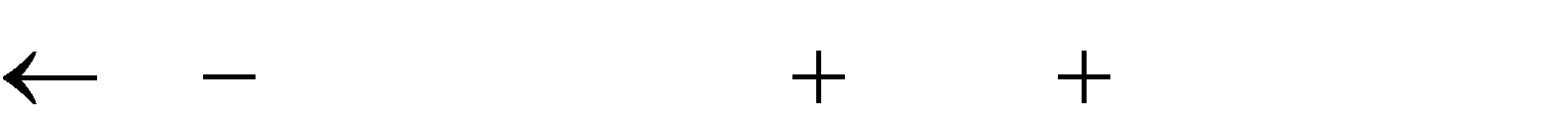
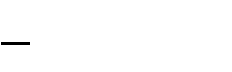
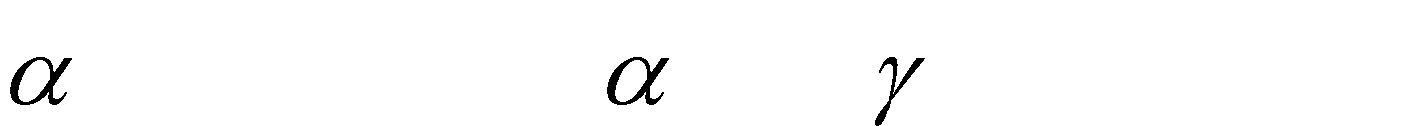
###### s'

trong đó *P s s a*( '| , ) l| x{c suất trạng th{i tiếp theo l| *s*' nếu thực hiện t{c động *a* tại trạng th{i *s* . Như vậy ta có thể biểu diễn *Q* một c{ch đệ quy nhờ tổng qu{t hóa công thức (9.6)

(9.9b)

Để ước lượng *Q*, ta không dùng được công thức (9.7) vì nó không hội tụ trong trường hợp không đơn định. Chẳng hạn, xét h|m thưởng không đơn định *r s a*( , ) sinh ra c{c gi{ trị kh{c nhau mỗi khi thực hiện lặp lại việc chuyển trạng th{i *s a*, . Khi đó, luật huấn luyện sẽ lặp lại việc thay đổi gi{ trị của *Q s a*ˆ( , ) , ngay cả khi ta khởi tạo bảng gi{ trị *Q*ˆ bằng đúng gi{ trị đúng của *Q* . Trong trường hợp n|y, luật huấn luyện sẽ không hội tụ. Khó khăn n|y được khắc phục như sau. Ta dùng ký hiệu *Q*ˆ*n* để biểu thị cho ước lượng của t{c tử tại bước lặp thứ *n* của thuật to{n, khi đó luật huấn luyện hiệu chỉnh sau đ}y đủ để đảm bảo cho *Q*ˆ hội tụ về gi{ trị *Q* :

*Q s a*ˆ*n*( , ) (1 *n*)*Q*ˆ*n* 1(*s a*, ) *n*[*r* max *Q*ˆ*n*-1(*s a*', ')] (9.10a)

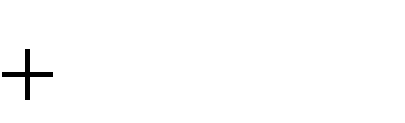
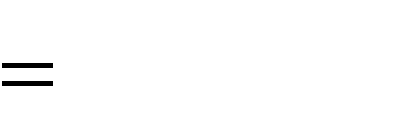
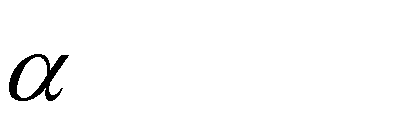


*a*'

với:

1

*n*  (9.10b)



1 *v*isits ( , )n *s a*

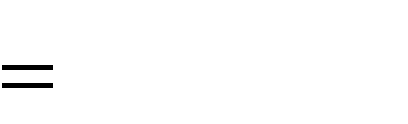
trong đó *s* và *a* l| trạng th{i v| t{c động đã được cập nhật cho đến bước lặp *n* , và *v*isits ( , )n *s a* l| tổng số lần cặp trạng th{i-t{c động n|y được thăm v| thực hiện tính tới

hết bước lặp thứ *n* .

Tư tưởng chính trong công thức huấn luyện hiệu chỉnh l| hiệu chỉnh gi{ trị *Q*ˆ một c{ch chậm hơn so với trường hợp hệ đơn định. Chú ý l| nếu ta đã đặt trong công thức (9.10), ta sẽ thu được luật huấn luyện cho trường hợp hệ đơn định. Với  có gi{ trị nhỏ, số hạng n|y sẽ trở th|nh lấy trung bình với gi{ trị *Q s a*ˆ( , ) hiện tại để sinh ra một gi{ trị cập nhật mới. Chú ý l| gi{ trị của trong công thức (9.10b) sẽ giảm khi *n* giảm, v| như vậy mức độ hiệu chỉnh của qu{ trình huấn luyện l| nhỏ. Bằng việc giảm n ở một mức độ phù hợp trong qu{ trình huấn luyện, ta có thể đạt được sự hội tụ về gi{ trị *Q* . Việc lựa chọn như ở trên l| một trong nhiều c{ch để đảm bảo điều kiện cho sự hội tụ, dựa trên định lý của Watkins v| Dayan (1992).

1

*n*



*n*



*n*



***Sự hội tụ của thuật toán học*** *Q* ***cho MDP không đơn định*.**

Xét một bộ học thực hiện học *Q* trong một MDP không đơn định với c{c gi{ trị thưởng bị chặn: ( *s a*, ) | *r s a*( , ) | *c*. Bộ học *Q* sử dụng luật học (9.10a) với bảng gi{ trị

*Q s a*ˆ( , ) được khởi tạo bằng những gi{ trị hữu hạn tùy ý v| sử dụng hệ số chiết khấu  thỏa mãn 0 1. Gọi *n i s a*( , , ) l| bước lặp ứng với lần thứ *i* t{c động *a* tại trạng



thái *s* được thực hiện thì khẳng định sau đúng.

**Định lý 9.2.** Nếu mỗi cặp trạng th{i – t{c động được thăm vô hạn lần , 0 1 và



thì với mọi cặp *s* và *a* , ta có *Q s a*ˆ*n*( , ) , với x{c suất bằng 1.

(

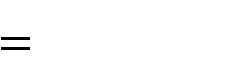
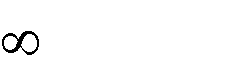
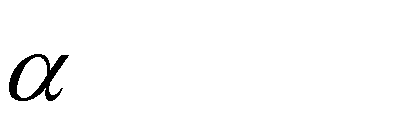
)

,,

1

*nisa*

*i*



**,**

2

(

,,

)

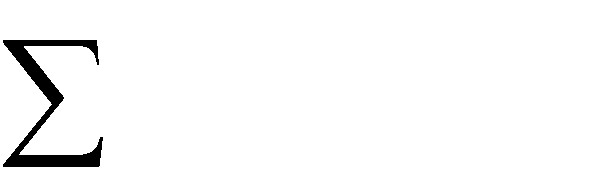
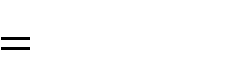
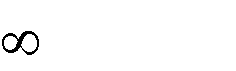
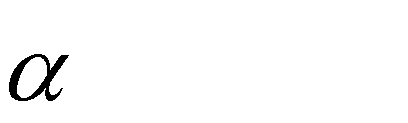
1

]

[

*nisa*

*i*

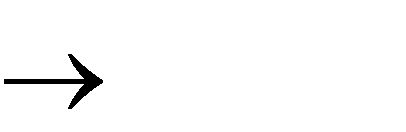


(

,

)

*Qsa*



khi

*n*



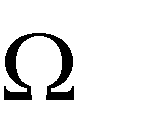
Trong thực tế thuật to{n học *Q* trong học tăng cường thường phải lặp lại nhiều nghìn lần lặp huấn luyện mới hội tụ. Để l|m ví dụ ứng dụng, dưới đ}y dưới thiệu phương ph{p tối ưu đ|n kiến giải c{c b|i to{n tối ưu tổ hợp.

