

Mục Lục

Danh Sách Các Hình.....	5
Danh Sách Các Bảng.....	7
Lời Mở Đầu.....	8
<i>Chương 1</i>	10
Lý Thuyết Tập Tho.....	10
1.1. Giới thiệu.....	10
1.2. Hệ thông tin.....	11
1.3. Quan hệ bất khả phân biệt	13
1.3.1. Sự dư thừa thông tin.....	13
1.3.2. Quan hệ tương đương - Lớp tương đương.....	13
1.3.3. Thuật toán xác định lớp tương đương.....	15
1.4. Xấp xỉ tập hợp.....	16
1.5. Sự không chắc chắn và hàm thuộc.....	25
1.6. Sự phụ thuộc giữa các tập thuộc tính	27
1.7. Rút gọn thuộc tính.....	28
1.7.1. Khái niệm	28
1.7.2. Ma trận phân biệt và hàm phân biệt	30
1.8. Một số thuật toán hiệu quả.....	36
1.8.1. Lớp tương đương.....	36
1.8.2. Xấp xỉ trên, xấp xỉ dưới.....	37
1.8.3. Vùng dương.....	38
1.8.4. Rút gọn thuộc tính	38
1.8.4.1. Chiến lược Johnson.....	39
1.8.4.2. Chiến lược ngẫu nhiên.....	40
1.8.4.3. Loại bỏ thuộc tính thừa trong một rút gọn.....	41

Chương 2	42
Bài Toán Nhận Dạng Mặt Người	42
2.1. Giới thiệu	42
2.2. Các nghiên cứu trước đây	45
2.3. Mô hình nhận dạng mặt người tiêu biểu	48
2.3.1. Mô hình	48
2.3.2. Rút trích đặc trưng	49
2.3.3. Nhận dạng mẫu	50
2.4. Một số khó khăn trong nhận dạng mặt người	51
2.5. Phương pháp nhận dạng mặt người bằng mặt riêng	54
2.5.1. Mô tả phương pháp	55
2.5.2. Vấn đề tìm các mặt riêng	57
2.5.3. Sử dụng mặt riêng để nhận dạng	60
2.5.4. Tóm tắt phương pháp nhận dạng bằng mặt riêng	62
2.6. Ứng dụng các thuật toán lượng hoá vector trong quá trình phân lớp	63
2.6.1. Giới thiệu	63
2.6.2. Một số thuật toán lượng hoá vector	64
2.6.2.1. Thuật toán LVQ1	64
2.6.2.2. Thuật toán OLVQ1	66
2.6.3. Vấn đề khởi tạo vector tham chiếu	67
Chương 3	70
Ứng Dụng Tập Thô Vào	70
Bài Toán Nhận Dạng Mặt Người	70
3.1. Giới thiệu	70
3.2.1. Phương pháp chung	71
3.2.2. Kết hợp heuristic và lý thuyết tập thô	71
3.2.2.1. Mô tả heuristic	71

3.2.2.2. Thuật toán.....	72
3.2.2.3. Ví dụ minh hoạ.....	73
3.3. Mô hình thử nghiệm.....	77
3.3.1. Tập dữ liệu.....	77
3.3.2. Mô hình 1.....	78
3.3.3. Mô hình 2.....	80
3.3.4. Vấn đề lựa chọn số khoảng rời rạc.....	84
Chương 4	86
Cài Đặt Chương Trình	86
Và Thử Nghiệm	86
4.1. Chương trình cài đặt.....	86
4.1.1. Ngôn ngữ và môi trường.....	86
4.1.2. Tổ chức thư mục mã nguồn.....	86
4.1.3. Một số lớp quan trọng.....	86
1. Lớp bảng quyết định.....	86
2. Các lớp thực hiện rút trích đặc trưng.....	87
3. Lớp rời rạc hoá.....	88
4. Lớp thuật toán tập thô.....	88
5. Các lớp rút gọn thuộc tính.....	88
6. Lớp mạng lượng hoá vector (LVQ).....	90
7. Lớp thuật toán phân loại người láng giềng gần nhất.....	90
4.2. Tổ chức dữ liệu thử nghiệm.....	90
4.3. Hướng dẫn và minh hoạ sử dụng chương trình.....	91
4.3.1. Màn hình chính.....	91
4.3.2. Nhập tập ảnh huấn luyện.....	92
4.3.3. Chọn thuật toán rút gọn thuộc tính.....	94
4.3.4. Quá trình huấn luyện.....	94

4.3.5. Quá trình phân lớp	96
4.3.6. Xem thông tin	97
4.4. Một số kết quả	98
4.4.1. Thư mục Face_10_24_20	98
4.4.2. Thư mục Face_15_24_20	99
4.4.3. Thư mục Face_20_24_20	100
4.4.4. Thư mục Face_25_24_20	101
4.5. Nhận xét kết quả	102
<i>Chương 5</i>	104
Tự Đánh Giá Và Hướng Phát	104
Triển Đề Nghị	104
5.1. Tự đánh giá	104
5.2. Hướng phát triển đề nghị	105
Tài Liệu Tham Khảo	106

Danh Sách Các Hình

Hình 1- 1 : Xấp xỉ tập đối tượng trong Bảng 1- 2 bằng các thuộc tính điều kiện Age và LEMS. Mỗi vùng được thể hiện kèm theo tập các lớp tương đương tương ứng. ..	19
Hình 1- 2 : Ma trận phân biệt của Bảng 1-7.....	31
Hình 1- 3 : Ma trận phân biệt của hệ thông tin Bảng 1-7 xây.....	32
Hình 1- 4 : Ma trận phân biệt giữa các lớp tương đương của.....	33
Hình 1- 5 : Ma trận phân biệt tương đối	33
Hình 1- 6 : Ma trận phân biệt Hình 1-2 sau khi chọn c	34
Hình 2- 1 : Mô hình nhận dạng mặt người tiêu biểu.....	49
Hình 2- 2 : Ảnh với nền phức tạp với	51
Hình 2- 3 : Kết quả của một bộ dò tìm thẳng.....	53
Hình 2- 4 : Vùng “đáng kể nhất” của gương mặt	53
Hình 2- 5 : Kết quả dò tìm trên ảnh có gương mặt được hoá trang	54
Hình 2- 6 : Tập ảnh huấn luyện và ảnh trung bình	58
Hình 2- 7 : Các mặt riêng tương ứng với bảy giá trị riêng lớn nhất	60
Hình 2- 8 : Vector tham chiếu được di chuyển gần với vector dữ liệu hơn – trường hợp hai vector này cùng lớp	66
Hình 2- 9 : Vector tham chiếu được đẩy ra xa vector dữ liệu hơn - trường hợp hai vector này khác lớp	66
Hình 2- 10 : Vector tham chiếu \overrightarrow{OC} khởi tạo không tốt nên sau khi cập nhật thành $\overrightarrow{OC_1}$ thì càng xa vector dữ liệu \overrightarrow{OA} hơn.	68
Hình 3- 1 : Ma trận phân biệt tương đối của hệ thông tin trong Bảng 3-1	75
Hình 3- 2 : Phân chia tập dữ liệu huấn luyện và kiểm tra.....	78
Hình 3- 3 : Ảnh của 10 người đầu tiên trong tập dữ liệu ORL	78

Hình 3- 4 : Giai đoạn huấn luyện tạo tập vector tham chiếu	79
Hình 3- 5 : Giai đoạn phân lớp tập ảnh kiểm tra.....	80
Hình 3- 6 : Giai đoạn huấn luyện tạo tập vector tham chiếu	84
Hình 3- 7 : Giai đoạn phân lớp tập ảnh kiểm tra.....	84

KHOA CNTT – ĐHQHTN

Danh Sách Các Bảng

Bảng 1- 1 : Một hệ thông tin đơn giản	11
Bảng 1- 2 : Một hệ quyết định với $C = \{Age, LEMS\}$ và $D = \{Walk\}$	12
Bảng 1- 3 : Một bảng dữ liệu dư thừa thông tin.....	13
Bảng 1- 4 : Một hệ quyết định điều tra vấn đề da cháy nắng.....	16
Bảng 1- 5 : Hệ thông tin về các thuộc tính của xe hơi.....	20
Bảng 1- 6 : Bảng quyết định dùng minh hoạ hàm thuộc thô	26
Bảng 1- 7 : Hệ thông tin dùng minh hoạ ma trận phân biệt.....	31
Bảng 1- 8 : Một hệ thông tin	35
Bảng 3- 1 : Bảng quyết định cho ví dụ minh hoạ	74
Bảng 3- 2 : Trạng thái ban đầu.....	75
Bảng 3- 3 : Trạng thái tiếp theo khi thêm a	76
Bảng 3- 4 : Trạng thái tiếp theo khi thêm c	76
Bảng 3- 5 : Trạng thái tiếp theo khi thêm d	76
Bảng 4- 1 : Kết quả huấn luyện, kiểm tra tập Face_10_24_20.....	99
Bảng 4- 2 : Kết quả huấn luyện, kiểm tra tập Face_15_24_20.....	100
Bảng 4- 3 : Kết quả huấn luyện, kiểm tra tập Face_20_24_20.....	101
Bảng 4- 4 : Kết quả huấn luyện, kiểm tra tập Face_25_24_20.....	102

Lời Mở Đầu

-----oOo-----

Trong chuyên ngành Trí tuệ nhân tạo, Nhận dạng là một trong những lĩnh vực phát triển sớm nhất và đã tìm được rất nhiều ứng dụng trong cuộc sống, chẳng hạn như dự báo tiềm năng khoáng sản từ ảnh vệ tinh, nhận diện tội phạm qua vân tay, hay gần đây người ta đưa ra khái niệm *ngôi nhà thông minh* với nhiều chức năng tự động hoá hoàn toàn dựa vào khả năng nhận biết các đặc điểm của chủ nhân (như tiếng nói, dáng người,...). Chính vì tầm quan trọng như vậy, lĩnh vực Nhận dạng đã thu hút được sự quan tâm nghiên cứu của nhiều nhà khoa học. Rất nhiều thuật toán và mô hình đã được đưa ra nhằm tăng tối đa hiệu suất của các giai đoạn trong một hệ thống nhận dạng. Trong số đó, vấn đề lựa chọn và rút gọn đặc trưng liên quan trực tiếp đến độ chính xác và tốc độ của hệ thống. Đây cũng là lý do của việc chọn đề tài :

“Khảo Sát Ứng Dụng Của Tập Thô Trong Lựa Chọn Và Rút Gọn Đặc Trưng Cho Bài Toán Nhận Dạng Mặt Người”

Việc lựa chọn lý thuyết Tập thô trong vấn đề nêu trên xuất phát từ những ứng dụng rất thành công của nó trong thực tế như các hệ dự báo hay chuẩn đoán dựa trên luật. Ngoài ra, ý tưởng gắn liền *đối tượng* với *thông tin* cũng như các khái niệm *rút gọn thuộc tính* được đưa ra trong lý thuyết này hứa hẹn khả năng thành công cho hệ thống nhận dạng kết hợp với lý thuyết Tập thô.

Cuối cùng, đối tượng nhận dạng được thử nghiệm trong luận văn này là khuôn mặt bởi đây là đối tượng nghiên cứu khá lý thú với nhiều đặc điểm phong phú mang hàm lượng thông tin cao như cảm xúc, tuổi tác,...và các hệ thống nhận dạng mặt người đang đóng vai trò quan trọng trong bảo mật và an ninh.

Với cách đặt vấn đề như trên, luận văn được cấu trúc thành 5 chương như sau :

- ❖ **Chương 1** : Lý thuyết Tập thô.
- ❖ **Chương 2** : Bài toán nhận dạng mặt người.
- ❖ **Chương 3** : Ứng dụng Tập thô vào bài toán nhận dạng mặt người.
- ❖ **Chương 4** : Cài đặt chương trình và thử nghiệm.
- ❖ **Chương 5** : Tự đánh giá và hướng phát triển đề nghị.

KHOA CNTT – ĐHQGHN

Chương 1

Lý Thuyết Tập Thô

-----oOo-----

1.1. Giới thiệu

Lý thuyết tập thô (*rough set theory*) lần đầu tiên được đề xuất bởi Z. Pawlak và nhanh chóng được xem như một công cụ xử lý các thông tin mơ hồ và không chắc chắn. Phương pháp này đóng vai trò hết sức quan trọng trong lĩnh vực trí tuệ nhận tạo và các ngành khoa học khác liên quan đến nhận thức, đặc biệt là lĩnh vực máy học, thu nhận tri thức, phân tích quyết định, phát hiện và khám phá tri thức từ cơ sở dữ liệu, các hệ chuyên gia, các hệ hỗ trợ quyết định, lập luận dựa trên quy nạp và nhận dạng [5].

Lý thuyết tập thô dựa trên giả thiết rằng để định nghĩa một tập hợp, chúng ta cần phải có thông tin về mọi đối tượng trong tập vũ trụ. Ví dụ, nếu các đối tượng là những bệnh nhân bị một bệnh nhất định thì các triệu chứng của bệnh tạo thành thông tin về bệnh nhân. Như vậy tập thô có quan điểm hoàn toàn khác với quan điểm truyền thống của tập hợp, trong đó mọi tập hợp đều được định nghĩa duy nhất bởi các phần tử của nó mà không cần biết bất kỳ thông tin nào về các phần tử của tập hợp. Rõ ràng, có thể tồn tại một số đối tượng giống nhau ở một số thông tin nào đó, và ta nói chúng có quan hệ bất khả phân biệt với nhau. Đây chính là quan hệ mấu chốt và là điểm xuất phát của lý thuyết tập thô : biên giới của tập thô là không rõ ràng, và để xác định nó chúng ta phải đi xấp xỉ nó bằng các tập hợp khác nhằm mục đích cuối cùng là trả lời được (tất nhiên càng chính xác càng tốt) rằng một đối tượng nào đó có thuộc tập hợp hay không. Lý thuyết tập thô với cách tiếp cận như vậy đã được ứng dụng trong rất nhiều lĩnh vực của đời sống xã hội.

Trong chương này chúng ta sẽ nghiên cứu các khái niệm và ý nghĩa cơ bản của lý thuyết tập thô. Đây là những kiến thức quan trọng cho việc áp dụng tập thô vào bài toán lựa chọn và rút gọn đặc trưng cho bài toán nhận dạng được đề cập trong chương 3.

1.2. Hệ thông tin

Một tập dữ liệu thể hiện dưới dạng bảng, trong đó mỗi dòng thể hiện cho một trường hợp, một sự kiện, một bệnh nhân hay đơn giản là một đối tượng. Mỗi cột của bảng thể hiện một thuộc tính (là một giá trị, một quan sát, một đặc điểm, ...) được “đo lường” cho từng đối tượng. Ngoài ra giá trị của thuộc tính cũng có thể được cung cấp bởi chuyên gia hay bởi người sử dụng. Một bảng như vậy được gọi là một *hệ thông tin* (*information system*).

Một cách hình thức, hệ thông tin là một cặp $\mathcal{A} = (U, A)$ trong đó U là tập hữu hạn không rỗng các đối tượng và được gọi là *tập vũ trụ*, A là tập hữu hạn không rỗng các *thuộc tính* sao cho $a: U \rightarrow V_a$ với mọi $a \in A$. Tập V_a được gọi là *tập giá trị* của thuộc tính a .

Ví dụ 1-1 : Bảng dữ liệu trong *Bảng 1-1* dưới đây cho ta hình ảnh về một hệ thông tin với 7 đối tượng và 2 thuộc tính [1].

	<i>Age</i>	<i>LEMS</i>
x_1	16 – 30	50
x_2	16 – 30	0
x_3	31 – 45	1 – 25
x_4	31 – 45	1 – 25
x_5	46 – 60	26 – 49
x_6	16 – 30	26 – 49
x_7	46 – 60	26 – 49

Bảng 1- 1 : Một hệ thông tin đơn giản

Ta có thể dễ dàng nhận thấy rằng trong bảng trên, các cặp đối tượng x_3 , x_4 và x_5 , x_7 có giá trị bằng nhau tại cả hai thuộc tính. Khi đó ta nói rằng các đối tượng này *không phân biệt* từng đôi đối với tập thuộc tính $\{Age, LEMS\}$. \square

Trong nhiều ứng dụng, tập vũ trụ được phân chia thành các tập đối tượng con bởi một tập các thuộc tính phân biệt được gọi là *tập thuộc tính quyết định*. Nói cách khác tập vũ trụ đã được phân lớp bởi thuộc tính quyết định. Hệ thông tin trong trường hợp này được gọi là một *hệ quyết định*. Như vậy hệ quyết định là một hệ thông tin có dạng $\mathcal{A} = (U, C \cup D)$ trong đó $A = C \cup D$, C và D lần lượt được gọi là *tập thuộc tính điều kiện* và *tập thuộc tính quyết định* của hệ thông tin.

Ví dụ 1-2 : Bảng 1-2 dưới đây thể hiện một hệ quyết định, trong đó tập thuộc tính điều kiện giống như trong Bảng 1-1 và một thuộc tính quyết định $\{Walk\}$ được thêm vào nhận hai giá trị kết xuất là *Yes* và *No* [1].

	<i>Age</i>	<i>LEMS</i>	<i>Walk</i>
x_1	16 – 30	50	Yes
x_2	16 – 30	0	No
x_3	31 – 45	1 – 25	No
x_4	31 – 45	1 – 25	Yes
x_5	46 – 60	26 – 49	No
x_6	16 – 30	26 – 49	Yes
x_7	46 – 60	26 – 49	No

Bảng 1- 2 : Một hệ quyết định với $C = \{Age, LEMS\}$ và $D = \{Walk\}$

Một lần nữa ta thấy rằng, các cặp đối tượng x_3 , x_4 và x_5 , x_7 vẫn có giá trị như nhau tại hai thuộc tính điều kiện, nhưng cặp thứ nhất $\{x_3, x_4\}$ thì có giá trị kết xuất khác nhau (tức giá trị tại thuộc tính quyết định khác nhau), trong khi đó cặp thứ hai $\{x_5, x_7\}$ thì bằng nhau tại thuộc tính quyết định. \square

1.3. Quan hệ bất khả phân biệt

1.3.1. Sự dư thừa thông tin

Một hệ quyết định (hay một bảng quyết định) thể hiện tri thức về các đối tượng trong thế giới thực. Tuy nhiên trong nhiều trường hợp bảng này có thể được tinh giảm do tồn tại ít nhất hai khả năng dư thừa thông tin sau đây :

- Nhiều đối tượng giống nhau, hay không thể phân biệt với nhau lại được thể hiện lặp lại nhiều lần.
- Một số thuộc tính có thể là dư thừa, theo nghĩa khi bỏ đi các thuộc tính này thì thông tin do bảng quyết định cung cấp mà chúng ta quan tâm sẽ không bị mất mát.

Ví dụ 1-3 : Trong bảng ở *Bảng 1-3* dưới đây, nếu chúng ta chỉ quan tâm tới tập thuộc tính $\{a, b, c\}$ của các đối tượng thì ta sẽ có nhận xét : có thể bỏ đi thuộc tính c mà thông tin về các đối tượng vẫn không đổi, chẳng hạn nếu ta có một đối tượng với hai thuộc tính a, b nhận hai giá trị 0, 1 thì có thể nói ngay rằng giá trị của nó tại thuộc tính c là 1.