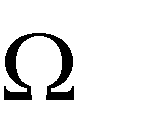
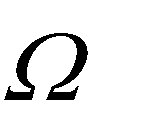
### 9.3. PHUƠNG PHÁP TỐI ƢU ĐÀN KIẾN (ACO)

Như đã trình b|y ở trên, phương ph{p họ Q dựa trên ý tưởng thực nghiệm mô phỏng hệ thống nhiều lần để quan s{t hệ thống v| gi{ trị thưởng tức thời được xem l| bình luận được lượng hóa cho mỗi t{c động, giá trị tích lũy thưởng biểu thị bình luận tổng hợp cho mỗi thí nghiệm. Trong mỗi thí nghiệm, các giá trị thường xa với thời điểm quan sát hiên thời sẽ chịu lượng chiết khấu lớn giải theo h|m mũ ( theo khoảng tới thời điểm hiện tại.

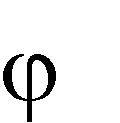
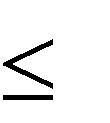
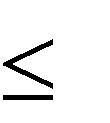
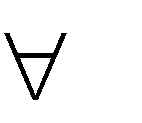
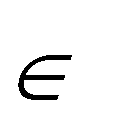
Trong tự nhiên, trên đường đi, mỗi con kiến tiết ra một hóa chất gọi l| vết mùi (pheromone trail) v| theo vết mùi của c{c con kiến kh{c để x{c định đường đi, lối rẽ có vết mùi c|ng cao thì khả năng kiến chọn c|ng lớn. Phương ph{p tối ưu đ|n kiến (ant colony optimization: ACO) do Dorigo đề xuất (1991) mô phỏng c{ch tìm đường đi của đ|n kiến để giải c{c b|i to{n tối ưu tổ hợp l| ứng dụng của ý tưởng học tăng cường n|y. Thuật to{n tìm đường đi của robot trong chương 1 l| một thể hiện ứng dụng của ACO. Trước hết ta ph{t biểu b|i to{n tối ưu tổ hợp tổng qu{t

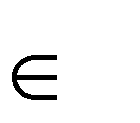
#### 9.3.1. Phát biểu bài toán tối ƣu tổ hợp tổng quát

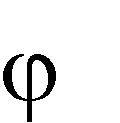
Xét b|i to{n cực tiểu ho{ (*S,f, ),* trong đó *S* l| tập hữu hạn trạng th{i*, f* là hàm mục tiêu x{c định trên *S* còn l| c{c r|ng buộc để x{c định S qua c{c th|nh phần của tập hữu hạn *C* v| c{c liên kết của tập n|y. C{c tập *S,C* và có c{c đặc tính sau.



1. *C*={*c1,...,cn*} l| tập hữu hạn gồm *n* th|nh phần. Ta ký hiệu *X* l| tập c{c dãy trong *C* độ d|i không qu{ *h*: *X*={<*u0,...,uk*>/ *ui C* *i* k *h*}.



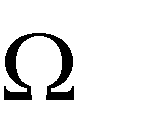
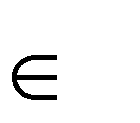
1. Tồn tại tập con *X\** của *X* v| {nh xạ từ *X\** lên *S* sao cho (s) không rỗng với mọi s S . Trong đó tập *X\** có thể x}y dựng được từ tập con *C0* n|o đó của *C* nhờ mở rộng tuần tự theo đặc tính 3 dưới đ}y.

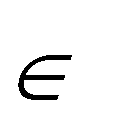


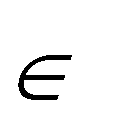
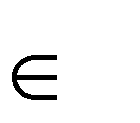
-

1

1. Từ *C0* mở rộng được th|nh *X\** theo thủ tục tuần tự:
   1. *x0*= <*u0*> l| mở rộng được với mọi *u*1 *C0*.
   2. Nếu *xk*=<*u1,...,uk*> l| mở rộng được thì tồn từ x{c định được tập con *J*(*xk*) của



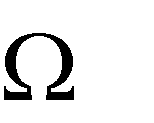
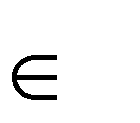
*C* sao cho với mọi *uk+1* *J*(*xk*) thì *xk+*1=<*u0,...,uk,uk+1*> l| mở rộng được hoặc *xk X\** khi J(*xk*) l| rỗng.



iii) Với mọi *u0 C0*, thủ tục mở rộng nêu trên x}y dựng được mọi phần tử của *X\**.

*Ví dụ*:*Bài toán người chào hàng*

B|i to{n người ch|o h|ng (TPS) được ph{t biểu như sau. Với *n* th|nh phố đã cho, người ch|o h|ng cần tìm một chu trình có đường đi ngắn nhất qua mỗi th|nh phố đúng một lần.

Để đơn giản, ta xét b|i to{n trên đồ thị vô hướng G= (V,E), trong đó tập đỉnh V (*n* đỉnh) ký kiệu tập c{c th|nh phố v| *E* l| tập c{c cạnh (*i,j*) biểu thi đường đi nối c{c đỉnh *i ,j V* v| có độ d|i *di,j* tương ứng (*di,j* l| khoảng c{ch từ th|nh phố i tới th|nh phố j ). Trong trường hợp n|y *S* l| c{c chu trình trên đồ thị đầy, *f* l| độ d|i đường đi, l| r|ng buộc c{c chu trình qua mọi đỉnh (v| mỗi đỉnh một lần) còn *C0* l| tập c{c

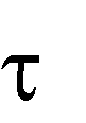
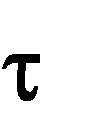
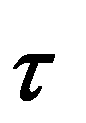
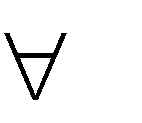
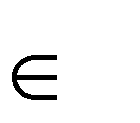
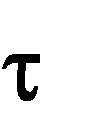
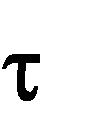
đỉnh của đồ thị.

C{c b|i to{n tối ưu tổ hợp như trên thường thuộc loại NP-khó, phương ph{p ACO đề xuất x}y dựng đồ thị cấu trúc với tập đỉnh *V* m| mỗi đỉnh của nó tương ứng với mỗi th|nh phần của *C* v| có thể dùng c{c thuật to{n theo lược đồ dưới đ}y để giải.

#### 9.3.2. Thuật toán tổng quát

Nói chung, đối với c{c b|i to{n thuộc loại NP-khó thường ta có c{c phương ph{p heuristic để tìm lời giải đủ tốt cho b|i to{n. C{c thuật to{n ACO kết hợp thông tin heuristic n|y với phương ph{p học tăng cường nhờ mô phỏng h|nh vi của đ|n kiến để tìm lời giải tốt hơn.

*Đồ thị cấu trúc*

 Giả sử với mỗi cạnh nối c{c đỉnh *i,j* *C* có trọng số heuristic *hi,j* để định hướng chọn th|nh phần mở rộng l| *j* khi th|nh phần cuối của *xk* là *i* theo thủ tục nêu trên ( hi,j >0 (*i,j*). Đ|n kiến m con sẽ x}y dựng lời giải trên đồ thị có trọng số *G*=(*V,E,H,)* , trong đó V l| tập đỉnh tương ứng với tập th|nh phần *C* đã nêu ở trên (về sau ta vẫn dùng ký hiệu *C* thay cho *V* để chỉ tập đỉnh), *E* l| tập c{c cạnh, *H* l| vectơ c{c trọng số heuristic của cạnh tương ứng (trong b|i to{n TPS nó l| vectơ m| th|nh phần l| nghịch đảo độ d|i của cạnh tương ứng) còn vectơ vết mùi tích luỹ được ban đầu được khởi tạo bằng 0cho mọi th|nh phần đều bằng 0>0. Để dễ trình b|y chúng tôi xét vết mùi để ở c{c cạnh tuy rằng nó có thể để ở c{c đỉnh của đồ thị (xem\*11+). Đồ thị G sẽ gọi l| đồ thị cấu trúc của b|i to{n.

Với điều kiện kết thúc đã chọn (có thể dùng số lần lặp *Nc* định trước) thuật to{n được mô tả hình thức như trong hình 9.4.

*Xây dựng lời giải trong mỗi bước*

Sau khi khởi tạo c{c tham số v| lượng mùi ban đầu, c{c con kiến thực hiện lặp thủ tục x}y dựng lời giải. Trong mỗi lần lặp *t*, mỗi con kiến h chọn ngẫu nhiên một đỉnh xuất ph{t trong *C0* v| kết hợp thông tin heuristic với thông tin mùi để x}y dựng lời giải ngẫu nhiên theo thủ tục mở rộng tuần tự nêu trên với x{c suất chọn đỉnh tiếp theo như sau.

**Procedure** of ACO algorithms.

**Begin**

**Initialize** //khởi tạo m con kiến

**Repeat**

Construct solutions// mỗi con kiến x}y dựng lời giải, Update trail//cập nhật mùi **until** End condition//điều kiện kết thúc **End**

**Hình 9.4.** Đặc tả thuật toán ACO

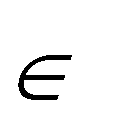
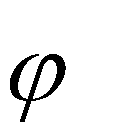
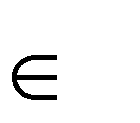
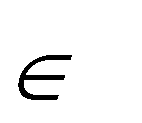
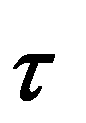
C{c thủ tục x}y dựng lời giải v| cập nhật mùi thực hiện như sau.

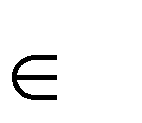
***Quy tắc chuyển trạng thái*.** Giả sử kiến s đã x}y dựng *xk*=<*u0...,uk*>, nó chọn đỉnh y thuộc J(*xk*) để *xk+1*=<*u0,...,u,y*> với x{c suất *P*(*y/* ***,xk*):**

*P*(*y/* ***,xk*) =** (9.11)

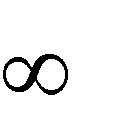
trong đó l| tham số chọn trước.

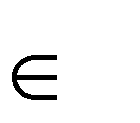
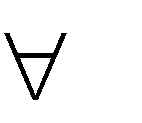
Qu{ trình n|y tiếp tục cho tới khi mỗi con kiến r đều tìm được một lời giải chấp nhận được *x(r) X\** và do dó *s(r) = (x(r)) S*. Khi đó ta sẽ nói c{c cạnh (*ui,ui+1*) thuộc *X(r).* Để tiện trình b|y, về sau ta sẽ xem *x(r)* và *s(r)* như nhau v| không ph}n biệt *X\** với *S*. Ký hiệu l| lời giải tốt nhất c{c con kiến tìm được cho tới lần lặp n|y v| l| lời giải tốt nhất trong bước lặp, nếu không tốt hơn ta sẽ có :*=*  . Ta sẽ quan t}m tới c{c lời giải gần đúng này.



Do giả thiết (3-iii) của b|i to{n v| để tiện trình b|y, về sau ta sẽ không ph}n biệt mỗi x X\* với trạng th{i s S tương ứng.

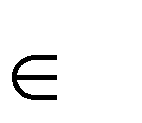
#### *Cập nhật mùi*

Quy tắc cập nhật mùi có vai trò quan trọng, quyết định chất lượng thuật to{n. Có nhiều quy tắc cập nhật mùi, sau đ}y l| một số quy tắc được sử dụng phổ biến nhất hiện nay. Giả sử g l| một h|m gi{ trị thực x{c định trên S sao cho 0< g(s) < s S v| g(s)>g(s’) nếu f(s) <f(s’) (trong b|i to{n TSP g(s) l| nghịch đảo độ d|i đường



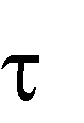
đi tương ứng), khi đó ở mỗi cuối mỗi bước lặp cường độ vết mùi sẽ thay đổi theo một trong c{c quy tắc sau đ}y.

**Quy tắc ACS:** Quy tắc n|y bao gồm cả cập nhật địa phương v| to|n cục.

 *Cập nhật mùi địa phương.* Nếu con kiến h thăm cạnh (*i,j*) ,tức l| (*i,j*) s(h) thì cạnh n|y sẽ thay đổi mùi theo công thức:

(9.12a)

trong đó ở đ}y 1 >0, l| tham số (có thể lấy0).



1

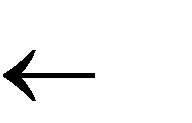
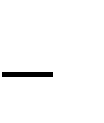
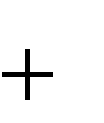
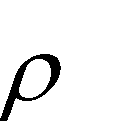
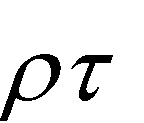
,

,

)

1

(



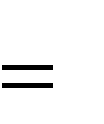
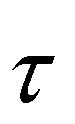
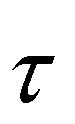
*j*

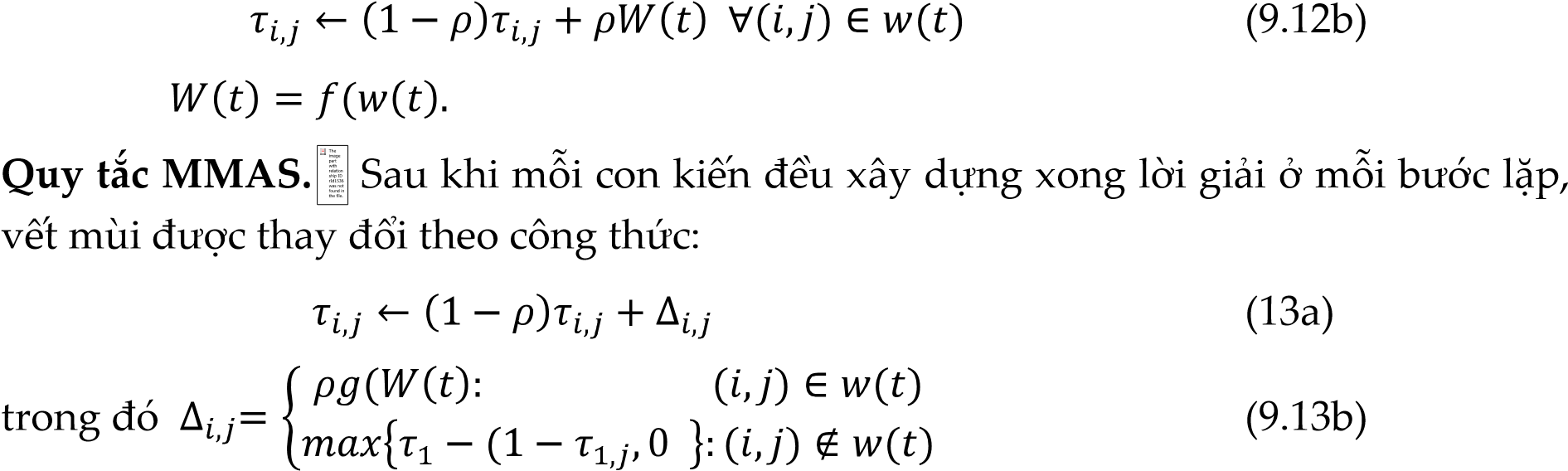
*i*

*j*

*i*

1



*Cập nhật mùi toàn cục.* Cập nhật mùi to|n cục chỉ cho c{c cạnh thuộc *w(t):* trong đó

**Quy tắc SMMAS. Vết mùi đƣợc thay đổi theo công thức:**

(9.14)

trong đó phản {nh chiến lược kh{m ph{ hay tìm kiếm tăng cường v| có ảnh hưởng quan trọng tới chất lượng của thuật toán.

l| hai hằng số chọn trước. Tỷ lệ

Các quy tắc cập nhật mùi trên là quy tắc G-best, nếu trong các công thức (9.12a) đến (9.14) thay thì ta nói l| quy tắc i-best, thường thì cập nhật i-best tốt hơn G-best.

bởi

Người ta đã chứng minh lời giải gần đúng hội tụ theo x{c suất tới lời giải tối ưu của b|i to{n. Trong c{c quy tắc cập nhật mùi ở trên, quy tắc SMMAS đơn giản, dễ sử dụng v| hiệu quả hơn.

**Nhận xét.** Phương ph{p ACO không đặt vấn đề tìm chiến lược tối ưu như b|i to{n học tăng cường m| chỉ x}y dưng qu{ trình quyết định Markov tối ưu với trạng th{i ban đầu thuộc tập l| lời giải cho b|i to{n. Việc chọn đỉnh kế tiếp trong thủ tục tuần tự được xem l| t{c động. Vết mùi cập nhật trên mỗi cạnh tương ứng với gi{ trị tích lũy thưởng, việc thưởng, c{c đại lượng l| lượng thưởng *tức thời*. Khác với học Q ở chỗ lượng chiết khấu {p dụng cho c{c gi{ trị thưởng tức thời trước đó, c|ng xa hiện tại thì số lần chiết khấu c|ng tăng, việc thưởng cho mỗi lần lặp trong c{c quy tắc cập nhật mùi thực hiện cho mọi cạnh (t{c động) sau khi đã được bình luận (đ{nh gi{). Như vậy, về bản chất ACO l| một vận dụng của phương ph{p học Q cho b|i to{n chỉ tìm một qu{ trình quyết định Markov tối ưu. C{c kết quả thực nghiệm cho thấy phương ph{p n|y hiệu quả hơn giải thuật di truyền.