Object no.	a	b	c	d
1	0	0	1	0
2	0	1	1	1
3	0	1	1	0
4	0	1	1	0
5	1	0	0	1
6	1	0	0	1
7	1	1	0	1
8	1	1	0	1
9	1	1	0	0

Bảng 1- 3 : Một bảng dữ liệu dư thừa thông tin

1.3.2. Quan hệ tương đương - Lớp tương đương

Chúng ta bắt đầu xem xét vấn đề dư thừa thông tin nói trên qua khái niệm *quan hệ tương đương*. Một quan hệ hai ngôi $R \subseteq X \times X$ được gọi là quan hệ tương đương khi và chỉ khi :

- R là quan hệ phản xạ : $xRx, \forall x \in X$.
- R là quan hệ đối xứng : $xRy \Rightarrow yRx, \forall x, y \in X$.
- R là quan hệ bắc cầu : xRy và $yRz \Rightarrow xRz, \forall x, y, z \in X$.

Một quan hệ tương đương R sẽ phân hoạch tập đối tượng thành các *lớp tương đương*, trong đó lớp tương đương của một đối tượng x là tập tất cả các đối tượng có quan hệ R với x .

Tiếp theo, xét hệ thông tin $\mathcal{A} = (U, A)$. Khi đó mỗi tập thuộc tính $B \subseteq A$ đều tạo ra tương ứng một quan hệ tương đương $IND_{\mathcal{A}}$:

$$IND_{\mathcal{A}}(B) = \{(x, x') \in U^2 \mid \forall a \in B, a(x) = a(x')\}$$

$IND_{\mathcal{A}}(B)$ được gọi là quan hệ *B-bất khả phân biệt*. Nếu $(x, x') \in IND_{\mathcal{A}}(B)$ thì các đối tượng x và x' là không thể phân biệt được với nhau qua tập thuộc tính B . Với mọi đối tượng $x \in U$, lớp tương đương của x trong quan hệ $IND_{\mathcal{A}}(B)$ được kí hiệu bởi $[x]_B$. Nếu không bị nhầm lẫn ta viết $IND(B)$ thay cho $IND_{\mathcal{A}}(B)$. Cuối cùng, quan hệ B -bất khả phân biệt phân hoạch tập đối tượng U thành các lớp tương đương mà ta kí hiệu là $U \mid IND(B)$.

Ví dụ 1-4 : Tập thuộc tính $\{a, b, c\}$ trong *Bảng 1-3* phân tập đối tượng $\{1, 2, \dots, 9\}$ thành tập lớp tương đương sau :

$$U \mid IND(B) = \{\{1\}, \{2, 3, 4\}, \{5, 6, 7\}, \{8, 9\}\}$$

Ta thấy, chẳng hạn, do đối tượng 2 và đối tượng 3 thuộc cùng một lớp tương đương nên chúng không phân biệt được với nhau qua tập thuộc tính $\{a, b, c\}$. \square

Ví dụ 1-5 : Trong ví dụ này chúng ta sẽ xem xét các quan hệ bất khả phân biệt được định nghĩa trong *Bảng 1-2*.

Chẳng hạn, xét tại thuộc tính $\{LEMS\}$, các đối tượng x_3, x_4 có cùng giá trị 1–25 nên thuộc cùng lớp tương đương định bởi quan hệ $IND(\{LEMS\})$, hay chúng bất khả phân biệt qua tập thuộc tính $\{LEMS\}$. Tương tự như vậy là ba đối tượng x_5, x_6 và x_7 cùng thuộc vào một lớp tương đương định bởi quan hệ $IND(\{LEMS\})$ tương ứng với giá trị thuộc tính $LEMS$ bằng 26–49.

Quan hệ IND định ra ba phân hoạch sau của tập các đối tượng trong vũ trụ :

$$IND(\{Age\}) = \{\{x_1, x_2, x_6\}, \{x_3, x_4\}, \{x_5, x_7\}\}$$

$$IND(\{LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_6, x_7\}\}$$

$$IND(\{Age, LEMS\}) = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5, x_7\}, \{x_6\}\}$$

□

1.3.3. Thuật toán xác định lớp tương đương

Vào :

- Tập đối tượng O
- Tập thuộc tính B

Ra :

- Tập các lớp tương đương L

Thuật toán :

Bước 1: $L = \emptyset$

Bước 2: Nếu $O = \emptyset$

Thì : Thực hiện bước 5.

Ngược lại : Thực hiện bước 3.

Hết nếu

Bước 3: Xét $x \in O$

$$P = \{x\}$$

$$O = O \setminus \{x\}$$

Với mọi phần tử $y \in O$:

Nếu x và y không thể phân biệt được qua tập thuộc tính B

$$\underline{\text{Thì}} : P = P \cup \{y\}$$

$$O = O \setminus \{y\}$$

Hết nếu

Hết với mọi

$$L = L \cup \{P\}$$

Bước 4: Thực hiện bước 2.

Bước 5: Kết thúc.

1.4. Xấp xỉ tập hợp

Như trên đã nói, một quan hệ tương đương cho ta một sự phân hoạch các đối tượng của tập vũ trụ. Các lớp tương đương này có thể được sử dụng để tạo nên các tập con của tập vũ trụ. Các tập con này thường chứa các đối tượng có cùng giá trị tại tập các thuộc tính quyết định. Trong trường hợp này ta nói rằng các *khái niệm*, hay tập các giá trị tại tập các thuộc tính quyết định, có thể được mô tả một cách rõ ràng thông qua tập các giá trị tại tập các thuộc tính điều kiện. Để làm rõ ý tưởng quan trọng này ta xem ví dụ dưới đây.

Ví dụ 1-6 : Xét hệ quyết định điều tra vấn đề da cháy nắng sau đây

STT	Trọng lượng	Dùng thuốc	Kết quả
1	Nhẹ	Có	Không cháy nắng
2	Nhẹ	Có	Không cháy nắng
3	Nặng	Không	Cháy nắng
4	Trung bình	Không	Không cháy nắng

Bảng 1- 4 : Một hệ quyết định điều tra vấn đề da cháy nắng

Trong hệ quyết định trên, thuộc tính *Kết quả* là thuộc tính quyết định và hai thuộc tính giữa là thuộc tính điều kiện. Tập thuộc tính điều kiện $C = \{\text{Trọng lượng, Dùng thuốc}\}$ phân hoạch tập các đối tượng thành các lớp tương đương :

$$U \mid IND(C) = \{\{1,2\}, \{3\}, \{4\}\}$$

Nhận xét rằng tất cả các đối tượng thuộc cùng một lớp tương đương đều có cùng giá trị tại thuộc tính quyết định. Do đó ta có thể mô tả thuộc tính quyết định như sau :

- Kết quả sẽ là *không cháy nắng* nếu và chỉ nếu trọng lượng là *nhẹ* và có dùng thuốc hoặc trọng lượng *trung bình* và *không* dùng thuốc.
- Kết quả sẽ là *cháy nắng* nếu và chỉ nếu trọng lượng là *nặng* và *không* dùng thuốc.

Ta nói hai khái niệm *Cháy nắng* và *Không cháy nắng* trong thuộc tính *Kết quả* có thể được định nghĩa rõ ràng qua 2 thuộc tính *Trọng lượng* và *Dùng thuốc*. Tuy vậy không phải lúc nào cũng có thể định nghĩa một khái niệm nào đó một cách rõ ràng như vậy. Chẳng hạn với bảng quyết định trong *Bảng 1-2*, khái niệm *Walk* không thể định nghĩa rõ ràng qua 2 thuộc tính điều kiện *Age* và *LEMS* : hai đối tượng x_3 và x_4 thuộc cùng một lớp tương đương tạo bởi 2 thuộc tính điều kiện nhưng lại có giá trị khác nhau tại thuộc tính *Walk*, vì vậy nếu một đối tượng nào đó có $(Age, LEMS) = (31-45, 1-25)$ thì ta vẫn không thể biết chắc chắn giá trị của nó tại thuộc tính *Walk* (*Yes* hay *No* ?), nói cách khác ta sẽ không thể có một luật như sau : “*Walk* là *Yes* nếu *Age* là 31-45 và *LEMS* là 1-25”. Và đây chính là nơi mà khái niệm tập thô được sử dụng! .

Mặc dù không thể mô tả khái niệm *Walk* một cách rõ ràng nhưng căn cứ vào tập thuộc tính $\{Age, LEMS\}$ ta vẫn có thể chỉ ra được chắc chắn một số đối tượng có *Walk* là *Yes*, một số đối tượng có *Walk* là *No*, còn lại là các đối tượng thuộc về biên giới của 2 giá trị *Yes* và *No*, cụ thể :

- Nếu đối tượng nào có giá trị tại tập thuộc tính $\{Age, LEMS\}$ thuộc tập $\{\{16-30, 50\}, \{16-30, 26-49\}\}$ thì nó có *Walk* là *Yes*.

- Nếu đối tượng nào có giá trị tại tập thuộc tính $\{Age, LEMS\}$ thuộc tập $\{\{16 - 30, 0\}, \{46 - 60, 26 - 49\}\}$ thì nó có *Walk* là *No*.
- Nếu đối tượng nào có giá trị tại tập thuộc tính $\{Age, LEMS\}$ thuộc tập $\{\{31 - 45, 1 - 25\}\}$ thì nó có *Walk* là *Yes* hoặc *No*. Những đối tượng này, như nói ở trên thuộc về biên giới của 2 giá trị *Yes* và *No*.

Những khái niệm trên được thể hiện một cách hình thức như sau.

Cho hệ thông tin $\mathcal{A} = (U, A)$, tập thuộc tính $B \subseteq A$, tập đối tượng $X \subseteq U$. Chúng ta có thể xấp xỉ tập hợp X bằng cách chỉ sử dụng các thuộc tính trong B từ việc xây dựng các tập hợp *B-xấp xỉ dưới* và *B-xấp xỉ trên* được định nghĩa như sau :

- *B-xấp xỉ dưới* của tập X : $\underline{B}X = \{x \mid [x]_B \subseteq X\}$
- *B-xấp xỉ trên* của tập X : $\overline{B}X = \{x \mid [x]_B \cap X \neq \emptyset\}$

Tập hợp $\underline{B}X$ là tập các đối tượng trong U mà sử dụng các thuộc tính trong B ta có thể biết chắc chắn được chúng là các phần tử của X .

Tập hợp $\overline{B}X$ là tập các đối tượng trong U mà sử dụng các thuộc tính trong B ta chỉ có thể nói rằng chúng có thể là các phần tử của X .

Tập hợp $BN_B(X) = \overline{B}X \setminus \underline{B}X$ được gọi là *B-biên* của tập X và chứa những đối tượng mà sử dụng các thuộc tính của B ta không thể xác định được chúng có thuộc tập X hay không.

Tập hợp $U \setminus \overline{B}X$ được gọi là *B-ngoài* của tập X , gồm những đối tượng mà sử dụng tập thuộc tính B ta biết chắc chắn chúng không thuộc tập X .

Một tập hợp được gọi là *thô* nếu đường biên của nó là không rỗng, ngược lại ta nói tập này là *rõ*. Lưu ý rằng do khái niệm biên của một tập đối tượng gắn liền với một tập thuộc tính nào đó nên khái niệm thô hay rõ ở đây cũng gắn liền với tập thuộc tính đó.

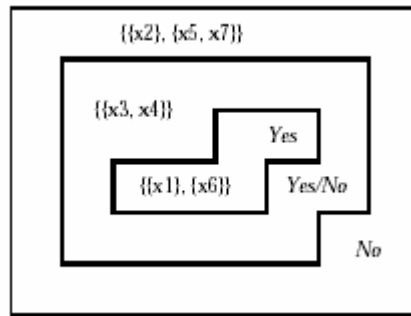
Trong đa số trường hợp, người ta luôn muốn hình thành các định nghĩa của các lớp quyết định từ các thuộc tính điều kiện.

Ví dụ 1-7 :

Chương 1 – Lý thuyết Tập thô

Xét Bảng 1-2 ở trên với tập đối tượng $W = \{x \mid Walk(x) = Yes\} = \{x_1, x_4, x_6\}$ và tập thuộc tính $B = \{Age, LEMS\}$. Khi đó ta nhận được các vùng xấp xỉ sau đây của W thông qua B :

$$\begin{aligned} \underline{BW} &= \{x_1, x_6\} \quad , \quad \overline{BW} = \{x_1, x_3, x_4, x_6\} \\ BN_B(W) &= \{x_3, x_4\} \quad , \quad U \setminus \overline{BW} = \{x_2, x_5, x_7\} \end{aligned}$$



Hình 1- 1 : Xấp xỉ tập đối tượng trong Bảng 1- 2 bằng các thuộc tính điều kiện Age và LEMS. Mỗi vùng được thể hiện kèm theo tập các lớp tương đương tương ứng.

□

Ví dụ 1-8 : Ta xét một ví dụ khác với bảng giá trị về thuộc tính của xe hơi như sau :

Đối tượng	Model	Cylinder	Door	Power	Weight	Mileage
1	USA	6	2	High	Medium	Medium
2	USA	6	4	Medium	Medium	Medium
3	USA	4	2	Medium	Medium	Medium
4	USA	4	2	Medium	Medium	Medium
5	USA	4	2	High	Medium	Medium
6	USA	6	4	High	Medium	Medium
7	USA	4	2	High	Medium	Medium
8	USA	4	2	High	Light	High

9	Japan	4	2	Low	Light	High
10	Japan	4	2	Medium	Medium	High
11	Japan	4	2	High	Medium	High
12	Japan	4	2	Low	Medium	High
13	Japan	4	2	Medium	Medium	High
14	USA	4	2	Medium	Medium	High

Bảng 1- 5 : Hệ thông tin về các thuộc tính của xe hơi

Ta có tập vũ trụ $U = \{1,2,...,14\}$. Giả sử chọn tập thuộc tính $B = \{Cylinder, Power, Weight\}$ và chọn thuộc tính quyết định là $D = Mileage$. Như vậy thuộc tính quyết định gồm 2 khái niệm $D_{Medium} = "Mileage = Medium"$ và $D_{High} = "Mileage = High"$.

$$D_{Medium} = \{1,2,3,4,5,6,7\}$$

$$D_{High} = \{8,9,10,11,12,13,14\}$$

Các lớp tương đương ứng với quan hệ $IND(B)$ là : $E_1 = \{1,6\}$, $E_2 = \{2\}$, $E_3 = \{3,4,10,13,14\}$, $E_4 = \{5,7,11\}$, $E_5 = \{8\}$, $E_6 = \{9\}$ và $E_7 = \{12\}$.

Xấp xỉ trên và xấp xỉ dưới của D_{Medium} và D_{High} là :

$$\underline{BD}_{Medium} = \{E_1, E_2\} = \{1,6,2\}$$

$$\overline{BD}_{Medium} = \{E_1, E_2, E_3, E_4\} = \{1,6,2,3,4,10,13,14,5,7,11\}$$

$$\underline{BD}_{High} = \{E_5, E_6, E_7\} = \{8,9,12\}$$

$$\overline{BD}_{High} = \{E_3, E_4, E_5, E_6, E_7\} = \{3,4,10,13,14,5,7,11,8,9,12\}$$

□

Một số tính chất của các tập hợp xấp xỉ

1. $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$
2. $\underline{B}(\emptyset) = \overline{B}(\emptyset) = \emptyset$, $\underline{B}(U) = \overline{B}(U) = U$

3. $\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$
4. $\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$
5. Nếu $X \subseteq Y$ thì $\underline{B}(X) \subseteq \underline{B}(Y), \overline{B}(X) \subseteq \overline{B}(Y)$
6. $\underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y)$
7. $\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$
8. $\underline{B}(U \setminus X) = U \setminus \overline{B}(X)$
9. $\overline{B}(U \setminus X) = U \setminus \underline{B}(X)$
10. $\underline{B}(\underline{B}(X)) = \overline{B}(\underline{B}(X)) = \underline{B}(X)$
11. $\overline{B}(\overline{B}(X)) = \underline{B}(\overline{B}(X)) = \overline{B}(X)$

Ta chứng minh một số định lý điển hình.

3. Từ định nghĩa xấp xỉ trên ta có:

$$o \in \overline{B}(X \cup Y) \Leftrightarrow \exists P \in U \mid IND(B) : (o \in P, P \cap (X \cup Y) \neq \emptyset)$$

Mặt khác : $P \cap (X \cup Y) \neq \emptyset \Leftrightarrow P \cap X \neq \emptyset$ hoặc $P \cap Y \neq \emptyset$.

Do đó :

$$\begin{aligned} o \in \overline{B}(X \cup Y) &\Leftrightarrow (o \in P, P \cap X \neq \emptyset) \text{ hoặc } (o \in P, P \cap Y \neq \emptyset) \\ &\Leftrightarrow (o \in \overline{B}(X)) \text{ hoặc } (o \in \overline{B}(Y)) \\ &\Leftrightarrow o \in \overline{B}(X) \cup \overline{B}(Y) \end{aligned}$$

\Rightarrow (dpcm)

4. Chứng minh tương tự 3.
5. Chứng minh : $(X \subseteq Y) \Rightarrow (\underline{B}(X) \subseteq \underline{B}(Y))$

Giả sử : $X \subseteq Y$

Xét $o \in \underline{B}(X)$. Khi đó : $\exists P, P \in U \mid IND(B) : o \in P, P \subseteq X$.

Mà $X \subseteq Y$ nên $P \subseteq Y$. Nhưng theo định nghĩa tập xấp xỉ dưới :

$$\underline{B}(Y) = \{x \mid x \in P, P \in U \mid IND(B), P \subseteq Y\}$$

Nên : $P \subseteq \underline{B}(Y)$, từ đó : $o \in \underline{B}(Y)$

Vậy : $\underline{B}(X) \subseteq \underline{B}(Y)$. Tương tự ta chứng minh được $\overline{B}(X) \subseteq \overline{B}(Y)$

6. Xét $o \in \underline{B}(X) \cup \underline{B}(Y) \Rightarrow \exists P, P \in U \mid IND(B), o \in P, (P \subseteq X \vee P \subseteq Y)$

$\Rightarrow P \subseteq X \cup Y$. Mặt khác theo định nghĩa tập xấp xỉ dưới :

$$\underline{B}(X \cup Y) = \{x \mid x \in P, P \in U \mid IND(B), P \subseteq X \cup Y\}$$

Vậy : $P \subseteq \underline{B}(X \cup Y)$, từ đó $o \in \underline{B}(X \cup Y)$

\Rightarrow đpcm.

7. Chứng minh tương tự 6

8. Ta có : $\underline{B}(U \setminus X) = \{\bigcup P \mid P \in U \mid IND(B), P \subseteq U \setminus X\}$

$$= U \setminus \{\bigcup P \mid P \in U \mid IND(B), P \cap X \neq \emptyset\}$$

$$= U \setminus \overline{B}(X) \quad (\text{đpcm}).$$

9. Chứng minh tương tự hoặc có thể suy ra từ 8.

10. Từ định nghĩa của tập xấp xỉ dưới :

$$\underline{B}(\underline{B}(X)) = \{x \in U \mid [x]_B \subseteq \underline{B}(X)\}$$

$$= \{x \in U \mid [x]_B \subseteq X\}, \text{ vì } \underline{B}(X) \subseteq X$$

$$= \underline{B}(X)$$

Tương tự : $\overline{B}(\underline{B}(X)) = \underline{B}(X)$. Vậy ta có đpcm.