#### KẾT LUẬN

Học tăng cường l| b|i to{n học tìm chính s{ch điều khiển tối ưu của một t{c tử tự trị. Trong đó giả thiết rằng thông tin học biểu diễn dưới dạng gi{ trị thưởng thực được cho đối với mỗi phép chuyển trạng th{i-t{c động. Mục đích của t{c tử l| học một chính s{ch t{c động l|m cực đại gi{ trị thưởng tích lũy m| nó nhận được từ bất cứ trạng th{i khởi đầu n|o.

C{c thuật to{n học tăng cường được trình b|y trong chương n|y l|m việc với qu{ trình quyết định Markov (MPS), trong đó kết quả của mỗi t{c động ở bất cứ trạng th{i n|o chỉ phụ thuộc v|o t{c động v| trạng th{i đang xét m| không phụ thuộc v|o c{c t{c động v| trạng th{i trước đó. C{c MPS thường gặp trong c{c b|i to{n ở phạm vi rộng, bao gồm những b|i to{n điều khiển robot, tự động hóa nh| m{y, lập lịch.

Học Q l| một dạng của học tăng cường, trong đó t{c tử học một h|m đ{nh gi{ c{c trạng th{i v| t{c động. Đặc biệt, h|m đ{nh gi{ *Q*(*s,a*) được định nghĩa như l| kỳ vọng tích lũy thưởng có chiết khấu m| t{c tử có thể nhận được khi thực hiện t{c động *a* ở trạng th{i *s*. c{c thuật to{n học Q có ưu đểm l| nó có thể sử dụng ngay cả khi bộ học không có tri thức trước về việc c{c t{c động ảnh hưởng tới môi trường ra sao.

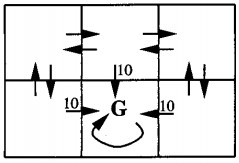
C{c ước lượng *Q s a*ˆ( , ) cho h|m Q được biểu diễn dưới dạng bảng tra cứu với c{c th|nh phần riêng biệt cho mỗi cặp (*s,a*). Với một số giả thiết trong học Q, ước lượng *Q s a*ˆ( , ) hội tụ đến h|m đúng *Q*(*s,a*) cả trong trường hợp c{c MDP đơn định v| ngẫu nhiên. Trong ứng dụng, học Q có thể đòi hỏi nhiều nghìn lần lặp huấn luyện cho một b|i to{n cỡ tầm thường.

Trong học Q, cơ chế chuyển t{c động-trạng th{i đóng vai trò bản chất nội tại của thuật to{n, gi{ trị thưởng tức thời v| tích lũy thưởng có chiết khấu đóng vai trò như lượng hóa sự bình luận hiệu quả của c{c t{c động v| c{c qu{ trình MDP, với mỗi b|i to{n, người thiết kế bộ học có thể thay đổi thích hợp c{c gi{ trị n|y.

Phương ph{p ACO l| một tiếp cận vận dụng phương ph{p học *Q* cho trường hợp chỉ cần tìm một xích Markov có trạng th{i khởi đầu thuộc tập . Thực nghiệm cho thấy hiệu quả nổi trội của nó so với c{c phương ph{p kh{c như mô phỏng luyện kim hay thuật to{n di truyền.

#### BÀI TẬP

1. Giải thích c{c số liệu trong hình 9.2b v| tìm một chính s{ch tối ưu thứ hai cho b|i to{n được mô tả trong hình 9.2
2. Đưa ra c{c chuỗi t{c động-trạng th{i trong thuật to{n học Q để giải thích l|m thế n|o có được c{c số liệu trong bản 9.3a?
3. Xét thế giới lưới đơn định với trạng th{i hút G cho trong hình bên dưới. Trong đó c{c phép chuyển trạng th{i có nhãn nhận gi{ trị thưởng tức thời bằng 10, c{c phép chuyển không có nhãn nhận gi{ trị thưởng bằng không.



* 1. Tính gi{ trị cho mỗi trạng thái trong thế giới lưới này. Với tính cho mỗi t{c động v| cuối cùng chỉ ra một chiến lược điều khiển tối ưu.
  2. Gợi ý một thay đổi h|m thưởng *r*(*s,a*) sao cho nó l|m thay đổi nhưng không thay đổi chính s{ch tối ưu.
  3. Gợi ý một thay đổi cho *r*(*s,a*) sao cho nó l|m thay đổi nhưng không thay đổi .

d)Xét thuật to{n học Q với bảng được khởi tạo bằng không, giả sử t{c tử khởi đầu ở trạng th{i bên tr{i-dưới v| đi theo chiều kim đồng hồ tới trạng th{i hút. Thực hiện hiệu chỉnh bảng cho mỗi t{c động ở giai đoạn n|y. Hãy mô tả tiếp thay đổi của bảng nếu t{c tử thực hiện lại lần hai cùng quỹ đạo.

1. Lưu ý rằng trong nhiều MDP, có thể tìm được hai chính s{ch điều khiển và sao cho khi t{c tử khởi đầu từ trạng th{i thì còn khi khởi đầu từ trạng th{i thì . Giải thích tại sao luôn tồn tại một chính s{ch l| cực đại đối với mọi trạng th{i ban đầu *s* (tức l| chính s{ch tối ưu ).
2. Chỉ rõ thuật to{n ACO giải b|i to{n TSP (tìm chu trình Hamilton có độ d|i ngắn nhất) với quy tắc cập nhật mùi SMMAS v| chỉ ra c{c yếu tố tương đồng với phương ph{p học Q trong cơ chế vết mùi v| gi{ trị thưởng.

**Chƣơng 10**

## KẾT HỢP CÁC BỘ HỌC

Đã có nhiều thuật to{n học kh{c nhau được giới thiệu trong c{c chương trước. Tuy nhiên, khi {p dụng thì không có thuật to{n n|o tạo nên được một bộ học thực sự chính x{c hơn hẳn c{c phương ph{p kh{c. Một c{ch tiếp cận có hiệu quả để tăng độ chính x{c l| kết hợp c{c bộ học nhận dạng với nhau để được một bộ tốt hơn hay còn gọi l| học tập thể.

Chương n|y giới thiệu c{c kỹ thuật thông dụng trong kết hợp c{c bộ học: bỏ phiếu, tạo c{c bộ học cơ sở bằng c{ch nhặt dữ liệu theo gói (bagging) v| nhặt định hướng (boosting), rừng ngẫu nhiên, kiến trúc bậc thang.

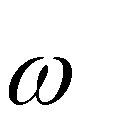
### 10.1. LÝ DO NÊN HỌC TẬP THỂ

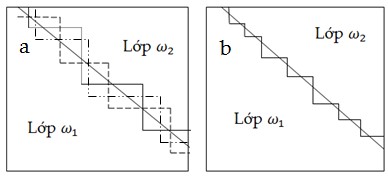
Với mỗi b|i to{n ph}n lớp hoặc hồi quy cụ thể, người ta thường có nhiều thuật to{n học để khi x}y dựng bộ học. Cùng một thuật to{n, có thể chọn c{c tham số kh{c nhau hoặc sử dụng tập dữ liệu huấn luyện kh{c nhau nên cho c{c bộ nhận khác nhau.

*Ví dụ*. Cùng dùng mạng MLP nhưng ta chọn số tầng ẩn/nơron tầng ẩn hay số đặc trưng kh{c nhau thì kết quả huấn luyện cho c{c bộ nhận dạng có chất lượng kh{c nhau. Cùng l| phương ph{p học k-NN nhưng với mỗi *k* cho ta một bộ nhận dạng v| rất khó khẳng định được tham số *k* n|o l| tối ưu.

Những thuật to{n cho cùng lớp b|i to{n thường tu}n theo luật “*không có bữa trưa miễn phí (no free lunch theory)*”, tức l| không có thuật to{n tốt hơn hẳn c{c thuật to{n m| mỗi thuật to{n có ưu /nhược điểm riêng, khi thực hiện nhận dạng thì mỗi bộ huấn luyện theo thuật to{n tương ứng có những lớp mẫu nhận dạng tốt v| tồi kh{c nhau. Kết hợp hợp lý c{c bộ nhận dạng có thể cho ta bộ nhận dạng mới có nhiều ưu điểm hơn v| ta gọi l| học tập thể (ensemble learning).