11. Chứng minh tương tự 10. □

Dựa vào ý nghĩa của các xấp xỉ trên và xấp xỉ dưới, người ta định nghĩa bốn lớp cơ bản của các tập thô, hay bốn hình thức của sự mơ hồ (*vagueness*) :

(a) X được gọi là B -định nghĩa được một cách thô (*roughly B-definable*) nếu và chỉ nếu $\underline{B}(X) \neq \emptyset$ và $\overline{B}(X) \neq U$.

(b) X được gọi là B -không định nghĩa được một cách nội vi (*internally B-undefinable*) nếu và chỉ nếu $\underline{B}(X) = \emptyset$ và $\overline{B}(X) \neq U$.

(c) X được gọi là B -không định nghĩa được một cách ngoại vi (*externally B-undefinable*) nếu và chỉ nếu $\underline{B}(X) \neq \emptyset$ và $\overline{B}(X) = U$.

(d) X được gọi là B -không định nghĩa được một cách hoàn toàn (totally B -undefinable) nếu và chỉ nếu $\underline{B}(X) = \emptyset$ và $\overline{B}(X) = U$.

Các khái niệm trên có thể diễn tả như sau :

- X là B -định nghĩa được một cách thô nghĩa là : với sự giúp đỡ của tập thuộc tính B ta có thể chỉ ra một số đối tượng của U thuộc về tập X và một số đối tượng của U thuộc về $U \setminus X$.
- X là B -không định nghĩa được một cách nội vi nghĩa là : sử dụng tập thuộc tính B ta có thể chỉ ra một số đối tượng của U thuộc về $U \setminus X$, nhưng lại không thể chỉ ra được các đối tượng thuộc về X .
- X là B -không định nghĩa được một cách ngoại vi nghĩa là : sử dụng tập thuộc tính B ta có thể chỉ ra một số đối tượng của U thuộc về X , nhưng không chỉ ra được các đối tượng thuộc về $U \setminus X$.
- X là B -không định nghĩa được một cách hoàn toàn nghĩa là : sử dụng tập thuộc tính B ta không thể chỉ ra bất kỳ đối tượng nào của U thuộc về X hay thuộc về $U \setminus X$.

Cuối cùng, một tập thô có thể được định lượng bởi hệ số :

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|}$$

được gọi là độ chính xác của xấp xỉ, trong đó $|X|$ chỉ số phần tử của tập X . Rõ ràng $0 < \alpha_B(X) \leq 1$. Nếu $\alpha_B(X) = 1$ thì X là rõ (chính xác) đối với tập thuộc tính B . Ngược lại, nếu $\alpha_B(X) < 1$ thì X là thô (mơ hồ) đối với tập thuộc tính B .

Chúng ta kết thúc mục này với thuật toán xác định các xấp xỉ trên và xấp xỉ dưới của một tập đối tượng theo một tập thuộc tính cho trước.

Thuật toán xác định xấp xỉ dưới

Vào :

- Tập các đối tượng X

- Tập các thuộc tính B

Ra :

- Tập các đối tượng $\underline{B}X$

Thuật toán :

Bước 1 : Khởi tạo $\underline{B}X = \emptyset$.

Xác định tập các phân hoạch P của tập vũ trụ U tạo bởi B .

Bước 2 : $U_1 = U$

Nếu $U_1 \neq \emptyset$

Thì : Thực hiện bước 3.

Ngược lại : Thực hiện bước 5

Hết nếu

Bước 3 : Xét $x \in U_1$

Tìm phân hoạch $P_i \in P$ sao cho : $x \in P_i$.

Nếu $P_i \subseteq X$

Thì : $\underline{B}X = \underline{B}X \cup P_i$

Hết nếu

$U_1 = U_1 \setminus P_i$.

Bước 4 : Thực hiện bước 2.

Bước 5 : Kết thúc

Thuật toán xác định xấp xỉ trên

Vào :

- Tập các đối tượng X
- Tập các thuộc tính B

Ra :

- Tập các đối tượng $\overline{B}X$

Thuật toán :

Bước 1 : Khởi tạo $\overline{B}X = \emptyset$.

Xác định tập các phân hoạch P của tập vũ trụ U tạo bởi B .

Bước 2 : $X_1 = X$

Nếu $X_1 \neq \emptyset$

Thì : Thực hiện bước 3.

Ngược lại : Thực hiện bước 5

Hết nếu

Bước 3 : Xét $x \in X_1$.

Tìm phân hoạch $P_i \in P$ sao cho : $x \in P_i$.

$$\overline{B}X = \overline{B}X \cup P_i$$

Với mọi $p \in P_i \cap X_1$

$$X_1 = X_1 \setminus \{p\}$$

Hết với mọi

Bước 4 : Thực hiện bước 2.

Bước 5 : Kết thúc.

1.5. Sự không chắc chắn và hàm thuộc

Chúng ta đã biết $BN_B(X)$ là tập các đối tượng trong tập vũ trụ U mà bằng cách sử dụng tập thuộc tính B ta không thể xác định được chắc chắn chúng có thuộc tập đối tượng X hay không. Do đó, sự không chắc chắn trong ngữ cảnh này gắn với một câu hỏi về *độ thuộc* (*membership*) của các phần tử vào một tập hợp.

Trong lý thuyết tập hợp cổ điển, một phần tử hoặc là thuộc vào tập hợp hoặc không. Như vậy hàm thuộc tương ứng là một hàm đặc trưng cho tập hợp, nghĩa là hàm sẽ nhận giá trị 0 và 1 tương ứng.

Trong lý thuyết tập thô, *hàm thuộc thô* μ_X^B là khái niệm dùng để đo mức độ thuộc của đối tượng x trong tập vũ trụ U vào tập các đối tượng $X \subseteq U$, và được tính bởi

Chương 1 – Lý thuyết Tập thô

mức độ giao nhau giữa tập X và lớp tương đương $[x]_B$ mà đối tượng x thuộc về. Một cách hình thức, ta có :

$$\mu_X^B : U \rightarrow [0,1]$$

$$x \mapsto \frac{|[x]_B \cap X|}{|[x]_B|}$$

Một số tính chất của hàm thuộc thô

1. $\mu_X^B(x) = 1 \Leftrightarrow x \in \underline{B}X$
2. $\mu_X^B(x) = 0 \Leftrightarrow x \in U - \overline{B}X$
3. $0 < \mu_X^B(x) < 1 \Leftrightarrow x \in BN_B(X)$
4. $\mu_X^B(x) = \mu_X^B(y)$ nếu $(x, y) \in IND(B)$
5. $\mu_{U-X}^B(x) = 1 - \mu_X^B(x), \forall x \in U$
6. $\mu_{X \cup Y}^B(x) = \max(\mu_X^B(x), \mu_Y^B(x)), \forall x \in U$
7. $\mu_{X \cap Y}^B(x) = \min(\mu_X^B(x), \mu_Y^B(x)), \forall x \in U$

Ví dụ 1-9 : Xét bảng quyết định dưới đây

	A_0	A_1	A_2	A_3	A_4
x_0	1	A	2	34	Black
x_1	2	A	3	23	Blue
x_2	4	B	3	32	White
x_3	1	B	2	12	Black
x_4	3	B	1	32	Blue
x_5	1	B	4	12	Black

Bảng 1- 6 : Bảng quyết định dùng minh hoạ hàm thuộc thô

Xét tập thuộc tính $B = \{A_0, A_1\}$ và tập đối tượng $X = \{x_0, x_1, x_3\}$. Lớp tương đương tương ứng với quan hệ $IND(B)$ là : $E_1 = \{x_0\}$, $E_2 = \{x_1\}$, $E_3 = \{x_2\}$, $E_4 = \{x_3, x_5\}$, $E_5 = \{x_4\}$.

Áp dụng định nghĩa hàm thuộc thô, ta thu được :

$$\mu_X^B(x_0) = \frac{|\{x_0\}|}{|\{x_0\}|} = 1.0$$

$$\mu_X^B(x_0) = \frac{|\{x_3\}|}{|\{x_3, x_5\}|} = 0.5 \quad \square$$

Từ định nghĩa hàm thuộc thô, hai khái niệm xấp xỉ trên và xấp xỉ dưới có thể được xây dựng một cách tổng quát tương ứng với một độ rõ bất kỳ $\pi \in (\frac{1}{2}, 1]$ như sau :

$$\underline{B}_\pi(X) = \{x \mid \mu_X^B(x) \geq \pi\}$$

$$\overline{B}_\pi(X) = \{x \mid \mu_X^B(x) \geq 1 - \pi\}$$

Lưu ý rằng hai khái niệm xấp xỉ trên và xấp xỉ dưới mà ta đã xây dựng trong phần 1.4 tương ứng với trường hợp độ rõ $\pi = 1.0$.

1.6. Sự phụ thuộc giữa các tập thuộc tính

Một vấn đề quan trọng trong phân tích dữ liệu là khám phá sự phụ thuộc giữa các thuộc tính. Một cách trực giác, một tập thuộc tính D được cho là phụ thuộc hoàn toàn vào tập thuộc tính C , ký hiệu $C \Rightarrow D$, nếu tất cả các giá trị của các thuộc tính trong D có thể được xác định duy nhất bởi các giá trị của các thuộc tính trong C . Nói cách khác, D phụ thuộc hoàn toàn vào C nếu tồn tại một ánh xạ từ các giá trị của tập C tới các giá trị của tập D . Khái niệm phụ thuộc thuộc tính được thể hiện dưới dạng hình thức như sau.

Cho C và D là các tập con của tập thuộc tính A . Ta nói D phụ thuộc C với độ phụ thuộc k ($0 \leq k \leq 1$), kí hiệu $C \Rightarrow_k D$ nếu :

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|}$$

trong đó

$$POS_C(D) = \bigcup_{X \in U \setminus IND(D)} \underline{C}(X)$$

được gọi là *C-vùng dương của D*. Đây là tập các đối tượng của U mà bằng cách sử dụng tập thuộc tính C ta có thể phân chúng một cách duy nhất vào các phân hoạch của U theo tập thuộc tính D .

Dễ dàng thấy rằng :

$$\gamma(C, D) = \sum_{X \in U \setminus IND(D)} \frac{|\underline{C}X|}{|U|}$$

Nếu $k = 1$ thì ta nói D phụ thuộc hoàn toàn vào C , ngược lại nếu $k < 1$ thì ta nói D phụ thuộc một phần vào C với độ phụ thuộc k .

Có thể nhận thấy rằng nếu D phụ thuộc hoàn toàn vào C thì $IND(C) \subseteq IND(D)$. Điều này có nghĩa là các phân hoạch tạo ra bởi tập thuộc tính C mịn hơn các phân hoạch tạo ra bởi D .

1.7. Rút gọn thuộc tính

1.7.1. Khái niệm

Trong phần 1.3 chúng đã đề cập đến hai khả năng dư thừa trong một hệ thông tin, đó là :

- Các đối tượng giống nhau theo một tập thuộc tính đang quan tâm được lặp lại nhiều lần.
- Một số thuộc tính có thể được bỏ đi mà thông tin chúng ta đang quan tâm do bảng quyết định cung cấp vẫn không bị mất mát.

Với trường hợp thứ nhất, khái niệm lớp tương đương hiển nhiên cho ta một tiếp cận tự nhiên trong việc tinh giảm thông tin cần lưu trữ trong một hệ thông tin : chỉ cần sử dụng một đối tượng để đại diện cho mỗi lớp tương đương. Trong phần này chúng ta

Chương 1 – Lý thuyết Tập thô

ngiên cứu tiếp cận cho loại dư thừa thông tin thứ hai, đó là chỉ giữ lại những thuộc tính bảo toàn quan hệ bất khả phân biệt, và do đó bảo toàn khả năng xấp xỉ tập hợp trong một hệ thông tin.

Xét hệ thông tin $\mathcal{A} = (U, A)$ và hai tập thuộc tính $P, Q \subseteq A$. Thuộc tính $a \in P$ được gọi là *có thể bỏ được* (*dispensible*) trong P nếu $IND(P) = IND(P - \{a\})$, ngược lại ta nói a là *không thể bỏ được* (*indispensible*) trong P . Rõ ràng thuộc tính có thể bỏ được không làm tăng / giảm khả năng phân loại khi có / không có mặt thuộc tính đó trong P . Tập tất cả các thuộc tính không thể bỏ được trong P được gọi là *lõi* (*core*) của P , ký hiệu $CORE(P)$. Lưu ý rằng lõi có thể là tập rỗng, và khi đó mọi tập con của P với lực lượng bằng $card(P) - 1$ đều giữ nguyên khả năng phân loại của P .

Khi loại ra khỏi P một số thuộc tính có thể bỏ được thì ta được một tập *rút gọn* của P . Nói cách khác, rút gọn của một tập thuộc tính P là tập thuộc tính $B \subseteq P$ giữ nguyên khả năng phân loại của P , hay $IND(B) = IND(P)$. Dễ dàng thấy rằng, vì lõi của P là tập các thuộc tính không thể bỏ được của P nên tất cả các rút gọn của P đều chứa tập thuộc tính lõi.

Một rút gọn B của tập thuộc tính P được gọi là *rút gọn hoàn toàn* nếu với mọi tập thuộc tính $B' \subset B$, B' không là rút gọn của P . Như vậy rút gọn hoàn toàn là tập thuộc tính nhỏ nhất trong tất cả các rút gọn có thể có của P và được ký hiệu là $RED(P)$.

Tính chất : Tập thuộc tính lõi của P là giao của tất cả các rút gọn hoàn toàn của P , tức là : $CORE(P) = \bigcap RED(P)$

Để minh hoạ cho những khái niệm trên, ta xét ví dụ sau.

Ví dụ 1-10 : Xét Bảng 1-3 với tập thuộc tính $P = \{a, b, c\}$. Ta có :

$$U \mid IND(P) = \{\{1\}, \{2,3,4\}, \{5,6\}, \{7,8,9\}\}$$

$$U \mid IND(\{a\}) = \{\{1,2,3,4\}, \{5,6,7,8,9\}\}$$

$$U \mid IND(\{b\}) = \{\{1,5,6\}, \{2,3,4,7,8,9\}\}$$

$$U \mid IND(\{c\}) = \{\{1,2,3,4\}, \{5,6,7,8,9\}\}$$

$$U \mid IND(\{a,b\}) = \{\{1\}, \{2,3,4\}, \{5,6\}, \{7,8,9\}\}$$

$$U \mid IND(\{b,c\}) = \{\{1\}, \{2,3,4\}, \{5,6\}, \{7,8,9\}\}$$

$$U \mid IND(\{c,a\}) = \{\{1,2,3,4\}, \{5,6,7,8,9\}\}$$

Vì $\{a,b\}$ và $\{b,c\}$ là hai tập thuộc tính con nhỏ nhất của P và giữ nguyên khả năng phân loại tập U của P , tức là : $U \mid IND(\{a,b\}) = U \mid IND(\{b,c\}) = U \mid IND(P)$ nên chúng là hai rút gọn hoàn toàn của P . Lỗi của P là $\{b\}$. \square

Thuộc tính a được gọi là Q - có thể bỏ được (Q – dispensible) trong P nếu $POS_P(Q) = POS_{P-\{a\}}(Q)$, ngược lại là Q - không thể bỏ được (Q -indispensible). Tập tất cả các thuộc tính Q - không thể bỏ được trong P được gọi là Q - lõi tương đối (Q – relative core) của P hay Q - lõi (Q – core) của P và được ký hiệu là $CORE_Q(P)$.

Tập thuộc tính $B \subseteq P$ được gọi là Q - rút gọn (Q – reduct) của P khi và chỉ khi $POS_B(Q) = POS_P(Q)$. Một tập Q - rút gọn B của P là Q - rút gọn hoàn toàn nếu với mọi tập thuộc tính $B' \subset B$, B' không là Q - rút gọn của P . Như vậy, Q - rút gọn hoàn toàn của P là tập thuộc tính nhỏ nhất trong tất cả các Q - rút gọn của P và được ký hiệu là $RED_Q(P)$.

Tính chất : Tập thuộc tính Q - lõi của P là giao của tất cả các tập thuộc tính Q - rút gọn tương đối của P , tức là : $CORE_Q(P) = \bigcap RED_Q(P)$.

Ví dụ 1-11 : Xét hệ thông tin trong Bảng 1-6 với tập thuộc tính $P = \{A_0, A_1, A_2\}$ và $Q = \{A_4\}$. Khi đó : $CORE_Q(P) = \emptyset$ và $RED_Q(P) = \{\{A_0\}, \{A_1, A_1\}\}$. \square

1.7.2. Ma trận phân biệt và hàm phân biệt

Phần trên cung cấp các khái niệm về rút gọn thuộc tính trong một hệ thông tin, tuy nhiên chúng chưa thật sự rõ nét và trực quan. Trong phần này chúng ta sẽ thấy được bản chất của một rút gọn của tập thuộc tính, và đây là cơ sở để hiểu được các thuật toán tìm tập rút gọn trong một hệ thông tin.

Chương 1 – Lý thuyết Tập thô

Xét hệ thông tin $\mathcal{A} = (U, A)$ có n đối tượng. Ma trận phân biệt của \mathcal{A} là ma trận đối xứng kích thước $n \times n$ có các phần tử c_{ij} được cho như sau :

$$c_{ij} = \{a \in A \mid a(x_i) \neq a(x_j)\} \text{ với } i, j = 1, 2, \dots, n$$

Như vậy mỗi phần tử c_{ij} của ma trận phân biệt là tập hợp các thuộc tính để phân biệt hai đối tượng x_i và x_j .

Ví dụ 1-12 : Xét một hệ thông tin đơn giản trong *Bảng 1-7* với 3 thuộc tính và 4 đối tượng. Phần tử tại dòng 1 cột 3 cũng như phần tử tại dòng 3 cột 1 là tập thuộc tính $\{a, c\}$ nói lên rằng hai đối tượng x_1 và x_3 nhận giá trị khác nhau tại hai thuộc tính a và c .

	a	b	c
x_1	1	0	1
x_2	1	1	2
x_3	0	0	2
x_4	1	0	1

Bảng 1- 7 : Hệ thông tin dùng minh họa ma trận phân biệt

Hệ thông tin trên sẽ có ma trận phân biệt kích thước 4×4 như sau :

	x_1	x_2	x_3	x_4
x_1	$\{\}$	$\{b, c\}$	$\{a, c\}$	$\{\}$
x_2	$\{b, c\}$	$\{\}$	$\{a, b\}$	$\{b, c\}$
x_3	$\{a, c\}$	$\{a, b\}$	$\{\}$	$\{a, c\}$
x_4	$\{\}$	$\{b, c\}$	$\{a, c\}$	$\{\}$

Hình 1- 2 : Ma trận phân biệt của *Bảng 1-7*

□

Ma trận phân biệt không chỉ được định nghĩa trên tập tất cả các thuộc tính của hệ thông tin mà còn có thể được xây dựng trên một tập thuộc tính $B \subseteq A$ bất kỳ. Trong trường hợp đó, phần tử c_{ij} là tập các thuộc tính trong B phân biệt hai đối tượng x_i, x_j . Chẳng hạn với hệ thông tin trong *Bảng 1-7*, ma trận phân biệt xây dựng trên tập thuộc tính $\{a, b\}$ được thể hiện trong *Hình 1-3*.

Xét ma trận phân biệt được xây dựng trên tập thuộc tính $B \subseteq A$. Giả sử tập thuộc tính B phân hoạch tập đối tượng thành các lớp tương đương X_1, X_2, \dots, X_K , và do hai đối tượng thuộc một lớp tương đương thì nhận giá trị như nhau tại các thuộc tính trong B nên thay vì xây dựng ma trận phân biệt giữa từng cặp đối tượng, ta xây dựng ma trận phân biệt giữa từng cặp lớp tương đương. Khi đó, phần tử $c_{ij}, \forall i, j \in \{1, 2, \dots, K\}$ là tập hợp thuộc tính phân biệt hai đối tượng bất kỳ thuộc hai lớp tương đương X_i và X_j , hay có thể nói c_{ij} là tập các thuộc tính phân biệt

	x_1	x_2	x_3	x_4
x_1	$\{\}$	$\{b\}$	$\{a\}$	$\{\}$
x_2	$\{b\}$	$\{\}$	$\{a, b\}$	$\{b\}$
x_3	$\{a\}$	$\{a, b\}$	$\{\}$	$\{a\}$
x_4	$\{\}$	$\{b\}$	$\{a\}$	$\{\}$

Hình 1-3 : Ma trận phân biệt của hệ thông tin *Bảng 1-7* xây

dựng trên tập thuộc tính $\{a, b\}$

hai lớp tương đương X_i và X_j . Rõ ràng, ma trận phân biệt giữa từng lớp tương đương vẫn giữ nguyên giá trị về thông tin như ma trận phân biệt giữa từng cặp đối tượng, ngoài ra kích thước ma trận phân biệt đã được giảm đi đáng kể.

Ví dụ 1-13 : Với hệ thông tin trong *Bảng 1-7*, tập thuộc tính $\{a, b\}$ phân tập đối tượng thành ba lớp tương đương : $X_1 = \{x_1, x_4\}$, $X_2 = \{x_2\}$ và $X_3 = \{x_3\}$. Ma trận phân

Chương 1 – Lý thuyết Tập thô

biệt giữa các lớp tương đương xây dựng trên tập thuộc tính $\{a,b\}$ sẽ có kích thước 3×3 và được thể hiện trong Hình 1-4.

	X_1	X_2	X_3
X_1	$\{\}$	$\{b\}$	$\{a\}$
X_2	$\{b\}$	$\{\}$	$\{a,b\}$
X_3	$\{a\}$	$\{a,b\}$	$\{\}$

Hình 1- 4 : Ma trận phân biệt giữa các lớp tương đương của hệ thông tin Bảng 1-7 xây dựng trên tập thuộc tính $\{a,b\}$.

□

Cuối cùng, trong một bảng quyết định người ta còn đưa ra khái niệm *ma trận phân biệt tương đối*. Phần tử c_{ij} của ma trận này sẽ là tập \emptyset nếu hai đối tượng x_i, x_j thuộc cùng một lớp tương đương, ngược lại c_{ij} là tập thuộc tính phân biệt hai đối tượng x_i, x_j nhưng không kể thuộc tính quyết định.

Ví dụ 1-14 : Xét hệ thông tin trong Bảng 1-7 : $\mathcal{A} = (U, \{a,b\} \cup \{c\})$. Ma trận phân biệt tương đối được thể hiện trong Hình 1-5 dưới đây.

	x_1	x_2	x_3	x_4
x_1	$\{\}$	$\{b\}$	$\{a\}$	$\{\}$
x_2	$\{b\}$	$\{\}$	$\{\}$	$\{b\}$
x_3	$\{a\}$	$\{\}$	$\{\}$	$\{a\}$
x_4	$\{\}$	$\{b\}$	$\{a\}$	$\{\}$

Hình 1- 5 : Ma trận phân biệt tương đối

□

Ma trận phân biệt cho ta thông tin về các thuộc tính phân biệt hai đối tượng bất kỳ của hệ thông tin. Và do rút gọn B của một tập thuộc tính P bảo toàn khả năng phân

loại của P nên B phải có giao khác rỗng với tất cả các phần tử của ma trận phân biệt xây dựng trên P , và tập thuộc tính con nhỏ nhất của P có giao khác rỗng với mọi phần tử của ma trận phân biệt chính là rút gọn hoàn toàn của tập thuộc tính P . Từ nhận xét này ta có thể đưa ra một heuristic tìm rút gọn của tập thuộc tính P dựa vào ma trận phân biệt : đưa thuộc tính v có mặt nhiều nhất trong ma trận phân biệt vào tập rút gọn, chuyển các phần tử của ma trận phân biệt có chứa v thành \emptyset và lặp lại quá trình này cho tới khi mọi phần tử của ma trận phân biệt đều là tập rỗng. Chẳng hạn với ma trận phân biệt của *Bảng 1-7* trong *Hình 1-2*, các thuộc tính a , b và c tương ứng xuất hiện 6, 6 và 8 lần nên đầu tiên ta chọn thuộc tính c vào tập rút gọn và biến những phần tử có chứa c thành tập rỗng. Ma trận phân biệt lúc này, thể hiện ở *Hình 1-6* bên dưới, có hai thuộc tính a và b cùng xuất hiện 2 lần. Việc chọn a hoặc b vào tập rút gọn ở bước tiếp theo đều làm cho ma trận phân biệt chứa toàn các phần tử là tập rỗng. Vậy tập rút gọn là $\{a, c\}$ hoặc $\{b, c\}$.

Tất cả các rút gọn của một hệ thông tin có thể tìm được thông qua *hàm phân biệt*. Với hệ thông tin $\mathcal{A} = (U, A)$ có ma trận phân biệt $M = (c_{ij})$, hàm phân biệt f_A của \mathcal{A} được xây dựng dưới *dạng tuyến chuẩn tắc* như sau :

	x_1	x_2	x_3	x_4
x_1	$\{\}$	$\{\}$	$\{\}$	$\{\}$
x_2	$\{\}$	$\{\}$	$\{a, b\}$	$\{\}$
x_3	$\{\}$	$\{a, b\}$	$\{\}$	$\{\}$
x_4	$\{\}$	$\{\}$	$\{\}$	$\{\}$

Hình 1- 6 : Ma trận phân biệt *Hình 1-2* sau khi chọn c

vào tập rút gọn

$$f_A = \bigwedge_{i,j, i < j, c_{ij} \neq \emptyset} \{ \vee c_{ij}^* \mid c_{ij}^* \in c_{ij} \}$$

Chẳng hạn, hàm phân biệt tương ứng với ma trận *Hình 1-2* là :

Chương 1 – Lý thuyết Tập thô

$$f_A = (b \vee c) \wedge (a \vee c) \wedge (a \vee b) \wedge (b \vee c) \wedge (a \vee c)$$

Sử dụng các tính chất trong đại số Boolean như luật hút, phân phối,... ta có thể đưa hàm phân biệt về *dạng hội chuẩn tắc*, từ đó tìm được các rút gọn của hệ thông tin.

Ví dụ 1-15 : Xét hệ thông tin với tập thuộc tính $\{a, b, c, d, e\}$ và tập đối tượng $\{o_1, o_2, \dots, o_5\}$ trong *Bảng 1-8*.

Hàm phân biệt cho hệ thông tin này là :

$$f_A = (a \vee c \vee d \vee e) \wedge (a) \wedge (a \vee d \vee e) \wedge (b) \wedge (c \vee e) \wedge \\ (a \vee b \vee c \vee d \vee e) \wedge (a \vee b) \wedge (a \vee b \vee d \vee e)$$

Áp dụng luật hút :

$$x \wedge (x \vee y) = x$$

$$x \vee (x \wedge y) = x$$

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>
<i>o</i> ₁	1	0	2	1	0
<i>o</i> ₂	0	0	1	2	1
<i>o</i> ₃	2	0	2	1	0
<i>o</i> ₄	0	0	2	2	2
<i>o</i> ₅	1	1	2	1	0

Bảng 1- 8 : Một hệ thông tin

hàm phân biệt được đơn giản thành:

$$f_A = (a) \wedge (b) \wedge (c \vee e)$$

Áp dụng luật phân phối, ta được :

$$f_A = (a \wedge b \wedge c) \vee (a \wedge b \wedge e)$$

Biểu thức trên nói lên rằng : để bảo toàn khả năng phân loại của tập thuộc tính ban đầu, ta cần sử dụng tập thuộc tính $\{a, b, c\}$ hoặc $\{a, b, e\}$. Đây cũng chính là hai rút gọn hoàn toàn của hệ thông tin. □

Cuối cùng một rút gọn của hệ thông tin tìm được dựa trên ma trận phân biệt tương đối được gọi là *rút gọn tương đối* của hệ thông tin.

Một số lưu ý về hàm phân biệt :

- Các toán tử \wedge và \vee sử dụng trong hàm phân biệt không phải là các toán tử Boolean vì chúng không nhận các giá trị *true* hay *false* mà thể hiện cho ngữ nghĩa *có mặt* hay *không có mặt* của một thuộc tính nào đó. Theo đó, hàm phân biệt :

$$f_A = (a \vee b \vee c \vee f) \wedge (b \vee d) \wedge (a \vee d \vee e \vee f) \wedge (a \vee b \vee c \vee d) \wedge (b \vee d \vee e \vee f) \wedge (d \vee c)$$

được hiểu như sau : các đối tượng trong hệ thông tin có thể được phân biệt với nhau bằng cách sử dụng (thuộc tính *a* hoặc *b* hoặc *c* hoặc *f*) và (thuộc tính *b* hoặc *d*) và (thuộc tính *a* hoặc *d* hoặc *e* hoặc *f*) và (thuộc tính *a* hoặc *b* hoặc *c* hoặc *d*) và (thuộc tính *b* hoặc *d* hoặc *e* hoặc *f*) và (thuộc tính *d* hoặc *c*).

- Hàm phân biệt có thể xem như một tập các tập hợp. Ví dụ, hàm phân biệt trong lưu ý trên tương đương với tập :

$$C = \{\{a, b, c, f\}, \{b, d\}, \{a, d, e, f\}, \{a, b, c, d\}, \{b, d, e, f\}, \{d, c\}\}$$

Và cũng giống như với ma trận phân biệt, tập nhỏ nhất có giao với tất cả các phần tử của *C* chính là các rút gọn của hệ thông tin tương ứng. Ví dụ : $\{a, d\}$ là một trong các tập nhỏ nhất có giao với tất cả các phần tử của *C* nên nó là một rút gọn của hệ thông tin.

1.8. Một số thuật toán hiệu quả

Trong những phần trên, cùng với phần trình bày khái niệm chúng ta cũng đã có một số thuật toán như xác định các lớp tương đương, tìm xấp xỉ trên, xấp xỉ dưới. Phần này trình bày một số thuật toán đặc biệt hiệu quả trên các bảng dữ liệu lớn [7].

1.8.1. Lớp tương đương

Vào :

- Hệ thông tin $\mathcal{A} = (U, A)$
- Tập thuộc tính $B \subseteq A$

Ra : Tập các lớp tương đương $\{X_1^B, \dots, X_m^B\}$ của quan hệ $IND(B)$.

Thuật toán :

Bước 1 : Sắp xếp tập đối tượng trong U dựa trên một thứ tự được định nghĩa trên tập thuộc tính B , ký hiệu $<_B$:

$$x_1^1 =_B \dots =_B x_{i_1}^1 <_B x_1^2 =_B \dots =_B x_{i_2}^2 <_B \dots <_B x_1^m =_B \dots =_B x_{i_m}^m$$

trong đó $0 < m \leq n$, $0 < i_1, \dots, i_m \leq n$ và $i_1 + \dots + i_m = n$.

Bước 2 : Đặt $X_j^B = \{x_1^j, \dots, x_{i_j}^j\}$, $\forall j = 1, \dots, m$. Khi đó các tập hợp $X_1^B, X_2^B, \dots, X_m^B$ là các lớp tương đương của quan hệ $IND(B)$.

1.8.2. Xấp xỉ trên, xấp xỉ dưới

Vào :

- Hệ thông tin $\mathcal{A} = (U, A)$
- Tập thuộc tính $B \subseteq A$
- Tập đối tượng $X \subseteq U$

Ra : Tập các đối tượng $\underline{B}X$ và $\overline{B}X$.

Thuật toán :

Bước 1 : Xác định các lớp tương đương $X_1^B, X_2^B, \dots, X_m^B$ của quan hệ $IND(B)$.

Bước 2 : Đặt $\underline{B}X = \emptyset$ và $\overline{B}X = \emptyset$.

Bước 3 :

Với mọi $j = 1, 2, \dots, m$

Nếu $X_j^B \subseteq X$

Thì : $\underline{B}X = \underline{B}X \cup X_j^B$

Hết nếu

Nếu $X_j^B \cap X \neq \emptyset$

Thì : $\overline{B}X = \overline{B}X \cup X_j^B$

Hết nếu

Hết với mọi

1.8.3. Vùng dương

Vào :

- Hệ thông tin $\mathcal{A} = (U, A)$
- Tập thuộc tính $C, D \subseteq A$

Ra : Tập các đối tượng C - vùng dương của D .

Thuật toán :

Bước 1 : Xác định các lớp tương đương $X_1^C, X_2^C, \dots, X_m^C$ của quan hệ $IND(C)$.

Bước 2 : $POS_C(D) = \emptyset$.

Bước 3 :

Với mọi $j = 1, 2, \dots, m$

Nếu mọi đối tượng trong X_j^C bằng nhau tại tất cả các thuộc tính trong D

Thì : $POS_C(D) = POS_C(D) \cup X_j^C$

Hết nếu

Hết với mọi

1.8.4. Rút gọn thuộc tính

Xét hệ thông tin $\mathcal{A} = (U, A)$. Với bất kỳ $B \subseteq A$ và $X \subseteq U$, quan hệ tương đương $IND(B)$ giới hạn trên tập đối tượng X được ký hiệu là $IND^X(B)$ và tập các lớp tương đương tạo bởi quan hệ này là $[IND^X(B)]$.

Với thuộc tính $a \in A$, giả sử $[IND^X(a)] = \{X_1, X_2, \dots, X_m\}$. Đặt $x = |X|$ và $x_i = |X_i|, i = 1, \dots, m$. Số $W^X(a)$ các cặp đối tượng trong X phân biệt nhau tại thuộc tính a được tính từ công thức :

$$W^X(a) = \frac{\sum_{i \neq j} x_i x_j}{2} = \frac{x^2 - \sum_{i=1}^m x_i^2}{2}$$

Bổ đề : Giả sử $[IND^X(B)] = \{X_1, X_2, \dots, X_m\}$ và $a \in A \setminus B$. Khi đó ta có

$$1. [IND^X(B \cup \{a\})] = \bigcup_{i=1}^m [IND^{X_i}(a)]$$

2. Với $W_B^X(a)$ là số lượng cặp đối tượng trong X phân biệt nhau tại thuộc tính a nhưng bằng nhau tại các thuộc tính trong B : $W_B^X(a) = \sum_{i=1}^m W^{X_i}(a) \quad \square$

Trong phần tiếp theo chúng ta đưa ra hai chiến lược tìm tập thuộc tính rút gọn : chiến lược Johnson và chiến lược ngẫu nhiên.

1.8.4.1. Chiến lược Johnson

Vào : Hệ thông tin $\mathcal{A} = (U, A)$.

Ra : Tập thuộc tính rút gọn $R \subseteq A$.

Chiến lược :

Bước 1 : Đặt $R = \emptyset, L = \{U\}$.

Bước 2 : Thực hiện bước 3.

Bước 3 :

Với mọi $a \in A$

Với mọi $X_i \in L$

- Tìm $[IND^{X_i}(a)]$.
- Tính $W^{X_i}(a)$

Hết với mọi

Tính $W_R^U(a) = W^{X_1}(a) + \dots + W^{X_m}(a)$.

Hết với mọi

Bước 4 : Chọn thuộc tính a có giá trị $W_R^U(a)$ lớn nhất.

Bước 5 : $A = A \setminus \{a\}$, $R = R \cup \{a\}$.

Bước 6 : $L = [IND^{X_1}(a)] \cup \dots \cup [IND^{X_m}(a)]$.

Bước 7 :

Nếu $W_R^U(a) = 0$ hoặc $A = \emptyset$

Thì : Dừng

Ngược lại : Thực hiện bước 2.

1.8.4.2. Chiến lược ngẫu nhiên

Vào : Hệ thông tin $\mathcal{A} = (U, A)$.

Ra : Tập thuộc tính rút gọn $R \subseteq A$.

Chiến lược :

Bước 1 : Đặt $R = \emptyset$, $L = \{U\}$.

Bước 2 : Thực hiện bước 3.

Bước 3 :

Với mọi $a \in A$

Tính $W_R^U(a)$ như bước 3 của chiến lược Johnson.

Hết với mọi

Bước 4 : Chọn ngẫu nhiên một thuộc tính a với xác suất :

$$P(a) = \frac{W_R^U(a)}{\sum W_R^U(a_j)} \text{ với mọi } a \in A.$$

Bước 5 : $A = A \setminus \{a\}$, $R = R \cup \{a\}$.

Bước 6 : $L = [IND^{X_1}(a)] \cup \dots \cup [IND^{X_m}(a)]$.

Bước 7 :

Nếu $W_R^U(a) = 0$ hoặc $A = \emptyset$

Thì : Dừng

Ngược lại : Thực hiện bước 2.

1.8.4.3. Loại bỏ thuộc tính thừa trong một rút gọn

Rút gọn tìm được bởi hai chiến lược trên vẫn có thể chứa các thuộc tính thừa, theo nghĩa, việc loại bỏ chúng ra khỏi rút gọn không làm vùng dương của tập rút gọn ban đầu so với thuộc tính quyết định bị thay đổi. Do đó ta cần một thuật toán loại bỏ các thuộc tính dư thừa này.

Vào :

- Hệ thông tin $\mathcal{A} = (U, C \cup D)$.
- Tập rút gọn R .

Ra : Tập rút gọn $R' \subseteq R$.

Thuật toán :

Bước 1 : Tính $POS_R(D)$ và đặt $m = |POS_R(D)|$.

Bước 2 :

Với mọi $a \in A$

- Tính $POS_{R \setminus \{a\}}(D)$ và đặt $m_a = |POS_{R \setminus \{a\}}(D)|$.
- Nếu $m_a = m$

Thì : $R = R \setminus \{a\}$.

Hết nếu

Hết với mọi

Bước 3 : $R' = R$.

Chương 2

Bài Toán Nhận Dạng Mặt Người

-----oOo-----

2.1. Giới thiệu

Trong thế giới ngày nay với sự phát triển mạnh mẽ của kỹ thuật số và mạng toàn cầu, vấn đề đảm bảo an toàn về thông tin cũng như vật chất trở nên ngày càng quan trọng và khó khăn. Thỉnh thoảng chúng ta lại nghe nói đến những vụ đánh cắp thẻ tín dụng, đột nhập trái phép vào các hệ thống máy tính hay toà nhà của cơ quan nhà nước, chính phủ. Hơn 100 triệu đô la là con số đã bị thất thoát ở Mỹ vào năm 1998 do các vụ gian lận và xâm nhập nói trên (theo Reuters, 1999) [5]. Trong đa số các vụ phạm pháp này, bọn tội phạm đã lợi dụng những khe hở cơ bản trong quá trình truy cập vào các hệ thống thông tin và kiểm soát. Phần lớn những hệ thống này không thực hiện quyền truy cập của người sử dụng dựa vào thông tin “chúng ta là ai” mà chỉ dựa vào “chúng ta có gì”. Nói cách khác, thông tin mà người sử dụng cung cấp cho hệ thống không đặc trưng được cho bản thân họ, mà chỉ là những gì họ hiện đang sở hữu như sổ chứng minh nhân dân, chìa khoá, mật mã, số thẻ tín dụng hoặc họ tên. Rõ ràng những thông tin hay vật dụng này không mang tính đặc trưng mà chỉ mang tính xác thực đối với người sử dụng, và nếu chúng bị đánh cắp hay sao chép thì kẻ trộm hoàn toàn có quyền truy nhập, sử dụng dữ liệu hay phương tiện của chúng ta bất cứ lúc nào họ muốn. Hiện nay, những công nghệ hiện đại đã cho phép việc xác thực dựa vào “bản chất” của từng cá nhân. Công nghệ này dựa trên lĩnh vực được gọi là *sinh trắc học*. Kiểm soát bằng sinh trắc học là những phương pháp tự động cho phép xác thực hay nhận dạng một cá nhân dựa vào các *đặc trưng sinh lý học* của người đó như đặc điểm vân tay, gương mặt, gen,... hoặc dựa trên những đặc điểm liên quan đến *đặc trưng hành vi* như dạng chữ viết, cách gõ phím, giọng nói... Vì những hệ thống nhận dạng bằng sinh trắc học sử

dụng thông tin sinh học của con người nên kết quả chính xác và đặc biệt là rất khó bị giả mạo.