Một bộ ph}n lớp được kết hợp từ 3 bộ ph}n lớp cơ sở kh{c nhau theo c{ch bỏ phiếu được minh họa trong hình 10.1. Hãy tưởng tượng hai lớp 1 và 2 có biên là đường thẳng nhưng hai bộ ph}n lớp cơ sở tạo nên c{c biên tương ứng l| đường liền v| c{c đường đứt như trong hình 10.1a, còn hình 10.1b chỉ ra rằng bộ ph}n lớp kết hợp theo hình thức bỏ phiếu cho ta biên quyết định gần với biên thực hơn.





**Hình 10.1.** a) Biên quyết định của mỗi bộ phân lớp. b) Biên kết hợp ba bộ

Như vậy, mỗi c{ch học cho ta một bộ nhận dạng cơ sở, nhờ kết hợp c{c bộ nhận dạng th|nh phần có được m| ta có một bộ nhận dạng tốt hơn. C{c bộ nhận dạng cơ sở n|y thường được x}y dựng theo c{c tiếp cận sau đ}y:

1. *Dùng các thuật toán huấn luyện khác nhau*. C{c thuật to{n n|y sử dụng c{c giả thiết kh{c nhau về dữ liệu, c{c bộ học có thể phụ thuộc tham số hoặc không. Khi kết hợp c{c bộ học, ta được giải phóng khỏi c{c giả thiết {p đặt n|y.
2. *Mỗi bộ học dùng cách chọn đặc trưng khác nhau*. Chẳng hạn chúng ta dùng thuật to{n SVM để ph}n biệt chữ viết tay nhưng c{ch chọn đặc trưng có thể l| nội dung ảnh hay qua phép biến đổi n|o đó.
3. *Có thể sử dụng cùng một thuật toán nhưng có tham số khác nhau*. Chẳng hạng đều sử dụng thuật to{n k-l{ng giềng gần nhất nhưng với k kh{c nhau.
4. *Cùng một thuật toán nhưng sử dụng các tập dữ liệu huấn luyện khác nhau*.

Thông thường thì c{c bộ nhận dạng được x}y dựng theo hai c{ch c{ch tiếp cận đầu có thời gian chạy kh{c nhau v| bộ nhận dạng chính x{c hơn thường đòi hỏi thời gian xử lý nhiều hơn.

Khi có c{c bộ nhận dạng cơ sở, bộ nhận dạng tập thể được kết hợp theo c{c kiểu tôpô đa dạng để cho ta những bộ mới tốt hơn c{c bộ th|nh phần. Trong đó phương thưc kết hợp đơn giản v| dễ dùng nhất l| phương ph{p bỏ phiếu.

### 10.2. PHƢƠNG PHÁP BỎ PHIẾU

Một c{ch đơn giản để kết hợp c{c bộ học cơ sở l| dùng phương ph{p bỏ phiếu nhờ kiến trúc song song, đầu ra được quyết định nhờ kết quả tổng hợp có trọng số của c{c bộ nhận dạng th|nh phần. Đối với đối tượng ***x*** cần g{n nhãn, nếu mỗi bộ học cơ sở *C i*cho quyết định **d** *i* với trọng số *ý kiến* *wi* tương ứng thì đầu ra của bộ kết hợp đối với mẫu n|yđược tính theo công thức:

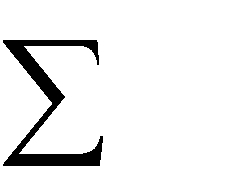
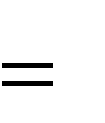
cho b|i to{n hồi quy, (10.1a)

v| theo đa số có trọng số của tập cho b|i to{n ph}n lớp, (10.1b) trong đó c{c trọng số thỏa mãn:

*L*

1 (10.1c)

1



*i*

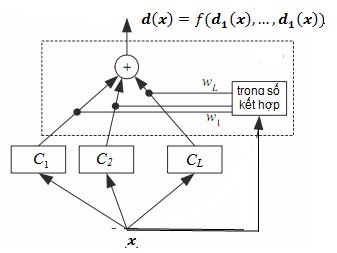
*i*

*w*

C{c trọng số có thể chọn bằng nhau. Tổng qu{t hơn, ta có thể quyết định bằng một h|m tổng hợp phi tuyến *f* n|o đó:

(10.2)

Sơ đồ quyết định tổng qu{t của quyết định theo hình thức bỏ phiếu được mô tả trong hình 10.2.



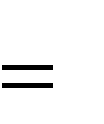
**Hình 10.2** . Sơ đồ kết hợp các bộ nhận dạng nhờ bỏ phiếu

Việc huấn luyện c{c bộ th|nh phần của bộ học tập thể n|y có thể sử dụng một trong c{c phương thức sau:

*L* thuật to{n huấn luyện kh{c nhau.



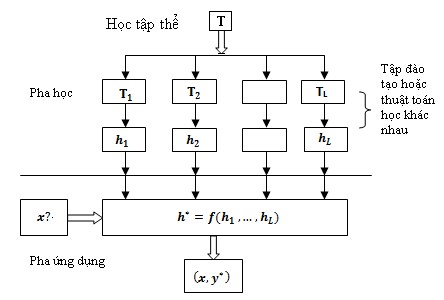
Một thuật to{n nhưng *L* tập dữ liệu đ|o tạo hay tham số kh{c nhau. Một thuật to{n nhưng dùng tập dữ liệu với tập đặc trưng kh{c nhau Kết hợp c{c phương thức trên.

 Việc học tập thể T bao gồm c{c qu{ trình huấn luyện **T***i* cho bộ học để cho giả thuyết *hi* tương ứng v| chúng được kết hợp th|nh giả thuyết *h*\* . Khi ứng dụng nhận dạng mẫu ***x***, giả thuyết *h*\* sẽ cho ta nhãn **y**\**h*\* (*x*) như minh họa trong hình 10.3.

**Hình**

**10.3.**

Sơ đồ học tập thể của các bộ học



### 10.3. KỸ THUẬT TẠO VÀ KẾT HỢP BỘ NHẬN DẠNG CƠ SỞ

Mục n|y giới thiêu ba kỹ thuật thông dụng để tạo v| kết hợp c{c bộ học cơ sở nhờ dùng một thuật to{n học với c{ch tạo v| dùng dữ liệu kh{c nhau: nhặt theo gói (bagging ), nhặt định hướng bao gồm boosting v| adaboost, rừng ngẫu nhiên.

#### 10.3.1. Nhặt theo gói

*Nhặt theo gói* (*Bagging*) l| phương ph{p học tập thể đơn giản v| thông dụng nhất, mặt kh{c, nó giúp giảm phương sai khi dùng cho b|i to{n hồi quy. Phương ph{p n|y dùng cùng một thuật to{n để x}y dựng c{c bộ nhận dạng cơ sở bằng c{ch lấy ngẫu nhiên từ tập dữ liệu đ|o tạo một tập dữ liệu cho một bộ học, sau đó kết nối song song c{c bộ nhận dạng có được v| quyết định với trọng số đều. Cụ thể như sau.

 Giả sử ta có tập dữ liệu đ|o tạo *D*  gồm N mẫu đã g{n nhãn . Từ một thuật to{n cơ sở ( chẳng hạn ID3, K-NN hay mạng nơ ron<) ta huấn luyện *L* bộ học trên *L* tập con dữ liệu được lấy ngẫu nhiên của *D*. Để có *L* tập dữ liệu n|y, ta chọn trước một số *M* < *N* v| với mỗi *i* 1,2,...,*L*, lấy ngẫu nhiên ph}n bố đều từ *D* tập *Di* gồm *M* đối tượng l|m dữ liệu đ|o tạo. Sau khi huấn luyện bằng thuật to{n đã chọn trên c{c tập dữ liệu *Di iL* 1 n|y ta có c{c bộ nhận dạng *Ci iL* 1 .. C{c bộ nhận dạng n|y được kết hợp theo hình thức bỏ phiếu với trọng số bằng nhau, kiến trúc của bộ học n|y được mô tả trong hình 10.4.