Các đặc trưng sinh lý học là duy nhất ở mỗi người và rất hiếm khi thay đổi, trong khi đó đặc trưng hành vi có thể thay đổi bất thường do các yếu tố tâm lý như căng thẳng, mệt mỏi hay bệnh tật. Chính vì lý do này, các hệ thống nhận dạng dựa trên đặc trưng sinh lý tỏ ra ổn định hơn các hệ thống dựa vào đặc trưng hành vi. Tuy nhiên, nhận dạng bằng các đặc trưng hành vi có ưu điểm là dễ sử dụng và thuận tiện hơn : thay vì phải đặt mắt trước một máy quét điện tử hay lấy ra một giọt máu, người sử dụng sẽ cảm thấy thoải mái hơn khi được yêu cầu ký tên hay nói vào một micro.

Nhận dạng gương mặt là một trong số ít các phương pháp nhận dạng dựa vào đặc trưng sinh lý cho kết quả chính xác cao đồng thời rất thuận tiện khi sử dụng. Hơn nữa, trong số các đặc trưng sinh lý học, gương mặt của mỗi người là yếu tố đầu tiên và quan trọng nhất cho việc nhận biết lẫn nhau cũng như biểu đạt cảm xúc. Khả năng nhận dạng nói chung và khả năng nhận biết gương mặt người nói riêng của con người thật đáng kinh ngạc. Chúng ta có khả năng nhận ra hàng ngàn gương mặt của những người mình đã gặp, đã giao tiếp trong cuộc sống chỉ bằng một cái nhìn thoáng qua, thậm chí sau nhiều năm không gặp cũng như những sự thay đổi trên gương mặt do tuổi tác, cảm xúc, trang phục, màu tóc,... Do đó, việc nghiên cứu các đặc tính của gương mặt người đã thu hút rất nhiều nhà triết học, nhà khoa học qua nhiều thế kỷ, trong đó có cả Aristotle và Darwin [1].

Chính vì những lý do trên, từ những năm 1970, nhận dạng mặt người đã thu hút sự quan tâm của nhiều nhà nghiên cứu trong các lĩnh vực như bảo mật, tâm lý học, xử lý ảnh và thị giác máy tính. Ngày nay các chương trình máy tính về nhận dạng mặt người đã tìm được những ứng dụng thực tế như [3] :

- *Nhận dạng tội phạm*

Các hệ thống nhận dạng mặt người đã được tích hợp vào trong các hệ thống kiểm soát sân bay và được sử dụng để tìm kiếm và nhận diện những tên khủng bố hay bọn buôn bán ma túy.

- *Kiểm soát truy cập vào các hệ thống máy tính trong môi trường cộng tác*

Việc kiểm tra đăng nhập vào các hệ thống máy PC được kết hợp giữa thông tin mật mã và / hoặc nhận dạng mặt người. Điều này giúp người làm việc không cảm thấy bị rối bời trong các thủ tục truy cập phức tạp đồng thời vẫn đảm bảo được độ tin cậy đối với thông tin khách hàng và các bí mật trong kinh doanh.

- *Giải pháp bảo mật bổ sung cho các giao dịch rút tiền tự động (ATM)*

Việc truy cập vào các máy rút tiền tự động và các dịch vụ khác của ngân hàng được kiểm soát bởi các thông tin như số tín dụng (PIN), giọng nói, tròng mắt kết hợp với nhận dạng gương mặt.

- *Đối sánh ảnh căn cước trong hoạt động của ngành luật pháp*

Các cơ quan luật pháp có thể sử dụng các hệ thống nhận dạng mặt người để đối sánh những mô tả của các nhân chứng với những tên tội phạm được lưu trữ trong cơ sở dữ liệu.

- *Ứng dụng trong các giao tiếp người – máy*

Sau khi xác định được người sử dụng và cảm xúc của họ tại thời điểm đó, các hệ thống máy tính có thể có các ứng xử thích hợp.

Trong chương này trước tiên chúng ta sẽ đi qua một số phương pháp đã được sử dụng trong lĩnh vực nhận dạng mặt người. Sau khi đưa ra một mô hình tiêu biểu cho một hệ thống nhận dạng mặt người và bàn luận về một số khó khăn cho toàn bộ quá trình nhận dạng, chúng ta sẽ tập trung vào hai giai đoạn rút trích đặc trưng và phân lớp với hai phương pháp : *phân tích thành phần chính (Principle Components Analysis – PCA)* và *mạng lượng hoá vector (Learning Vector Quantization Network – LVQ)*. Đây

là hai công cụ được sử dụng trong luận văn nhằm đánh giá hiệu suất của hệ thống nhận dạng có kết hợp với lý thuyết tập thô.

2.2. Các nghiên cứu trước đây

Rất nhiều nghiên cứu tập trung vào việc xác định các đặc điểm riêng trên gương mặt như mắt, mũi, miệng, khuôn hình của đầu, và định nghĩa một gương mặt thông qua vị trí, kích thước và mối liên hệ giữa các đặc điểm này. Những cách tiếp cận này thực sự rất khó mở rộng cho trường hợp tổng quát và khiến hệ thống dễ đổ vỡ. Ngoài ra, những nghiên cứu về cách thức con người sử dụng trong nhận dạng mặt người cho thấy những đặc trưng trên cùng những mối quan hệ trước mắt giữa chúng là chưa đủ để nhận biết gương mặt của con người. Tuy vậy, tiếp cận này vẫn còn được sử dụng rộng rãi trong lĩnh vực này [1].

Năm 1966, Bledsoe đã xây dựng hệ nhận dạng bán tự động đầu tiên có sự tương tác giữa người và máy. Đặc trưng dùng để phân lớp là các dấu hiệu cơ bản được con người thêm vào các ảnh. Các tham số sử dụng trong quá trình nhận dạng là những khoảng cách chuẩn và tỉ lệ giữa các điểm như góc của đôi mắt, góc của miệng, chóp mũi và điểm cằm [1].

Năm 1971, phòng thí nghiệm Bell đưa ra hệ nhận dạng dựa vào vector đặc trưng 21 chiều và sử dụng các kỹ thuật phân lớp mẫu để nhận dạng. Tuy nhiên, các đặc trưng này được lựa chọn một cách rất chủ quan (như màu tóc, chiều dài vành tai,...) và rất khó khăn cho quá trình tự động hoá [1].

Fischer và Elschlager năm 1973 đã cố gắng đo lường các đặc trưng tương tự nhau một cách tự động. Họ đưa ra một thuật toán tuyến tính so khớp các đặc trưng cục bộ kết hợp với các độ đo thích nghi toàn cục để tìm kiếm và định lượng các đặc trưng của gương mặt. Kỹ thuật so khớp này sau đó được tiếp tục nghiên cứu và phát triển trong các công trình của Yuille, Cohen và Hallinan năm 1988 [1].

Một số phương pháp nhận dạng liên kết (*connectionist approach*) dựa vào việc nắm bắt các cấu hình hay bản chất tựa cấu trúc của bài toán. Kohonen và Lahtio năm 1981

và 1989 đã đưa ra mạng kết hợp (*associative network*) và một thuật toán học đơn giản cho phép phân lớp một ảnh mặt cũng như gợi nhớ lại một gương mặt từ dữ liệu không hoàn chỉnh và bị nhiễu. Sử dụng cùng ý tưởng này, năm 1990, Fleming và Cottrell đã sử dụng các đơn vị phi tuyến và huấn luyện mạng bằng kỹ thuật lan truyền ngược. Hệ nhận dạng WISARD năm 1986 của Stonham đã được sử dụng thành công trong xác định mặt người cũng như nhận biết cảm xúc của họ. Hầu hết các hệ sử dụng phương pháp liên kết nói trên đều xem các ảnh mặt đầu vào như là các mẫu hai chiều tổng quát, tức là chúng không sử dụng thêm bất kỳ tri thức nào khác liên quan đến các đặc tính của các ảnh gương mặt. Ngoài ra, một số hệ thống trong số này lại cần số lượng rất lớn các mẫu dùng cho huấn luyện mới có thể đạt được hiệu quả sử dụng chấp nhận được [1].

Các phương pháp khác tiếp cận bài toán nhận dạng mặt người tự động bằng cách đặc trưng mỗi gương mặt bởi một tập các tham số hình học và thực hiện nhận dạng thông qua tập các tham số này. Hệ thống của Kanade năm 1973 có lẽ là hệ thống đầu tiên và là một trong số ít các hệ thống trong đó các bước nhận dạng được thực hiện hoàn toàn tự động, sử dụng chiến lược điều khiển từ trên xuống được định hướng bởi các đặc trưng được chọn. Hệ thống này tìm tập các tham số của gương mặt từ một ảnh đưa vào, sau đó sử dụng các kỹ thuật nhận dạng để so khớp với tập tham số của các ảnh đã biết. Đây là kỹ thuật thống kê thuần túy chủ yếu phụ thuộc vào phân tích histogram cục bộ và các giá trị độ xám tuyệt đối [1].

Năm 1991, M. Turk và A. Pentland đã sử dụng phương pháp phân tích thành phần chính trong lý thuyết thông tin để đặc trưng cho các ảnh mặt người. Ý tưởng chính của phương pháp này là tìm kiếm một không gian có số chiều nhỏ hơn, thực chất là tìm kiếm một hệ vector cơ sở sao cho hình chiếu của đám mây điểm trên chúng thể hiện rõ nét nhất hình dạng của đám mây điểm. Đám mây điểm ở đây chính là tập các vector ảnh mặt trong không gian có chiều bằng kích thước của ảnh. Mỗi ảnh mặt người sau đó

sẽ được chiếu lên không gian con này, và bộ thông số nhận được từ phép chiếu này được xem như vector đặc trưng cho từng ảnh mặt.

Năm 1998, K. Okada, J. Steffens, T. Maurer, Hai Hong, E. Elagin, H. Neven và Christoph đưa ra mô hình nhận dạng mặt người bằng sóng Gabor và phương pháp phù hợp đồ thị bó. Với ý tưởng dùng đồ thị để biểu diễn gương mặt, ảnh khuôn mặt được đánh dấu tại các vị trí đã được xác định trước trên khuôn mặt, các vị trí này được gọi là các vị trí chuẩn. Khi thực hiện so khớp đồ thị với một ảnh, các điểm chuẩn sẽ được trích ra từ ảnh và được so sánh với tất cả các điểm chuẩn tương ứng trong các đồ thị khác nhau, và đồ thị nào phù hợp nhất với ảnh sẽ được chọn [4].

Năm 1998, B. Moghaddam và A. Pentland đưa ra phương pháp phù hợp đồ thị trực tiếp từ các ảnh cần sử dụng cho mục đích nhận dạng và dùng độ đo xác suất để tính độ tương tự này [4].

Năm 1998, M. Tistaelli và E. Grosso đưa ra kỹ thuật thị giác động. Do khả năng quan sát các chuyển động của khuôn mặt và xử lý các tình huống theo dự định là thông tin rất quan trọng nên có thể sử dụng chúng để mô tả đầy đủ hơn về khuôn mặt cho mục đích thu thập mẫu và nhận dạng [4].

Năm 1998, J. Huang, C. Liu và H. Wechsler đề xuất thuật toán căn cứ trên tính tiến hoá và di truyền cho các tác vụ nhận dạng khuôn mặt. Trong cách tiếp cận này, hai mắt sẽ được dò tìm trước tiên và thông tin này được xem là vết để quan sát gương mặt, trình xử lý dò tìm mắt được tiếp tục thực hiện bằng cách sử dụng một thuật toán lai để kết hợp thao tác học và tiến hoá [4].

Năm 1998, Oi Bin Sun, Chian Prong Lam và Jian Kang Wu sử dụng phương pháp tìm vùng hai chân mày, hai mắt, mũi, miệng và cằm. Ảnh khuôn mặt thẳng ban đầu được chiếu theo chiều ngang để tìm các giá trị điểm ảnh thoả ngưỡng cho trước, đồ thị biểu diễn theo trục ngang sẽ định vị biên trên và biên dưới của hình chữ nhật bao các đặc trưng cục bộ của khuôn mặt. Tương tự với chiều đứng để tìm ra đường biên bên trái và phải cho các vùng đặc trưng [4].

Năm 1998, A. Nefian và Monson H. Hayes trình bày hướng tiếp cận theo mô hình Markov ẩn (HMM) trong đó ảnh khuôn mặt được lượng hoá thành chuỗi quan sát trên khuôn mặt theo quan niệm dựa trên thứ tự xuất hiện các đặc trưng gương mặt {hai chân mày, hai lông mi, mũi, miệng, cằm}. Trong chuỗi quan sát đó, mỗi quan sát là một vector nhiều chiều sẽ được sử dụng để đặc trưng cho mỗi trạng thái trong chuỗi trạng thái của HMM. Mỗi người được ước lượng bởi một mô hình của HMM [4].

Năm 2001, Guodong Guo, Stan Z. Li, Kap Luk Chan sử dụng phương pháp SVM để nhận dạng khuôn mặt, sử dụng chiến lược kết hợp nhiều bộ phân loại nhị phân để xây dựng bộ phân loại SVM đa lớp [4].

2.3. Mô hình nhận dạng mặt người tiêu biểu

2.3.1. Mô hình

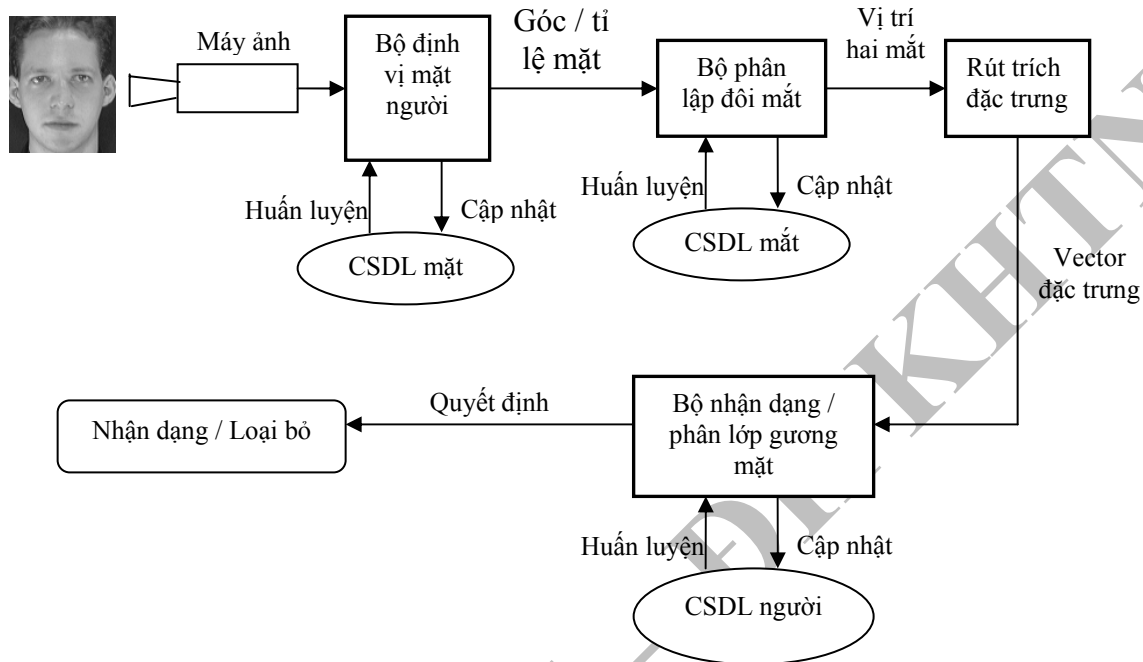
Trong đa số các trường hợp, một hệ nhận dạng mặt người bao gồm hai bộ phận sau đây [5] :

- *Bộ dò tìm hay định vị gương mặt người (Face Image Detector)* có nhiệm vụ xác định vị trí của gương mặt từ một bức ảnh bình thường.
- *Bộ phận nhận dạng hay phân lớp gương mặt (Face Recognizer)* được sử dụng để xác định người có gương mặt tương ứng là ai.

Cả hai bộ phận trên sử dụng cùng một mô hình trong hoạt động : chúng đều có một chức năng *rút trích đặc trưng (feature extractor)* nhằm biến đổi các điểm ảnh trong ảnh sang dạng biểu diễn vector có ý nghĩa, và một chức năng *nhận dạng mẫu (pattern recognizer)* có nhiệm vụ tìm kiếm một cá nhân có ảnh được lưu trong cơ sở dữ liệu trùng khớp nhất với ảnh mặt đưa vào.

Tuy nhiên, hai bộ phận trên khác nhau ở chỗ : trong bộ phận dò tìm gương mặt, chức năng nhận dạng mẫu sẽ phân lớp vector đặc trưng cần nhận dạng vào hai lớp : *lớp ảnh mặt người* và *lớp không phải ảnh mặt người*. Trong khi đó, chức năng nhận dạng mẫu của bộ phận nhận dạng / phân lớp gương mặt sẽ phân loại các vector đặc trưng (đã được cho là ảnh mặt người bởi bộ phận dò tìm gương mặt) vào lớp của các cá nhân xác

định trong cơ sở dữ liệu (chẳng hạn mặt của anh A, anh B,...). Mô hình tiêu biểu của một hệ nhận dạng mặt người được thể hiện trong *Hình 2-1* [5].



Hình 2- 1 : Mô hình nhận dạng mặt người tiêu biểu

Trong bức ảnh ban đầu có thể có rất nhiều thông tin hỗn tạp nên bộ dò tìm thường chỉ cho ra vị trí tương đối của gương mặt. Nhằm xác định chính xác vị trí của gương mặt cho quá trình nhận dạng, những người thiết kế hệ thống nhận dạng mặt người thường sử dụng vị trí của đôi mắt như một thông tin bổ sung cho quá trình định vị. Chính vì vậy mô hình đưa ra trong *Hình 2-1* có bổ sung thêm một bộ phận gọi là *bộ phân lập đôi mắt (Eye Localizer)*. Trong hệ thống này, cả ba bộ phận dò tìm gương mặt, phân lập đôi mắt và nhận dạng mặt đều dựa theo mô hình “rút trích đặc trưng + nhận dạng mẫu”.

2.3.2. Rút trích đặc trưng

Giả sử rằng mỗi bức ảnh gương mặt được thể hiện dưới dạng một ma trận số hai chiều các giá trị điểm ảnh, hay mỗi ảnh được viết dưới dạng một vector $X = \{x_i \in S\}$ với S là lưới vuông đại diện cho lưới ảnh. Đôi khi người ta thể hiện X dưới dạng

vector một chiều theo từng dòng của bức ảnh : $X = [x_1 x_2 \dots x_N]^T$ với N là tổng số điểm ảnh. Như vậy với một ảnh kích thước 320×240 , kích thước ảnh là $N = 76800$. Một vector có số chiều lớn như vậy thường không hiệu quả trong tính toán và hạn chế khả năng nhận dạng. Chính vì vậy người ta đã đưa ra nhiều phương pháp nhằm đưa vector X về một vector đặc trưng $f(X) = [f_1(X) f_2(X) \dots f_M(X)]^T$ trong đó $f_i, i = 1, 2, \dots, M$ có thể là các hàm tuyến tính hoặc phi tuyến. Trong đa số trường hợp, nhằm làm tăng hiệu quả tính toán, kích thước vector đặc trưng M thường nhỏ hơn rất nhiều kích thước của vector ảnh ban đầu N .

2.3.3. Nhận dạng mẫu

Do nhiều biến đổi tồn tại trong ảnh mặt người như góc nhìn, độ sáng ảnh hay cảm xúc thể hiện trên gương mặt,... nên các thành phần trong vector đặc trưng trong phần trên có khả năng tuân theo các biến đổi ngẫu nhiên, và do đó các vector này có thể được mô hình hoá dưới dạng các vector ngẫu nhiên. Nếu ta cho rằng xác suất một người đưa vào hệ thống nhận dạng thuộc về các lớp người trong cơ sở dữ liệu là như nhau (hay xác suất tiên nghiệm – a priori probability – của những người trong cơ sở dữ liệu là bằng nhau) thì theo lý thuyết quyết định Bayes, lỗi nhận dạng sẽ đạt giá trị cực tiểu nếu quá trình nhận dạng được thực hiện dựa trên tiêu chuẩn *maximum – likelihood* (ML). Nghĩa là, giả sử rằng người được đưa vào hệ thống có vector đặc trưng là $Y = f(X)$ và giả sử có K người trong cơ sở dữ liệu thì người này được gán vào lớp người k_o được cho bởi phương trình : $k_o = \arg \min_{1 \leq k \leq K} \{\log(p(Y|k))\}$, trong đó $p(Y|k)$ là phân bố xác suất của vector đặc trưng Y với điều kiện lớp người k .

Nếu chúng ta xem các biến đổi trong vector đặc trưng của gương mặt được tạo bởi hàm nhiễu Gaussian với giá trị trung bình zero, khi đó tiêu chuẩn ML nói trên trở thành phép đối sánh khoảng cách cực tiểu thông thường. Nghĩa là, hệ thống sẽ gán ảnh cần nhận dạng vào lớp k_o nếu khoảng cách Euclidean từ vector đặc trưng của ảnh này gần vector đặc trưng trung bình của lớp người k_o nhất so với các lớp ảnh khác trong cơ sở

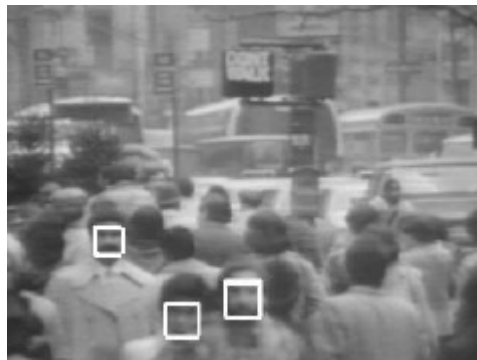
dữ liệu. Tuy vậy, các biến đổi xảy ra trong thế giới thực trên ảnh thường phức tạp hơn rất nhiều so với phân bố Gaussian nói trên [5].

2.4. Một số khó khăn trong nhận dạng mặt người

Nhận dạng mặt người là một trong những bài toán khó khăn nhất trong lĩnh vực nhận dạng ảnh. Một gương mặt người không chỉ là đối tượng ba chiều mà còn là một thực thể mang tính động rất cao. Ngoài ra, do ảnh mặt người thường được chụp trong điều kiện môi trường tự nhiên nên thông thường nền ảnh rất phức tạp và độ chiếu sáng có thể rất kém. *Hình 2-2* là một ví dụ về một bức ảnh với nền phức tạp có chứa mặt người.

Các yếu tố xuất hiện trên ảnh tạo nên khó khăn cho hệ thống nhận dạng có thể được phân thành các loại sau đây [5] :

- Máy ảnh không rõ và nhiễu
- Nền phức tạp
- Độ sáng
- Sự dịch chuyển, xoay, biến đổi tỉ lệ giữa các thành phần
- Cảm xúc thể hiện trên gương mặt
- Hoá trang, kiểu tóc



Hình 2- 2 : *Ảnh với nền phức tạp với
ba mặt người được định vị.*

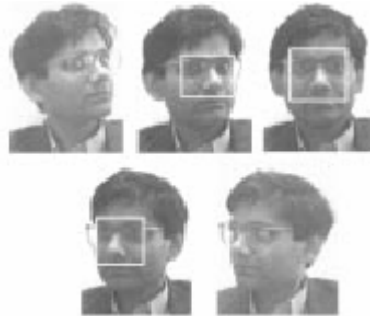
Sự không rõ của máy ảnh và nhiễu là hai hạn chế rất cơ bản trong bài toán nhận dạng. Nhiều nhà nghiên cứu đã đưa ra một số phương pháp nhằm gia tăng tỉ lệ giữa độ lớn tín hiệu so với cường độ nhiễu. Để giải quyết vấn đề *nền ảnh phức tạp*, các bộ nhận dạng hay phân lớp phải nhận được kết quả đáng tin cậy từ bộ dò tìm gương mặt, vì thế bộ phận này phải được thiết kế với độ chính xác cao. *Độ sáng* cũng là một yếu tố tác động đến kết quả nhận dạng, và để làm giảm bớt tác động của nó, người ta thường sử dụng các kỹ thuật tăng cường ảnh như threshold động, cân bằng histogram, hoặc sử dụng một mạng nơron để rút trích đặc trưng [5]. Một tiếp cận khác để giảm ảnh hưởng của độ sáng là sử dụng các mặt riêng nhận được thông qua phép phân tích thành phần chính. Chúng ta sẽ tìm hiểu phương pháp này một cách chi tiết ở phần sau.

Sự dịch chuyển, xoay hay tỉ lệ của ảnh mặt người cũng cần phải được giải quyết trong giai đoạn dò tìm gương mặt. Trong số các yếu tố này, yếu tố dịch chuyển là dễ giải quyết nhất, chẳng hạn bằng phương pháp *khoanh vùng cửa sổ (windowing)*. Vấn đề tỉ lệ sẽ được giải quyết nếu chúng ta biểu diễn mỗi ảnh dưới dạng tập các ảnh với độ phân giải khác nhau. Cuối cùng, thách thức thực sự nằm ở các ảnh mặt bị xoay theo ba trục. Rowley (1998) đã xây dựng một mạng nơron đặt trước giai đoạn dò tìm gương mặt nhằm xác định góc quay quanh trục Z của ảnh, trong đó Z là trục vuông góc với mặt phẳng ảnh. Kết xuất của mạng được sử dụng để xoay ảnh trở về vị trí cân bằng [5]. Tuy nhiên khó khăn nhất vẫn là khi ảnh bị xoay theo hai trục X và Y hoặc theo cả hai trục này. Ảnh mặt trong trường hợp này thường không thích hợp cho việc nhận dạng, vì vậy người thiết kế hệ thống thường chỉ sử dụng những bộ dò tìm gương mặt nhìn thẳng. *Hình 2-3* là ví dụ về kết quả của một bộ dò tìm thẳng.

Ảnh gương mặt với những *trạng thái cảm xúc* hay *kiểu tóc* khác nhau cũng là hai vấn đề quan trọng. Nếu đứng dưới góc độ thực thể tĩnh thì một gương mặt đang mỉm cười và một gương mặt đang nhăn nhó là hai khuôn dạng ảnh hoàn toàn khác nhau. Để khắc phục tình trạng này người ta đưa ra thuật toán *so khớp mềm dẻo (elastic matching)* trong đó sử dụng một mạng nơron có tổ chức giống như một lưới hai chiều

để mô hình hoá bề mặt của ảnh mặt thông qua quá trình huấn luyện. Nếu được huấn luyện thành công thì mạng có khả năng nâng cao chất lượng trong quá trình nhận dạng (Lades,1993) [5].

Một phương pháp khác được đưa ra để giải quyết vấn đề thay đổi cảm xúc trên



Hình 2- 3 : Kết quả của một bộ dò tìm thẳng

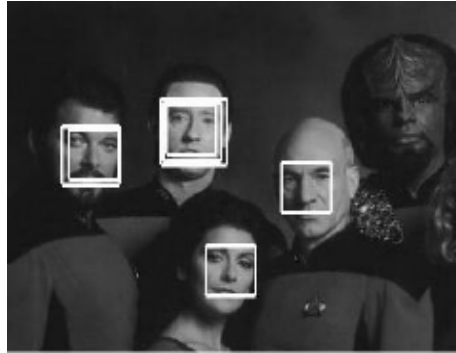
giương mặt là thay vì sử dụng toàn bộ gương mặt cho quá trình nhận dạng, người ta chỉ dùng vùng gương mặt “đáng kể nhất”. Vùng này nằm xung quanh tâm gương mặt và chỉ chứa hai mắt và lỗ mũi, loại bỏ đi miệng và hai lỗ tai. Các kết quả thực nghiệm cho thấy, cảm xúc và kiểu tóc không ảnh hưởng nhiều đến vùng mặt này, và do đó vùng mặt này vẫn có thể sử dụng được cho quá trình nhận dạng [5]. Hình 2-4 là ví dụ về vùng “đáng kể nhất” của gương mặt.



Hình 2- 4 : Vùng “đáng kể nhất” của gương mặt

Cuối cùng, việc *hoá trang* không tác động đáng kể đến quá trình dò tìm mặt, trừ trường hợp gương mặt được hoá trang quá mức như trong điện ảnh hay sân khấu. Hình 2-5 là kết quả của một bộ dò tìm được áp dụng trên ảnh có gương mặt được hoá trang. Trong hình này, gương mặt của người đóng vai quý dũ đã bị bỏ qua bởi bộ dò tìm.

Thông thường cơ sở dữ liệu của các hệ thống nhận dạng mặt người không lưu trữ ảnh mặt được hoá trang, vì vậy tất nhiên trong quá trình nhận dạng loại ảnh này cũng sẽ không được sử dụng.



Hình 2- 5 : Kết quả dò tìm trên ảnh có gương mặt được hoá trang

2.5. Phương pháp nhận dạng mặt người bằng mặt riêng

Theo mô hình đã xét trong phần 2.3, với cùng một bộ dò tìm, kết quả của các hệ thống nhận dạng phụ thuộc vào quá trình rút trích đặc trưng và phân lớp. Cho đến nay vẫn chưa có một nghiên cứu nào trả lời chính xác câu hỏi : đâu là đặc trưng thực sự để phân biệt hai gương mặt với nhau?. Thực chất của vấn đề là đi tìm một mô hình mô tả gương mặt của con người. Tuy vậy, việc mô hình hoá gương mặt một cách tổng quát là một công việc không hề đơn giản, do mặt người rất phức tạp về chi tiết, đa chiều kèm theo các yếu tố về trực giác có ý nghĩa (như cảm xúc,...) [1]. Do đó, trong phần này chúng ta nghiên cứu một mô hình của mặt người không phụ thuộc vào thông tin về chiều cũng như các đặc điểm hình học chi tiết. Từ phương pháp này chúng ta có thể xây dựng một mô hình nhận dạng mặt người nhanh, tương đối đơn giản và chính xác trong điều kiện môi trường tương đối ràng buộc, như trong văn phòng hay căn hộ gia đình. Phương pháp này dựa trên mô hình của lý thuyết thông tin, phân chia gương mặt người thành một tập nhỏ các ảnh đặc trưng gọi là các *mặt riêng*. Các mặt riêng này được xem như các thành phần chính của tập các ảnh gương mặt ban đầu. Quá trình nhận dạng được thực hiện bằng cách chiếu gương mặt mới lên không gian con được định hướng bởi các mặt riêng, sau đó so sánh nó với vị trí của các ảnh trong tập ban

đầu trong không gian mặt riêng. Phương pháp này tỏ ra vượt trội hơn các mô hình nhận dạng mặt người khác ở tốc độ, tính đơn giản, khả năng học và không nhạy cảm với những thay đổi tương đối nhỏ hay từ từ của gương mặt.

2.5.1. Mô tả phương pháp

Hầu hết các hệ thống nhận dạng tự động đã nêu đều không xét đến một câu hỏi : yếu tố nào trên ảnh mặt là quan trọng cho nhiệm vụ nhận dạng. Liên quan đến vấn đề này, lý thuyết thông tin đưa ra một phương pháp giúp mã hoá và giải mã các ảnh mặt, cho chúng ta tiếp cận những thông tin bên trong của gương mặt trong đó nhấn mạnh các đặc trưng cục bộ cũng như toàn cục có ý nghĩa. Các đặc trưng này không nhất thiết đồng nhất với những yếu tố mà chúng ta cho là đặc trưng của gương mặt như mắt, mũi, môi và tóc. Theo lý thuyết này, chúng ta cần rút trích các thông tin có liên quan trên một gương mặt người, mã hoá nó, và so sánh với từng dữ liệu của các ảnh trong tập lưu trữ được mã hoá một cách tương tự. Một phương pháp đơn giản để rút trích các thông tin có trên một ảnh mặt là định lượng các sai khác giữa các ảnh trong tập dữ liệu và sử dụng thông tin này để mã hoá cũng như so sánh các gương mặt với nhau.

Theo thuật ngữ toán học, chúng ta xem mỗi ảnh huấn luyện như là một điểm trong không gian có số chiều cực lớn. Tập các ảnh này tạo thành một phân bố tập trung trong không gian, và ta cần tìm các thành phần chính của phân bố này. Các thành phần chính này chính là các vector riêng của ma trận hiệp phương sai của tập ảnh. Các vector này, theo thứ tự tăng dần của các giá trị riêng tương ứng, là cơ sở để đánh giá độ sai khác ngày càng rõ nét giữa các ảnh trong tập huấn luyện.

Các vector riêng của ma trận hiệp phương sai nói trên được sử dụng để đặc trưng cho sự sai khác giữa các ảnh trong không gian ảnh mặt. Mỗi ảnh mặt sẽ đóng góp nhiều hay ít vào từng vector riêng này, vì vậy các vector riêng này còn được gọi là các *mặt riêng*.

Mỗi ảnh mặt người trong tập huấn luyện có thể được xây dựng lại từ tổ hợp tuyến tính của các mặt riêng. Tuy nhiên do các vector riêng xác định đường thẳng mà hình

chiều của tập ảnh huấn luyện trên đó thể hiện sự sai khác với nhau theo mức độ ngày càng giảm dần – tương ứng với sự giảm dần của các giá trị riêng – nên các ảnh cũng có thể được xấp xỉ chỉ bằng tổ hợp tuyến tính của M các vector riêng tốt nhất - tức các vector riêng tương ứng với các trị riêng lớn nhất. Các vector riêng còn lại có thể được bỏ đi mà không làm ảnh hưởng nhiều đến chất lượng nhận dạng.

Ý tưởng sử dụng các mặt riêng được xuất phát từ một kỹ thuật được phát triển bởi Sirovich và Kirby nhằm biểu diễn các ảnh gương mặt thông qua phân tích các thành phần chính. Với một tập các ảnh ban đầu, họ đi tìm một hệ trục tọa độ mới để nén các ảnh lại, trong đó mỗi trục thực sự là một ảnh được gọi là *ảnh riêng (eigenpicture)*. Họ lập luận rằng, ít nhất là về nguyên tắc, bất kỳ tập ảnh mặt nào cũng có thể được tái tạo lại một cách gần đúng bằng cách lưu trữ một tập trọng số cho từng ảnh mặt và một tập các ảnh mặt riêng chuẩn. Các trọng số của mỗi ảnh mặt có được bằng cách chiếu ảnh này lên từng ảnh mặt riêng.

Như vậy, chúng ta có thể sử dụng các trọng số trên như là đặc trưng của từng ảnh. Quá trình học sẽ tương ứng với quá trình tính toán các trọng số này, còn quá trình nhận dạng một ảnh mặt người mới là quá trình gồm 2 bước : tìm tập trọng số cho ảnh mới và so sánh với tập các trọng số được lưu trữ. Phương pháp nhận dạng mặt người bằng mặt riêng được bắt đầu từ các bước sau :

1. Thu thập tập các ảnh mặt ban đầu (dùng để huấn luyện).
2. Tìm các mặt riêng từ tập huấn luyện, giữ lại M mặt riêng tương ứng với các giá trị riêng lớn nhất. M mặt riêng này xác định một *không gian mặt*.
3. Tìm tập các trọng số đặc trưng cho từng ảnh huấn luyện bằng cách chiếu chúng lên M mặt riêng đã chọn.

Tiếp theo là các bước cần thực hiện để nhận dạng một ảnh mặt người mới :

4. Tìm tập các trọng số đặc trưng cho ảnh mặt cần nhận dạng bằng cách chiếu ảnh này lên M mặt riêng trong không gian mặt.

5. Xác định xem ảnh này có thực sự là một ảnh mặt người hay không bằng cách kiểm tra xem nó có đủ gần với không gian mặt hay không.
6. Nếu ảnh này là ảnh của một mặt người, sử dụng các thuật toán phân lớp để xác định người được lưu trước đây “gần” với người mới này nhất.

2.5.2. Vấn đề tìm các mặt riêng

Giả sử các ảnh mặt đang xét là ma trận độ sáng 8 – bits $N \times N$. Các ảnh này cũng có thể được xem như một vector N^2 chiều, như vậy một ảnh điển hình 256×256 sẽ tương ứng với một vector 65536 chiều, hay tương đương với một điểm trong không gian 65536 chiều. Theo đó, một tập ảnh huấn luyện đã được ánh xạ vào một tập điểm trong không gian có số chiều cực lớn.

Do các gương mặt có cấu trúc chung tương tự như nhau nên không phân bố một cách ngẫu nhiên mà phân bố tập trung trong không gian này, vì thế có thể biểu diễn chúng trong không gian con có số chiều nhỏ hơn. Ý tưởng chính của phân tích thành phần chính là tìm các vector “thâu tóm” một cách tốt nhất phân bố của các ảnh mặt trong không gian có số chiều cực lớn. Các vector này cho ta một không gian con mà ta sẽ gọi là *không gian mặt* có số chiều nhỏ hơn rất nhiều số chiều của không gian ban đầu. Vì các vector này là các vector riêng của ma trận hiệp phương sai tương ứng với tập ảnh mặt ban đầu, và cũng vì chúng trông giống như mặt người (tuy có phần không rõ nét) nên người ta gọi các vector này là các mặt riêng.

Giả sử tập ảnh huấn luyện là $\Gamma_1, \Gamma_2, \dots, \Gamma_M$. Ảnh mặt trung bình của tập này cho bởi biểu thức :

$$\Psi = \frac{1}{M} \sum_M \Gamma_i \quad (2-1)$$

Ví dụ về tập ảnh huấn luyện và ảnh trung bình được cho trong hình *Hình 2-6a* và *Hình 2-6b*. Độ sai khác giữa ảnh huấn luyện Γ_i so với ảnh trung bình Ψ là $\Phi_i = \Gamma_i - \Psi$.