*N*

*k*

*k*

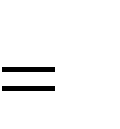
*k*

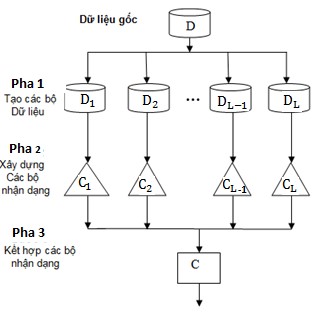
*y*

*x*

1

,





**Hình 10.4.** Kiến trúc của hệ học theo phương pháp Bagging

Thuật to{n x}y dựng bộ nhận dạng được mô tả trong bảng 10.1.

**Bảng 10.1**. Thuật toán Bagging

|  |
| --- |
| Bước 1. Lặp với i=1,, 2, <, K:  1.1. *Di*  M mẫu đ|o tạo lấy ngẫu nhiên từ D;  1.2. *hi*  Học từ *Di* theo thuật to{n được chọn;  Bước 2. Kết hợp c{c *hi* theo hình thức bỏ phiếu với trọng số đều. |

Kỹ thuật n|y có thể {p dụng cho cả b|i to{n ph}n lớp v| hồi quy. Dietterich v| Bakiri (1991) thử nghiệm ph}n lớp cho c{c tập dữ liệu thực v| dữ liệu chuẩn trên UCI với 100 bộ ph}n lớp cơ sở đã giảm 10% lỗi cho dữ liệu thực v| 6% đến 42% lỗi cho dữ liệu trên UCI.

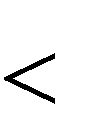
#### 10.3.2. Nhặt định hƣớng

Mục đích của phương ph{p nhặt định hướng l| để cải tiến độ chính x{c của thuật to{n ph}n lớp đã cho. Trong phương ph{p n|y người ta x}y dựng c{c *bộ học yếu*, trong đó tập dữ liệu huấn luyện dùng cho bộ học sau được lấy định hướng dựa trên lỗi của c{c bộ học trước.

Dưới đ}y giới thiệu hai kỹ thuật thông dụng của c{ch tiếp cận n|y l| *boosting* và *adaboost*

#### Boosting

Đối với b|i to{n hai lớp, phương ph{p n|y tạo ra 3 bộ ph}n lớp yếu theo qúa trình sau đ}y.

Trước tiên người ta lấy ngẫu nhiên tập *D*1 gồm *n*1 (*N*) mẫu trong tập dữ liệu huấn luyện *D*. Sau đó huấn luyện bộ ph}n lớp *C*1 từ tập dữ liệu *D*1 . *C*1 l| bộ học *yếu*, tức l| tỷ lệ lỗi cao (nếu độ chính x{c cao thì mức độ cải tiến sẽ thấp, tuy nhiên nó có thể có tỷ lệ lỗi thấp trong tập dữ liệu huấn luyện).

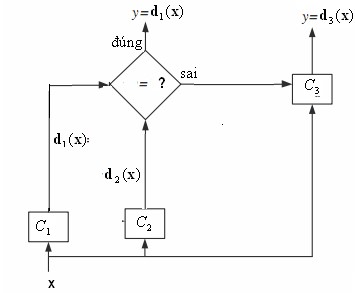
Tập dữ liệu huấn luyện *D*2 được x}y dựng dựa trên thông tin định hướng của bộ ph}n lớp *C*1, trong đó một nửa được *C*1 ph}n lớp đúng còn một nửa ph}n lớp sai. Cụ thể như sau: lấy ngẫu nhiên, *r* (0,1) nếu *r* 1/2thì lấy từng mẫu một trong D chưa thuộc *D*2 cho *C*1 đo{n nhận cho đến lúc sai thì bổ sung v|o *D*2 , ngược lại nếu*r* 1/2 thì lấy từng mẫu một còn trong D chưa thuộc *D* 2 cho *C*1 đo{n nhận cho đến lúc đúng thì bổ sung v|o *D* 2 . Thủ tục lặp theo c{ch n|y cho đến khi *D* 2 có *n*2 dữ liệu gồm một nửa *C*1 nhận dạng đúng v| một nửa nhận dạng sai. Huấn luyện bằng tập dữ liệu *D*2 ta được bộ ph}n lớp *C* 2 .

Tiếp theo ta x}y dựng tập dữ liệu huấn luyện *D*3 gồm c{c dữ liệu m| *C*1 và *C* 2 không thống nhất. Ta lần lượt lấy mẫu ngẫu nhiên từ *D*, nếu *C*1 và *C* 2 ph}n loại không thống nhất thì bổ sung nó v|o *D*3 , ngược lại thì mẫu n|y bị bỏ qua. Theo c{ch n|y ta x}y dựng được *D*3 v| dùng để huấn luyện bộ ph}n lớp *C*3 .

B}y giờ ba bộ n|y được kết hợp như sau. Nếu mẫu ***x*** m| kết quả đo{n nhận của

*C*1 và *C*2 thống nhất thì nhận nhãn chung n|y, ngược lại, nếu *C*1 và *C*2 không thống nhất thì dùng nhãn đo{n nhận của *C*3 .

Ký hiệu l| kết quả đo{n nhận của bộ học *Ci* đỗi với mẫu ***x*** thì nhãn ***y*** của nó được đo{n nhận theo sơ đồ ở hình 10.5.



**Hình 10.5**. Sơ đồ quyết định sử dụng boosting

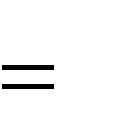
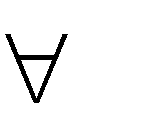
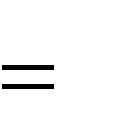
Freund v| Schapire (1996) đề xuất phương ph{p nhặt định hướng thích nghi (Adaboost) cho phép thiết kế tiếp tục c{c bộ học cho đến khi chất lượng được cải tiến thực sự.

#### Adaboost

Trong Adaboost, mỗi lần x}y dựng tập huấn luyện *Di* thì c{c mẫu **x** *j* trong *D*

được lấy với x{c suất *pij* . Ban đầu c{c x{c suất n|y như nhau: *p*1*j* 1 1,..,*N* để

*j*



*N*

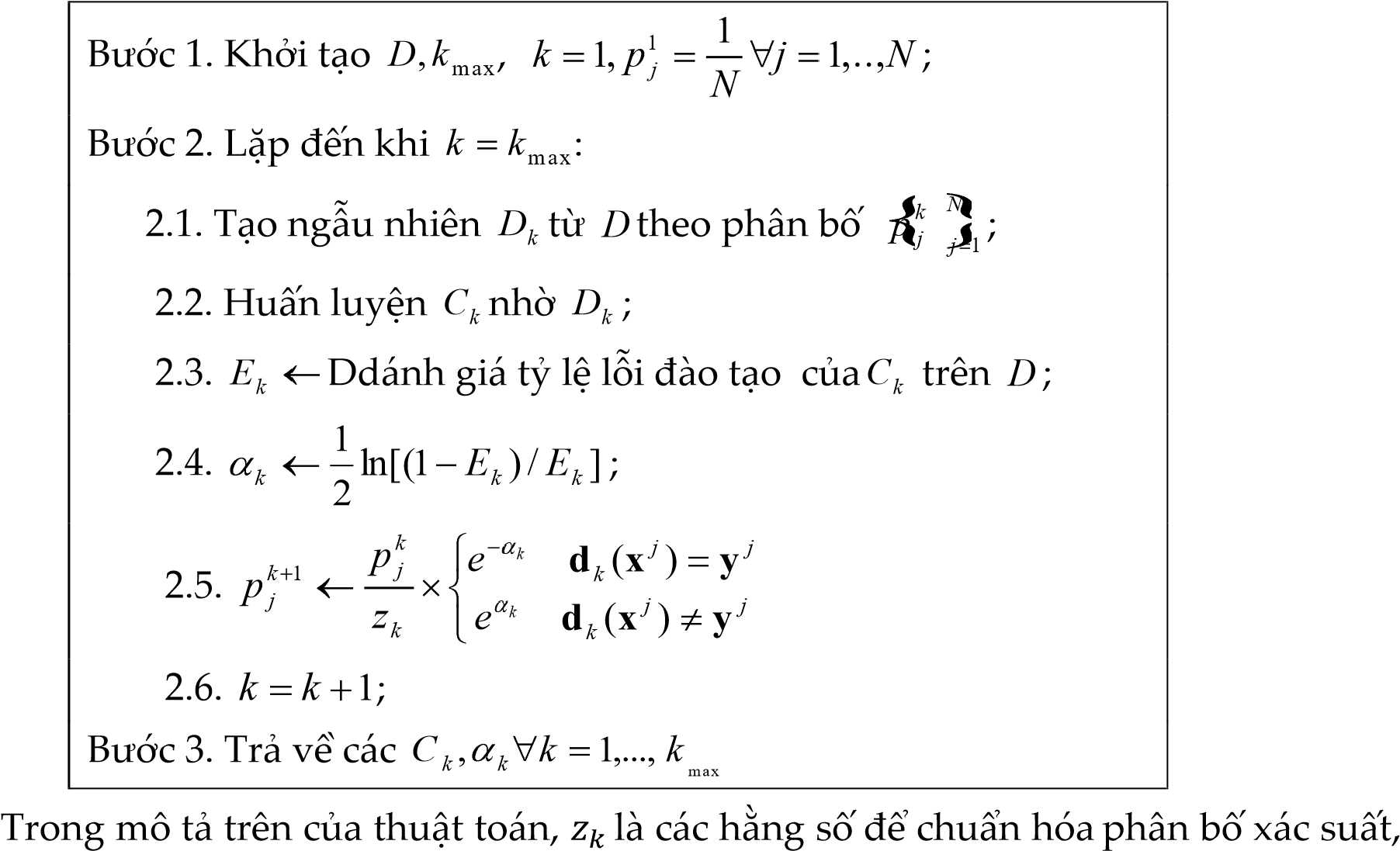
x}y dựng *D*1 v| dùng nó để huấn luyện bộ ph}n lớp*C*1 . Sau mỗi bước lặp c{c x{c suất n|y tăng dần với c{c mẫu ph}n lớp sai v| giảm dần với mẫu ph}n lớp đúng. Giả sử ở bước lặp thứ *k*, ta đã chọn tập đ|o tạo *Dk* với ph}n bố x{c suất *p kj Nj* 1 trong