Tập các vector ảnh huấn luyện có số chiều rất lớn này sẽ là đối tượng của phép phân tích thành phần chính. Mục đích của phép phân tích này là tìm M vector đôi một trực

Chương 2 – Bài toán Nhận dạng mặt người

giao với nhau và thể hiện rõ nét nhất phân bố của tập ảnh huấn luyện. Vector thứ k , u_k , được chọn sao cho :

$$\lambda_k = \frac{1}{M} \sum_{i=1}^M (u_k^T \Phi_i)^2 \quad (2-2)$$



Hình 2- 6 : Tập ảnh huấn luyện và ảnh trung bình

đạt giá trị cực đại, trong đó u_k là vector đơn vị trực giao với tất cả các vector đơn vị khác, hay :

$$u_l^T u_k = 0 \text{ nếu } l \neq k, \text{ và } u_k^T u_k = 1. \quad (2-3)$$

Vector u_k và vô hướng λ_k lần lượt là các vector riêng và trị riêng của ma trận hiệp phương sai :

$$C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = A A^T \quad (2-4)$$

trong đó : $A = [\Phi_1 \Phi_2 \dots \Phi_M]$. Tuy nhiên, ma trận C có kích thước N^2 nên việc xác định các giá trị riêng và vector riêng của nó là không thể thực hiện được với kích thước ảnh tương đối. Vì vậy chúng ta cần đưa ra một phương pháp tính toán khác giải quyết vấn đề này.

Nếu số lượng điểm dữ liệu trong không gian ảnh nhỏ hơn nhiều số chiều của không gian ảnh, tức $M < N^2$ thì chỉ có $M-1$ thay vì N^2 vector riêng có ý nghĩa, các vector riêng còn lại tương ứng với các giá trị riêng bằng 0. Như vậy để tìm các vector riêng N^2 chiều, đầu tiên chúng ta có thể tìm các vector riêng của ma trận M^2 sau đó thực hiện một phép biến đổi tuyến tính thích hợp để nhận được kết quả mong muốn ban đầu. Biến đổi này có được từ phân tích sau đây.

Giả sử v_i là vector riêng của ma trận $A^T A$, tức là :

$$A^T A v_i = \mu_i v_i \quad (2-5)$$

Nhân trái hai vế phương trình trên với ma trận A ta được :

$$A A^T A v_i = \mu_i A v_i \quad (2-6)$$

Phương trình này chứng tỏ $A v_i$ là vector riêng của ma trận $C = A A^T$. Theo phân tích trên, chúng ta sẽ xây dựng ma trận

$$L = A^T A \quad (2-7)$$

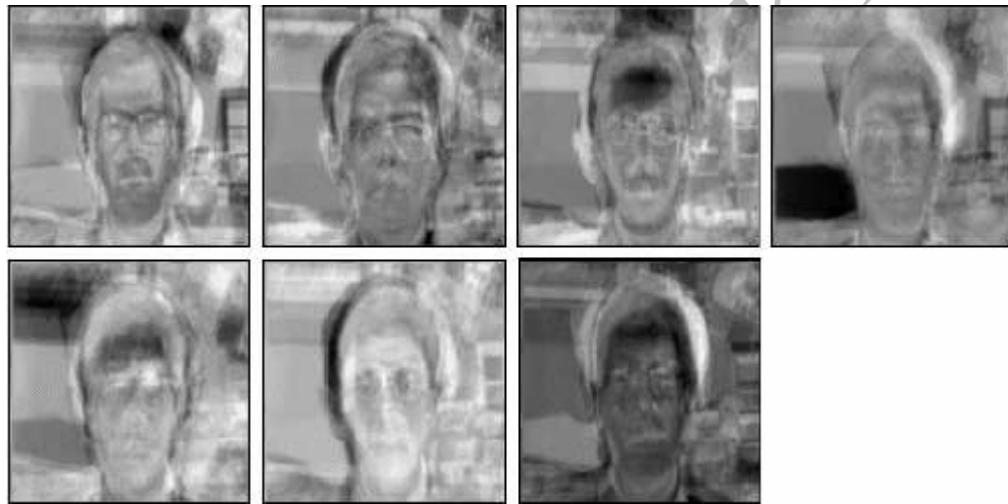
kích thước $M \times M$, trong đó các phần tử của L được xác định bởi

$$L_{m,n} = \Phi_m^T \Phi_n \quad (2-8)$$

và tìm M vector riêng v_i của nó. Các vector riêng này xác định tổ hợp tuyến tính của M ảnh mặt huấn luyện để nhận được các vector riêng u_i của ma trận C :

$$u_i = \sum_{k=1}^M v_{ik} \Phi_k, i = 1, 2, \dots, M \quad (2-9)$$

Áp dụng phân tích trên, số lượng phép tính đã giảm đi một cách đáng kể, từ bậc N^2 - số pixel trên các ảnh huấn luyện xuống còn M - số lượng ảnh trong tập huấn luyện. Trong thực tế, số lượng ảnh huấn luyện nhỏ hơn nhiều so với số lượng pixel trên mỗi ảnh mặt nên khối lượng tính toán này hoàn toàn có thể thực hiện được. Các giá trị riêng tương ứng với các vector riêng cho phép chúng ta sắp xếp các vector riêng theo thứ tự hữu dụng của chúng trong việc định lượng độ sai khác giữa các ảnh huấn luyện. Hình 2-7 là bảy mặt riêng tương ứng với bảy giá trị riêng lớn nhất của tập ảnh huấn luyện cho trong Hình 2-6.



Hình 2- 7 : Các mặt riêng tương ứng với bảy giá trị riêng lớn nhất

2.5.3. Sử dụng mặt riêng để nhận dạng

Các mặt riêng hay các vector riêng của ma trận L nói trên tạo thành cơ sở để mô tả các ảnh mặt : mỗi ảnh mặt sẽ là tổ hợp tuyến tính của các mặt riêng. Tuy nhiên, các vector này theo thứ tự phân bố giảm dần của các giá trị riêng của chúng sẽ có vai trò tương ứng giảm dần trong việc mô tả các ảnh. Do đó, trong bài toán nhận dạng mặt người, do ta không quan tâm đến nhiệm vụ khôi phục ảnh nên chúng ta có thể bỏ qua các vector riêng ứng với các giá trị riêng nhỏ. Và thực tế cho thấy chỉ một số vector riêng ứng với các giá trị riêng lớn là đủ để cho kết quả nhận dạng chấp nhận được.

Chương 2 – Bài toán Nhận dạng mặt người

Một ảnh mặt mới Γ sẽ được chiếu lên không gian mặt M' chiều, trong đó M' là số vector riêng ứng với các giá trị riêng lớn nhất được giữ lại ($M' \leq M$). Kết quả là chúng ta sẽ có một vector

$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_{M'}] \quad (2-10)$$

với thành phần ω_i được tính như sau :

$$\omega_i = u_i^T (\Gamma - \Psi), i = 1, 2, \dots, M' \quad (2-11)$$

Vector Ω sẽ được sử dụng như là vector đặc trưng của mặt Γ và sử dụng kết hợp với các phương pháp nhận dạng để tìm lớp mặt phù hợp nhất với nó. Phương pháp đơn giản nhất là tìm lớp mặt thứ k sao cho khoảng cách Euclide

$$\varepsilon_k = \|\Omega - \Omega_k\|^2 \quad (2-12)$$

đạt giá trị nhỏ nhất, trong đó Ω_k là vector mô tả hay đại diện cho lớp mặt k . Trong trường hợp dữ liệu huấn luyện ứng với lớp mặt k có nhiều ảnh thì Ω_k có thể cho là vector trung bình của các vector đặc trưng cho các ảnh của lớp k , hay cụ thể là ảnh của cùng một người mà ta đánh số thứ tự là k . Một ảnh được phân vào lớp k nếu giá trị nhỏ nhất ε_k nhỏ hơn một giá trị ngưỡng θ_k , ngược lại ảnh này được cho là *không biết*, và có thể được đưa vào một lớp mặt mới.

Một vấn đề nữa đặt ra là : do các vector đặc trưng của ảnh nhận được bằng cách chiếu ảnh lên không gian mặt nên với một ảnh bất kỳ cùng kích thước (hoặc được xử lý cho cùng kích thước để phép chiếu có nghĩa) với ảnh trong tập huấn luyện thì đều có thể tạo ra được vector đặc trưng, và như vậy có thể được phân vào một lớp mặt nào đó, thậm chí trong trường hợp ảnh này không phải là ảnh mặt người. Vấn đề này được giải quyết cũng bằng cách đưa ra giá trị ngưỡng θ . Xét ảnh *bất kỳ* Σ cùng kích thước với ảnh trong tập huấn luyện, sai khác với ảnh trung bình một lượng

$$\Pi = \Sigma - \Psi \quad (2-13)$$

Vector hình chiếu của Π lên không gian mặt cho bởi

$$\Pi_p = \sum_{i=1}^{M'} \omega_i u_i \quad (2-14)$$

trong đó ω_i là thành phần vô hướng thứ i của vector Π . Lúc này ta nói rằng ảnh Σ gần với không gian mặt nếu khoảng cách từ Σ đến không gian mặt không vượt quá ngưỡng θ , tức là :

$$\varepsilon^2 = \|\Pi - \Pi_p\|^2 \leq \theta \quad (2-15)$$

trong trường hợp ngược lại ta nói rằng Σ ở xa không gian mặt.

Như vậy với một ảnh kiểm tra bất kỳ, xét khoảng cách từ nó đến không gian mặt và đến các lớp mặt trong tập huấn luyện ta có bốn trường hợp sau đây :

1. Ảnh ở gần không gian mặt và gần một lớp ảnh.
2. Ảnh ở gần không gian mặt và ở xa tất cả các lớp ảnh.
3. Ảnh ở xa không gian mặt và ở gần một lớp ảnh.
4. Ảnh ở xa không gian mặt và ở xa tất cả các lớp ảnh.

Trong trường hợp thứ nhất, ảnh kiểm tra được nhận dạng và chỉ định vào một cho trước. Trường hợp thứ hai, ảnh kiểm tra được nhận dạng là ảnh mặt người nhưng không được chỉ định vào lớp nào trong tập huấn luyện, tức là có thể đưa vào lớp mặt người *không biết*. Hai trường hợp cuối chứng tỏ ảnh kiểm tra không phải là ảnh mặt người.

2.5.4. Tóm tắt phương pháp nhận dạng bằng mặt riêng

Chúng ta tóm tắt phương pháp nhận dạng mặt người bằng mặt riêng, trong đó sử dụng thuật toán người láng giềng gần nhất làm thuật toán phân lớp. Các bước cần tiến hành như sau :

1. Chuẩn bị tập các ảnh mặt của một số người đã biết. Mỗi người có thể có nhiều ảnh với một số biểu hiện cảm xúc, trong điều kiện chiếu sáng,...khác nhau. Ví dụ : có 10 người, mỗi người gồm 4 ảnh, ta có $M = 40$ ảnh.

2. Tính ma trận L theo (2–7), tìm các vector riêng, trị riêng của nó và chọn M' vector riêng tương ứng với các trị riêng lớn nhất. Tính M' vector riêng của ma trận C theo công thức (2–9).
3. Với mỗi lớp người thứ k trong tập ảnh huấn luyện, tính vector mẫu trung bình từ các vector đặc trưng của lớp người này. Chọn tham số θ_k cho các lớp người thứ k và tham số ngưỡng θ cho khoảng cách từ một ảnh mặt tới không gian mặt.
4. Với mỗi ảnh mới cần nhận dạng, tính vector đặc trưng Ω và khoảng cách ε_i của vector đặc trưng này đến các lớp huấn luyện và khoảng cách ε tới không gian mặt. Nếu khoảng cách nhỏ nhất ε_k thỏa : $\varepsilon_k < \theta_k$, đồng thời $\varepsilon < \theta$ thì ảnh mới này được phân vào lớp k . Nếu $\varepsilon_k > \theta_k$ và $\varepsilon < \theta$ thì ảnh mới này xem như *không biết*, và có thể được đưa vào một lớp huấn luyện mới.
5. Nếu ảnh mới được phân vào một lớp đã biết thì nó có thể được sử dụng để tính toán lại các mặt riêng. Phương pháp này làm cho hệ thống ngày càng hoàn thiện hơn.

2.6. Ứng dụng các thuật toán lượng hoá vector trong quá trình phân lớp

2.6.1. Giới thiệu

Trong bài toán nhận dạng mặt người, sau khi đã rút trích đặc trưng của từng ảnh mặt của mỗi người, chúng ta cần phải chọn một thuật toán để phân lớp các ảnh mặt mới vào một trong các lớp ảnh huấn luyện. Một tiếp cận đơn giản được nêu ra trong chương trước là thuật toán người láng giềng gần nhất : ta sẽ tìm vector đặc trưng v cho ảnh mặt người cần phân lớp, sau đó tính khoảng cách từ v đến các vector đặc trưng của từng ảnh mặt người, hoặc đến vector đặc trưng trung bình cho từng lớp người trong tập

huấn luyện. Lớp có vector đặc trưng (trung bình) gần v nhất sẽ được gán cho ảnh cần nhận dạng.

Tuy nhiên, phương pháp trên có hai nhược điểm sau đây :

- Tổng số lượng vector của tập huấn luyện có thể rất lớn, khi đó để phân lớp một vector v mới ta có thể phải duyệt hết tất cả các vector này để tìm vector gần v nhất.
- Không tận dụng một đặc điểm là các vector thuộc cùng một lớp thường có xu hướng tập trung quanh một hoặc nhiều vị trí mà ta có thể xem như trọng tâm của lớp. Do đó một vector u thuộc lớp \mathcal{S}_i nằm *lạc lõng* - tức không nằm tập trung quanh một trọng tâm của lớp, hoặc do tập huấn luyện chưa đủ lớn để ta có thể *nhận thức* được tâm của lớp \mathcal{S}_i tương ứng với u - có thể làm cho kết quả phân lớp vector v không chính xác.

Để khắc phục các nhược điểm trên, người ta sử dụng *thuật toán lượng hoá vector* (*Learning Vector Quantization - LVQ*) để tìm kiếm các vector thể hiện các trọng tâm của cluster hay lớp dữ liệu. Mỗi cluster dữ liệu trong bài toán nhận dạng mặt người là tập các điểm tương ứng với vector đặc trưng của ảnh thuộc cùng một người trong tập huấn luyện. Các vector trọng tâm này được gọi là các *vector tham chiếu* (*codebook vector*). Sau khi đã tìm được các vector tham chiếu của mỗi lớp, chúng ta sẽ sử dụng thuật toán người láng giềng gần nhất để phân loại các ảnh mới đưa vào hệ thống.

2.6.2. Một số thuật toán lượng hoá vector

Trong phần này chúng ta nghiên cứu hai thuật toán lượng hoá vector : *thuật toán LVQ1* và cải tiến của nó, *thuật toán LVQ1 với tốc độ học tối ưu* (*Optimized – learning – rate LVQ1*, hay *OLVQ1*).

2.6.2.1. Thuật toán LVQ1

Giả sử trong không gian các vector đặc trưng ban đầu chúng ta đặt một số vector tham chiếu m_i , mỗi vector này được “gán” vào một lớp trong tập huấn luyện. Có thể có

nhiều vector tham chiếu cùng thuộc vào một lớp. Các vector tham chiếu này sẽ được điều chỉnh dần trong quá trình học để có thể hội tụ về các điểm mà ta có thể xem như trọng tâm của cluster tương ứng.

Đặt

$$c = \arg \min_i \{\|x - m_i\|\} \quad (2-16)$$

là chỉ số vector tham chiếu gần với vector dữ liệu x nhất. Hiển nhiên chúng ta muốn rằng tất cả các vector m_c nói trên sẽ thuộc vào cùng một lớp với vector x tương ứng của nó. Quá trình học sau đây sẽ làm cực tiểu một cách tương đối sai số này.

Gọi $x(t)$ là vector dữ liệu và $m_i(t)$ là vector tham chiếu m_i tại thời điểm học t . Với các giá trị được khởi tạo thích hợp, quá trình cập nhật các vector tham chiếu được thể hiện như sau :

$$m_c(t+1) = m_c(t) + \alpha(t)[x(t) - m_c(t)] \quad (2-17)$$

nếu x và m_c thuộc vào cùng một lớp.

$$m_c(t+1) = m_c(t) - \alpha(t)[x(t) - m_c(t)] \quad (2-18)$$

nếu x và m_c thuộc vào hai lớp khác nhau.

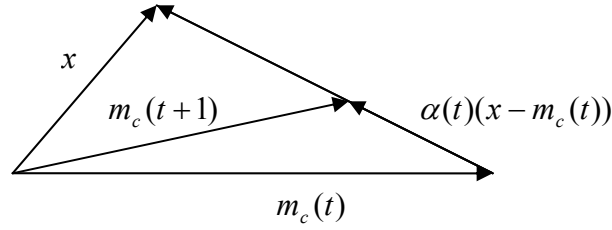
$$m_i(t+1) = m_i(t) \text{ với mọi } i \neq c \quad (2-19)$$

trong đó $\alpha(t) \in (0,1)$ là tốc độ học. Giá trị này có thể là hằng số hoặc giảm tuyến tính theo thời gian, chẳng hạn như :

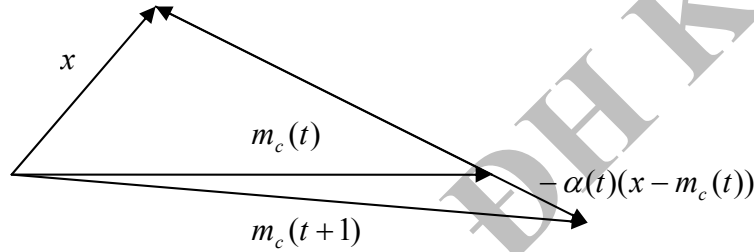
$$\alpha(t) = 0.1(1 - \frac{t}{N}) \quad (2-20)$$

với N là số chu kỳ của quá trình học.

Theo thuật toán trên, tại mỗi thời điểm học t , ta xét mỗi vector dữ liệu x và tìm vector tham chiếu m_c tương ứng với x . Nếu hai vector này thuộc cùng một lớp thì vector tham chiếu sẽ được di chuyển vào gần vector dữ liệu hơn nữa, ngược lại nó sẽ bị “đẩy” ra xa vector dữ liệu hơn. Hình 2-8 và Hình 2-9 dưới đây minh họa cho hai trường hợp này.



Hình 2- 8 : Vector tham chiếu được di chuyển gần với vector dữ liệu hơn – trường hợp hai vector này cùng lớp



Hình 2- 9 : Vector tham chiếu được đẩy ra xa vector dữ liệu hơn - trường hợp hai vector này khác lớp

2.6.2.2. Thuật toán OLVQ1

Thuật toán này cũng dựa trên ý tưởng của LVQ1, chỉ khác là mỗi vector tham chiếu sẽ có một giá trị tốc độ học riêng. Như vậy, quá trình học được thể hiện bởi các biểu thức sau :

$$m_c(t+1) = m_c(t) + \alpha_c(t)[x(t) - m_c(t)] \quad (2-21)$$

nếu x và m_c thuộc vào cùng một lớp.

$$m_c(t+1) = m_c(t) - \alpha_c(t)[x(t) - m_c(t)] \quad (2-22)$$

nếu x và m_c thuộc vào hai lớp khác nhau.

$$m_i(t+1) = m_i(t) \text{ với mọi } i \neq c \quad (2-23)$$

Phương trình (2-21) và (2-22) có thể được viết lại như sau :

$$m_c(t+1) = [1 - s(t)\alpha_c(t)]m_c(t) + s(t)\alpha_c(t)x(t) \quad (2-24)$$

trong đó $s(t) = 1$ nếu $m_c(t)$ và x cùng lớp, ngược lại $s(t) = -1$.