*D* v| huấn luyện được bộ ph}n lớp *Ck* , ta tăng x{c suất chọn *p kj* 1 *Nj* 1 đối với c{c mẫu trong *D* mà *Ck* ph}n lớp sai v| giảm x{c suất chọn đối với c{c mẫu ph}n lớp đúng.

Đến nay có nhiều biến thể của Adaboost được sử dụng, một thể hiện của nó được mô tả trong bảng 10.2.

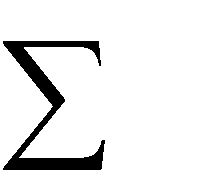
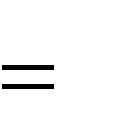
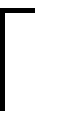
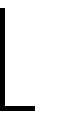
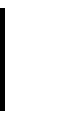
**Bảng 10.2**. Một biến thể của Adaboost



**y** *j*l| nhãn đúng của **x** *j* , **d** *i* (**x**)l| kết quả đo{n nhận của bộ học *Ci* đối với mẫu ***x*.** Kết

qủa ph}n lớp cuối cùng của mẫu ***x*** sẽ l|:

*g*(**x**) (10.3)



max

1

)

(

*k*

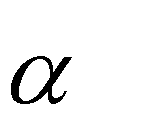
*k*

*k*

*k*

**x**

**d**



trong đó tổng sẽ lấy theo từng nhãn lớp, còn ngoặc vuông được hiểu l| lấy nhãn lớp có trọng số tổng hợp lớn nhất.

##### **10.3.3. Rừng ngẫu nhiên**

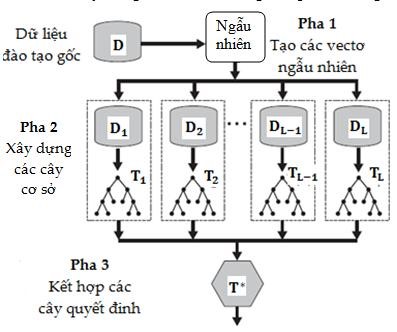
Rừng ngẫu nhiên l| phương ph{p học tập thể rất thích hợp cho xử lý dữ liệu có số chiều cao, vì vậy đang thu hút nhiều người quan t}m nghiên cứu, {p dụng. Định nghĩa sau đ}y giải thích tên gọi của phương ph{p n|y.

***Định nghĩa 10.1***. Rừng ngẫu nhiên l| một bộ nhận dạng bao gồm một tập bộ ph}n lớp cơ sở dạng c}y quyết định được kết hợp theo phương thức bỏ phiếu. C{c bộ cơ sở được x}y dựng từ c{c tập con dữ liệu với đặc trưng kh{c nhau được lấy ngẫu nhiên từ tập dữ liệu đ|o tạo.

Định nghĩa cho thấy thủ tục x}y dựng rừng ngẫu nhiên gồm ba pha: tạo dữ liệu (tạo vectơ ngẫu nhiên), x}y dựng c{c c}y cơ sở, kết hợp c{c c}y cơ sở theo phương thức bỏ phiếu.

Điểm mới nhất ở đ}y l| pha tạo dữ liệu. Giả sử l| tập dữ liệu đ|o tạo với số chiều *n* lớn, pha tạo vectơ ngẫu nhiên thực hiện như sau.

Chọn trước c{c số tự nhiên *M* (< *N*) và *m* (< *n*). Để có mỗi tập dữ liệu cho x}y đựng c}y quyết định , ta chọn ngẫu nhiên *m* đặc trưng trong số *n* đặc trưng của D v| lấy ngẫu nhiên *M* đối tượng từ D rồi chiếu nó lên c{c đặc trưng được chọn n|y. Việc chọn đặc trưng v| lấy dữ liệu từ D gọi l| tạo vectơ ngẫu nhiên. Qu{ trình x}y dựng rừng ngẫu nhiên được mô tả trong hình 10.6.



**Hình 10.6.** Sơ đồ xây dựng rừng ngẫu nhiên

*Chọn số đặc trưng m*

Rừng ngẫu nhiên thường {p dụng cho c{c b|i to{n ph}n lớp hoăc hồi quy có số chiều *n* lớn, khi đó số đặc trưng *m* cho mỗi tập dữ liệu c}y được chọn nhỏ hơn *n* nhiều. Breiman gợi ý chọn *m* như sau:

Đối với bài toán phân lớp, , trong đó ký hiệu phần nguyên của *a*.

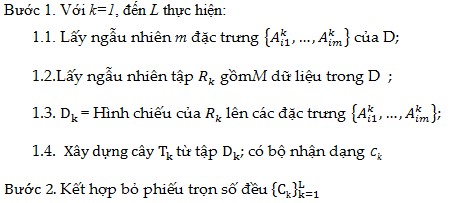


Đối với bài toán hồi quy,

Hiện nay việc chọn số đặc trưng , số đối tượng dữ liệu *M* cho mỗi tập và c{ch chọn ngẫu nhiên để lấy c{c đặc trưng thế n|o l| tốt cho từng lớp b|i to{n vẫn đang l| chủ đề mở v| được nhiều người nghiên cứu. C{c c}y quyết định được kết hợp song song theo phương ph{p bỏ phiếu với đầu ra có trọng số đều cho bởi c{c công thức (10.1a-b). Thuật to{n x}y dựng rừng ngẫu nhiên được mô tả trong bảng

10.3.

**Bảng 10.3.** Thuật toán xây dựng rừng ngẫu nhiên cho phân lớp hoặc hồi quy



Đầu ra của hệ cho đối tượng

***x***

sẽ l|:



Đ

ố

i v

ớ

i bài toán h

ồ

i quy:

(10.4a)



Đ

ố

i v

ớ

i bài toán phân l

ớ

p:

(10.4b)

ngoặc vuông chỉ quyết định theo đa số.

### 10.4. KIẾN TRÚC BẬC THANG

Trong mục 10.3, ta đã xét c{c bộ ph}n lớp cơ sở tạo nên nhờ cùng một thuật to{n huấn luyện nhưng dùng c{c tập dữ liệu kh{c nhau để huấn luyện. Trong thực tế, có thể dùng c{c thuật to{n v| c{ch trích chọn đặc trưng kh{c nhau để x}y dựng c{c bộ nhận dạng cơ sở. Khi đó c{c bộ ph}n lớp n|y thường có hiệu quả v| thời gian chạy kh{c nhau. Thông thường thì bộ ph}n lớp có độ chính x{c cao sẽ tốn nhiều thời gian chạy hơn. Trong trường hợp đó, kiến trúc bậc thang l| phương ph{p hiệu quả để kết hợp c{c bộ ph}n lớp n|y.

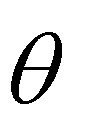
Ta xét *L* bộ ph}n lớp, được sắp theo thứ tự tăng dần về độ chính x{c:

*C*1 , *C* 2 ,..., *C L* . Mỗi bộ ph}n lớp *Ci* có h|m đ{nh gi{ độ tin cậy *wi* v| ngường tin cậy sao cho với mẫu ***x*** mà thì nhãn hầu chắc chắn đúng (độ chính x{c cao). Khi đó c{c bộ ph}n lớp n|y có thể kết hợp theo kiến trúc bậc thang như trong hình 10.7. C{c mẫu được đưa tuần tự cho từng bộ ph}n lớp bắt đầu từ *C*1 , nếu bộ ph}n lớp *Ck* đang xét dự đo{n nhãn mẫu **x** với độ tin cậy lớn hơn ngưỡng thì lấy nhãn **d** *k* (**x**)cho nó, ngược lại thì đưa cho bộ *Ck*1 đo{n nhận, cuối cùng thì d|nh cho bộ *C L*. Bộ ph}n lớp kết hợp theo kiến trúc n|y thường có độ chính x{c cao hơn c{c bộ th|nh phần v| thời gian chạy trung bình để xử lý một mẫu cũng ít hơn.

*i*



*k*





**Hình 10.7**. Kiến trúc bậc thang của L bộ phân lớp

#### KẾT LUẬN

Trong học có gi{m s{t để x}y dựng bộ ph}n lớp hoặc hồi quy, nếu dùng một thuật to{n v| dùng một tập dữ liệu đ|o tạo thì ta chỉ được một bộ nhận dạng yếu, tức l| sai số lớn. Học tập thể l| một phương ph{p để tăng độ chính x{c của bộ nhận dạng. Trong c{ch tiếp cận n|y, người ta x}y dựng c{c bộ nhận dạng cơ sở theo c{c phương thức: 1) dùng c{c thuật to{n huấn luyện kh{c nhau. 2) dùng một thuật to{n nhưng sử dụng c{c tập dữ liệu đ|o tạo hay tham số kh{c nhau, 3) dùng một thuật to{n nhưng dùng tập dữ liệu với tập đặc trưng kh{c nhau. 4) kết hợp c{c phương thức trên .

Bỏ phiếu l| phương ph{p kết hợp c{c bộ nhận dạng đơn giản v| thông dụng nhất. Trong sơ đồ n|y, c{c bộ nhận dạng cơ sở được kết nối song song v| đầu ra tổng hợp thường dùng nhất l| dạng tuyến tính, có thể lấy trọng số đều. Phương ph{p bỏ phiếu không chỉ tăng độ chính x{c m| trong c{c b|i to{n hồi quy, nó còn cho kết quả có phương sai thấp hơn c{c bộ th|nh phần.

Bagging l| phương ph{p đơn giản nhất để x}y dựng bộ học tập thể , trong đó c{c bộ nhận dạng th|nh phận được x}y dựng nhờ dùng cùng một thuật to{n nhưng dùng c{c tập dữ liệu đ|o tạo kh{c nhau được lấy ngẫu nhiên có ho|n lại c{c tập dữ liệu gốc.

C{c kỹ thuật nhặt theo hướng bao gồm boosting v| adaboost kh{c với bagging ở chỗ ph}n bố x{c suất lấy dữ liệu thay đổi theo thứ tự lấy tập dữ liệu đ|o tạo của bộ nhận dạng cơ sở. Phương thức kết hợp c{c bộ cơ sở trong kỹ thuật boosting không theo kiến trúc song song, nhờ đó tiết kiệm thời gian chạy hơn.

Rừng ngẫu nhiên l| phương ph{p thích hợp cho c{c b|i to{n ph}n lớp hoặc hồi quy với dữ liệu có số chiều cao. Trong đó mỗi bộ nhận dạng cơ sở l| một c}y quyết định, được x}y dựng từ tập dữ liệu có số chiều nhỏ nhờ chiếu một tập dữ liệu lấy ngẫu nhiên từ tập dữ liệu đ|o tạo gốc lên tập đặc trưng đươc chọn ngẫu nhiên.

Khi c{c bộ ph}n lớp được x}y dựng nhờ dùng c{c thuật to{n kh{c nhau, có độ chính x{c v| thời gian chạy kh{c nhau thì có thể dùng kiến trúc bậc thang để tăng độ chính x{c v| giảm thời gian chạy trung bình so với bộ nhận dạng th|nh phần có độ chính x{c cao nhất nhưng thời gian chạy cũng l}u nhất.

#### BÀI TẬP

1. Giả sử mỗi bộ ph}n lớp cơ sở đều có x{c suất đúng p >1/2, hãy ước lượng x{c suất đúng khi bỏ phiếu theo đa số của *L* bộ ph}n lớp.
2. Hãy đề xuất một lược đồ boosting cho b|i to{n nhiều lớp.
3. Tìm một lược đồ điều chỉnh x{c suất chọn tập đ|o tạo theo adaboost kh{c với lược đồ ở mục 10.3.
4. Hãy đề xuất một c{ch chọn đặc trưng để tạo c{c tập dữ liệu cho c}y quyết định của phương ph{p rừng ngẫu nhiên. Giải thích vì sao c{ch chọn n|y tốt hơn chọn ph}n bố đều.
5. Tại sao trong kiến trúc bậc thang ta xếp c{c bộ ph}n lớp tăng dần về độ chính xác?

### TÀI LIỆU THAM KHẢO

1. E. Alpaydın, *Introduction to Machine Learning*, Massachusetts Institute of Technology, Second Edition, 2010
2. T. Mitchell, *Machine learning*, McGraw-Hill, 1997
3. N.J Nielsson, *Introduction to Machine Learning, Robotic Laboratory*, Stanford University, 1997
4. K.P. Murphy, *Machine Learning*, MIT Press, 2012
5. T. Hastie, R.Tibshirani and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd edition), Springer, 2009
6. C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006
7. G. James, D. Witten, T. Hastie and R.t Tibshirani, *An Introduction to Statistical Learning* *with Applications in R*, Springer 2013
8. R.O. Duda et al, Pattern Clasification, John Wiley& Sons, Inc, 2001
9. Hoàng Xuân Huấn, *Giáo trình nhận dạng mẫu*, NXB ĐHQG HN, 2012
10. J. P**.** Masques de Sá, *Pattern Recognition - Concepts, Methods and applications*. Springer, 2001
11. S. Theodoridis, *Pattern Recognition*, Elssevier Academic Press (4th edition), 2009
12. A.R. Webb and K.D. Copsey, *Statistical Pattern Recognition*, John Wiley& Sons, Inc. (3rd edition), 2011
13. H.B. Demuth and M. Beale, *Neural Network Design*, PWS Publishing company, 1996
14. S. Haykin, *Neural Network and Machine Learning* ,Pearson International Edition (3rd edition), 2009
15. C. G. *Looney, Pattern Recognition using Neural Networks*: Theory and Algorithms forEngineers and Scientists, Oxford University Press, New York,1997.
16. M. Dorigo and T.Stutzle: *Ant Colony Optimization*. MIT Press, Cambridge, 2004.

### BẢNG CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| ACO | Tối ưu đ|n kiến |
| AIS | Hệ miễn dịch nh}n tạo |
| BP | Lan truyền ngược |
| CART | C}y ph}n lớp v| hồi quy |
| CRB | Lập luận dựa trên tình huống |
| DNA | Deoxyribonucleic acid |
| EEG | Điện não đồ |
| FPR | Tỷ lệ chấp nhận sai |
| FNR | Tỷ lệ b{c bỏ sai |
| GA | Thuật to{n di truyền |
| K-NN | K l{ng giềng gần nhất |
| ĐTCT | Đặc trưng chi tiết |
| MAP | X{c suất hậu nghiệm cực đại |
| MDP | Qu{ trình quyết định Markov |
| ML | Có khả năng nhất |
| MLP | Perceptron nhiều tầng |
| PAC | Gần đúng x{c suất |
| PCA | Ph}n tích th|nh phần chính |
| RBF | H|m cơ sở b{n kính |
| RNA | Ribonucleic acid |
| SSE | Tổng bình phương sai số |
| SVM | M{y vectơ tựa |
| TSP | B|i to{n người ch|o h|ng |
| VC | Vapnik-Chervonekis |

1. Nếu cần, người đọc có thể tham khảo tài liệu [13] để hiểu thêm phương pháp này [↑](#footnote-ref-1)