Vì (2-24) đúng với mọi thời điểm huấn luyện t nên ta có thể khai triển về phải của nó thêm một cấp nữa, khi đó vector tham chiếu tại thời điểm $t+1$ sẽ phụ thuộc vào $\alpha_c(t)$ phần vector dữ liệu tại thời điểm t và $[1-s(t)\alpha_c(t)]\alpha_c(t-1)$ phần vector dữ liệu tại thời điểm $t-1$. Ở đây, nếu ta cho rằng các vector dữ liệu đóng góp một lượng như nhau vào quá trình điều chỉnh vector tham chiếu tại mọi thời điểm thì ta sẽ có biểu thức :

$$\alpha_c(t) = [1-s(t)\alpha_c(t)]\alpha_c(t-1), t = 1, 2, \dots, N \quad (2-25)$$

Biểu thức trên cho ta phương pháp cập nhật tốc độ học tương ứng với vector tham chiếu m_c :

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1+s(t)\alpha_c(t-1)} \quad (2-26)$$

Rõ ràng là với phương pháp cập nhật trên thì giá trị của tham số học α_c sẽ giảm nhanh với $s(t) = 1$, tức là khi vector tham chiếu m_c đã rơi đúng vào vùng cluster của nó. Tuy vậy, do giá trị α_c cũng có thể tăng lên, khi $s(t) = -1$, nên chúng ta phải chặn giá trị học này tại một ngưỡng nào đó, chẳng hạn như giá trị ngưỡng bằng 1, hoặc ta có thể khởi tạo giá trị học cao một chút, ví dụ bằng 0.3, và sau đó không chế tốc độ học không được vượt quá giá trị này.

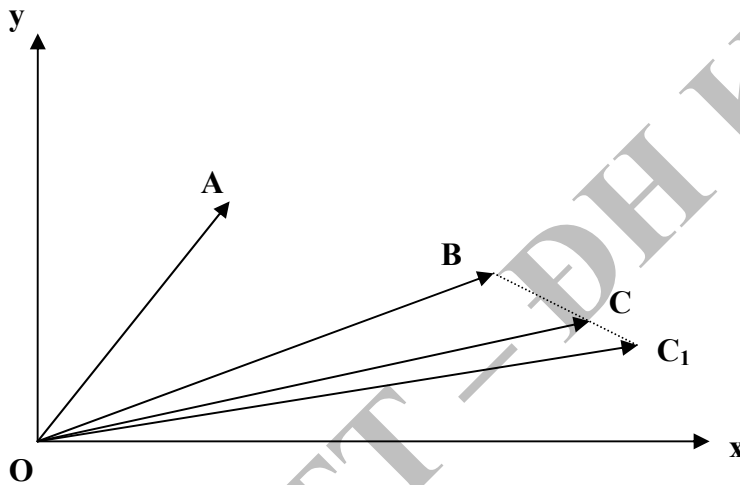
Như vậy, với một chút giả thiết tự đưa ra cho quá trình cập nhật tốc độ học, chúng ta đã tìm được một phương pháp giúp cho thuật toán hội tụ rất nhanh. Thực nghiệm chứng tỏ rằng, thuật toán thường hội tụ sau khoảng thời gian $N \approx k|M|$, với k nhận giá trị từ 30–50 và $|M|$ là tổng số vector tham chiếu được sử dụng cho tất cả các lớp dữ liệu.

2.6.3. Vấn đề khởi tạo vector tham chiếu

Chương 2 – Bài toán Nhận dạng mặt người

Trong các thuật toán học LVQ, vấn đề khởi tạo các vector tham chiếu có vai trò rất quan trọng. Hình 2-9 dưới đây là một ví dụ.

Trong ví dụ này ta có hai vector dữ liệu : vector \vec{OA} thuộc lớp 1 và vector \vec{OB} thuộc lớp 2. Gọi \vec{OC} là vector tham chiếu thuộc lớp 1 và được khởi tạo gần \vec{OB} hơn \vec{OA} như hình vẽ. Sau bước cập nhật đầu tiên, vector \vec{OC} trở thành vector $\vec{OC_1}$.



Hình 2- 10 : Vector tham chiếu \vec{OC} khởi tạo không tốt nên sau khi cập nhật thành $\vec{OC_1}$ thì càng xa vector dữ liệu \vec{OA} hơn.

Ta nhận thấy rằng, vector $\vec{OC_1}$ lại càng xa vector \vec{OA} hơn nữa, và thực sự là nếu tiếp tục cập nhật thì vector tham chiếu này sẽ đi ra xa tới vô cùng.

Ví dụ trên cho thấy tầm quan trọng của vấn đề khởi tạo các vector tham chiếu. Trong chương trình cài đặt cho luận văn này, chúng ta sẽ khởi tạo các vector tham chiếu như sau :

- Sử dụng thuật toán người láng giềng gần nhất để tự phân lớp các vector dữ liệu.
- Với mỗi lớp k , nếu các vector dữ liệu thuộc lớp này và, qua quá trình phân lớp trên, lại được phân lớp chính xác vào lớp k của chúng thì ta sẽ khởi tạo

các vector tham chiếu của lớp k trùng với các vector dữ liệu này. Bằng cách này, qua lượt đầu tiên cập nhật tất cả các vector tham chiếu, chúng ta luôn đảm bảo rằng các vector tham chiếu sẽ được di chuyển gần về một vector dữ liệu thuộc cùng lớp với nó.

Kết quả kiểm nghiệm cho thấy sau một thời gian huấn luyện, các vector tham chiếu luôn hội tụ hay chỉ dao động xung quanh một vị trí được coi là trọng tâm của lớp.

KHOA CNTT – ĐH KHNTN

Chương 3

Ứng Dụng Tập Thô Vào Bài Toán Nhận Dạng Mặt Người

-----oOo-----

3.1. Giới thiệu

Như chúng ta đã biết, rút trích đặc trưng là một giai đoạn quan trọng trong toàn bộ quy trình hoạt động của một hệ thống nhận dạng. Các đặc trưng của mỗi mẫu sau khi rút trích từ tập dữ liệu ban đầu sẽ được thể hiện dưới dạng các vector nhiều chiều. Tập các vector này sẽ được sử dụng cho quá trình huấn luyện hay phân lớp của bộ nhận dạng. Có hai vấn đề chúng ta cần quan tâm :

- Kích thước vector đặc trưng ảnh hưởng trực tiếp đến tốc độ huấn luyện và phân lớp. Những trường hợp sử dụng mạng nơron lan truyền ngược để huấn luyện các mẫu học với thời gian thực hiện hàng giờ, thậm chí nhiều ngày không phải là hiếm.
- Trong các thành phần của vector đặc trưng, không phải tất cả các thành phần đều có ích cho quá trình phân lớp. Điều đó có nghĩa là có khả năng chúng ta giảm được kích thước của vector đặc trưng trước khi đưa vào sử dụng cho quá trình huấn luyện hay phân lớp.

Từ hai vấn đề đặt ra ở trên, người ta đã quan tâm đến việc lựa chọn và rút gọn vector đặc trưng cho bài toán nhận dạng nói chung và nhận dạng mặt người nói riêng. Trong chương này chúng ta sẽ nghiên cứu một số phương pháp cũng như đưa ra mô hình thử nghiệm cho vấn đề ứng dụng lý thuyết tập thô vào giai đoạn lựa chọn và rút gọn đặc trưng cho bài toán nhận dạng mặt người.

3.2. Ứng dụng tập thô trong lựa chọn đặc trưng [1]

3.2.1. Phương pháp chung

Giả sử $\mathcal{A} = (U, C \cup D)$ là bảng quyết định tạo bởi các vector đặc trưng của tập dữ liệu mẫu. Các thuộc tính trong C tương ứng là các thành phần đặc trưng của mỗi vector mẫu, tập thuộc tính D có duy nhất một phần tử là thuộc tính thể hiện lớp của các vector mẫu.

Khả năng phân loại của tập thuộc tính quyết định C đối với tập các lớp đối tượng được thể hiện bởi tập đối tượng $POS_C(D)$. Nhiệm vụ chính của các phương pháp lựa chọn đặc trưng là tìm một tập thuộc tính rút gọn $R \subseteq C$ bảo toàn khả năng phân loại này. Trong phần lý thuyết tập thô, chúng ta biết rằng trong một hệ thống tin có thể tồn tại một số thuộc tính C -không thể bỏ được mà việc loại bỏ chúng sẽ ngay lập tức làm giảm khả năng phân loại của tập thuộc tính C ban đầu, tập thuộc tính đó chính là lõi $CORE_D(C)$. Do đó, tập thuộc tính lõi phải tồn tại trong mọi rút gọn cũng như rút gọn hoàn toàn của hệ thống tin.

Như vậy, bài toán lựa chọn tập thuộc tính dựa trên lý thuyết tập thô trở thành bài toán lựa chọn các thuộc tính C -có thể bỏ được để bổ sung vào tập $CORE_D(C)$ cho đến khi tập thuộc tính nhận được R trở thành rút gọn của tập thuộc tính C ban đầu, tức là điều kiện sau phải được thỏa mãn : $POS_R(D) = POS_C(D)$.

3.2.2. Kết hợp heuristic và lý thuyết tập thô

3.2.2.1. Mô tả heuristic

Trong phần này chúng ta sẽ đưa ra một heuristic cho việc lựa chọn dần các thuộc tính C -có thể bỏ được cho đến khi nhận được tập rút gọn R . Tiêu chuẩn lựa chọn các thuộc tính này là một biến thể của tiêu chuẩn đã được sử dụng trong hệ thống khám phá luật $GDT - RS$. Trong hệ thống này, các thuộc tính được chọn để phát sinh ra các luật tuân theo chiến lược được nêu sau đây :

1. Để nhận được tập thuộc tính nhỏ nhất có thể, chúng ta ưu tiên chọn thuộc tính a_0 mà việc thêm nó vào tập rút gọn R hiện có sẽ làm cho số lượng đối tượng bền vững tăng lên nhanh nhất, tức là :

$$a_0 = \arg \max_{a \in C \setminus R} POS_{R \cup \{a\}}(D)$$

2. Khi thêm thuộc tính a_0 vào R , tập các phân hoạch các đối tượng bền vững theo tập thuộc tính được chọn, tức tập hợp $POS_{R \cup \{a_0\}}(D) | IND(R \cup \{a_0\} \cup D)$, sẽ thay đổi, từ đó làm thay đổi tập các luật phát sinh. Trong các lớp tương đương thuộc tập phân hoạch mới, giả sử M là lớp có nhiều phần tử nhất và r là luật được phát sinh tương ứng với tập các đối tượng M . Ta nhận xét rằng, kích thước của tập M càng lớn bao nhiêu thì tính bao phủ của luật r càng lớn bấy nhiêu, cụ thể hơn là số lượng đối tượng thỏa mãn r càng lớn. Như vậy ta có thể lấy kích thước của M như là tiêu chuẩn thứ hai trong lựa chọn thuộc tính.

Tóm lại : Ta sử dụng hai chỉ số sau :

- Số lượng đối tượng bền vững :

$$v_a = card(POS_{R \cup \{a\}}(D))$$

- Kích thước lớp tương đương lớn nhất :

$$m_a = \max_size(POS_{R \cup \{a\}}(D) | IND(R \cup \{a\} \cup D))$$

trong đó a là thuộc tính chưa được chọn : $a \in C \setminus R$.

Hai chỉ số trên có xu hướng cạnh tranh với nhau, do đó ta sử dụng tích của chúng làm tiêu chuẩn cuối cùng để chọn thuộc tính.

3.2.2.2. Thuật toán

Trong phần này chúng ta trình bày thuật toán lựa chọn thuộc tính dựa vào tiêu chuẩn đánh giá được nêu ở phần trên. Chúng ta sử dụng tập thuộc tính lõi như là tập xuất phát, kể đến ta chọn dần các thuộc tính còn lại cho đến khi nhận được một rút gọn. Chiến lược tìm kiếm sử dụng trong thuật toán này là tìm kiếm tham lam, điều này

Chương 3 – Ứng dụng Tập thô vào bài toán nhận dạng mặt người

không đảm bảo tìm được tập rút gọn hoàn toàn, nhưng cho phép thuật toán hoạt động hiệu quả trên tập dữ liệu lớn với nhiều thuộc tính.

Vào :

- Hệ quyết định $\mathcal{A} = (U, C \cup D)$.
- Ngưỡng *threshold*.

Ra :

- Tập thuộc tính rút gọn R .

Cấu trúc dữ liệu :

- P : Tập các thuộc tính chưa được chọn.
- k : Tỷ lệ đối tượng bền vững tại bước hiện tại.

Thuật toán :

Bước 1 : Khởi tạo : $R = CORE_D(C)$, $P = C \setminus CORE_D(C)$, $k = 0$.

Bước 2 : Loại bỏ các đối tượng bền vững : $U = U \setminus POS_R(D)$

Bước 3 : Đặt $k = \frac{card(POS_R(D))}{card(U)}$.

Nếu $k \geq threshold$ hoặc $POS_R(D) = POS_C(D)$

Thì : Dừng.

Hết nếu

Bước 4 : Với mọi $a \in P$

$$v_a = card(POS_{R \cup \{a\}}(D))$$

$$m_a = \max_size(POS_{R \cup \{a\}}(D) \mid IND(R \cup \{a\} \cup D))$$

Hết với mọi

Bước 5 : Đặt $a_0 = \arg \max_{a \in P} (v_a \times m_a)$

Bước 6 : $R = R \cup \{a_0\}$, $P = P \setminus \{a_0\}$

Bước 7 : Thực hiện bước 2.

3.2.2.3. Ví dụ minh họa

Chương 3 – Ứng dụng Tập thô vào bài toán nhận dạng mặt người

Trong phần này chúng ta sẽ minh hoạ thuật toán trên bằng một ví dụ cụ thể. Bảng quyết định được cho trong *Bảng 3-1*, trong đó $U = \{x_1, x_2, \dots, x_7\}$, $C = \{a, b, c, d\}$ và $D = \{E\}$. Giả sử giá trị ngưỡng *threshold* được chọn là 1.0.

	a	b	c	d	E
x_1	1	0	2	1	1
x_2	1	0	2	0	1
x_3	1	2	0	0	2
x_4	1	2	2	1	0
x_5	2	1	0	0	2
x_6	2	1	1	0	2
x_7	2	1	2	1	1

Bảng 3- 1 : *Bảng quyết định cho ví dụ minh hoạ*

Ma trận phân biệt tương đối của hệ thống tin trên được thể hiện trong *Hình 3-1*.

Từ ma trận này ta nhận được tập lõi $CORE_D(C) = \{b\}$.

Từ các lớp tương đương :

$$U \mid IND(\{b\}) = \{\{x_1, x_2\}, \{x_5, x_6, x_7\}, \{x_3, x_4\}\}$$

$$U \mid IND(\{E\}) = \{\{x_4\}, \{x_1, x_2, x_7\}, \{x_3, x_5, x_6\}\}$$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	$\{\}$	$\{\}$	$\{b, c, d\}$	$\{b\}$	$\{a, b, c, d\}$	$\{a, b, c, d\}$	$\{\}$
x_2		$\{\}$	$\{b, c\}$	$\{b, d\}$	$\{a, b, c\}$	$\{a, b, c\}$	$\{\}$
x_3			$\{\}$	$\{c, d\}$	$\{\}$	$\{\}$	$\{a, b, c, d\}$
x_4				$\{\}$	$\{a, b, c, d\}$	$\{a, b, c, d\}$	$\{a, b\}$

Chương 3 – Ứng dụng Tập thô vào bài toán nhận dạng mặt người

x_5					$\{\}$	$\{\}$	$\{c, d\}$
x_6						$\{\}$	$\{c, d\}$
x_7							$\{\}$

Hình 3- 1 : Ma trận phân biệt tương đối của hệ thống tin trong Bảng 3-1

ta nhận được b -vùng dương của E : $POS_{\{b\}}(\{E\}) = \{x_1, x_2\}$. Như vậy ở trạng thái khởi đầu, ta có : $R = \{b\}$, $P = \{a, c, d\}$ và tập các trạng thái không bền vững $U = \{x_3, x_4, x_5, x_6, x_7\}$. Trạng thái ban đầu này được cho trong Bảng 3-2.

	b	E
x_3	2	2
x_4	2	0
x_5	1	2
x_6	1	2
x_7	1	1

Bảng 3- 2 : Trạng thái ban đầu

Vì tỉ lệ đối tượng bền vững là $k = 2 / 7 < threshold = 1$ nên R chưa phải là rút gọn. Chúng ta phải tiếp tục việc lựa chọn thuộc tính để đưa vào R . Các thuộc tính ứng viên còn lại của chúng ta là a , c và d . Bảng 3-3, 3-4 và 3-5 dưới đây lần lượt là kết quả của việc chọn các thuộc tính a , c và d .

	a	b	E
x_3	1	2	2
x_4	1	2	0
x_5	2	1	2
x_6	2	1	2

x_7	2	1	1
-------	---	---	---

Bảng 3- 3 : *Trạng thái tiếp theo khi thêm a*

	b	c	E
x_3	2	0	2
x_4	2	2	0
x_5	1	0	2
x_6	1	1	2
x_7	1	2	1

Bảng 3- 4 : *Trạng thái tiếp theo khi thêm c*

	b	d	E
x_3	2	0	2
x_4	2	1	0
x_5	1	0	2
x_6	1	0	2
x_7	1	1	1

Bảng 3- 5 : *Trạng thái tiếp theo khi thêm d*

Từ ba bảng trên ta nhận được các tập hợp sau đây :

$$U \mid IND(\{E\}) = \{\{x_3, x_5, x_6\}, \{x_4\}, \{x_7\}\}$$

$$U \mid IND(\{a, b\}) = \{\{x_3, x_4\}, \{x_5, x_6, x_7\}\}$$

$$U \mid IND(\{b, c\}) = \{\{x_3\}, \{x_4\}, \{x_5\}, \{x_6\}, \{x_7\}\}$$

$$U \mid IND(\{b, d\}) = \{\{x_3\}, \{x_4\}, \{x_5, x_6\}, \{x_7\}\}$$

$$POS_{\{a,b\}}(E) = \emptyset$$

$$POS_{\{b,c\}}(E) = POS_{\{b,d\}}(E) = \{x_3, x_4, x_5, x_6, x_7\}$$

$$\max_size(POS_{\{b,c\}}(E) | IND(\{b,c,E\})) = 1$$

$$\max_size(POS_{\{b,d\}}(E) | IND(\{b,d,E\})) = card(\{x_5, x_6\}) = 2$$

Ta nhận thấy rằng thuộc tính a không làm giảm số lượng thuộc tính không bền vững, trong khi đó việc chọn c hoặc d đều làm cho tất cả các thuộc tính còn lại trở nên bền vững. Theo thuật toán của chúng ta, thuộc tính d sẽ được chọn trước tiên.

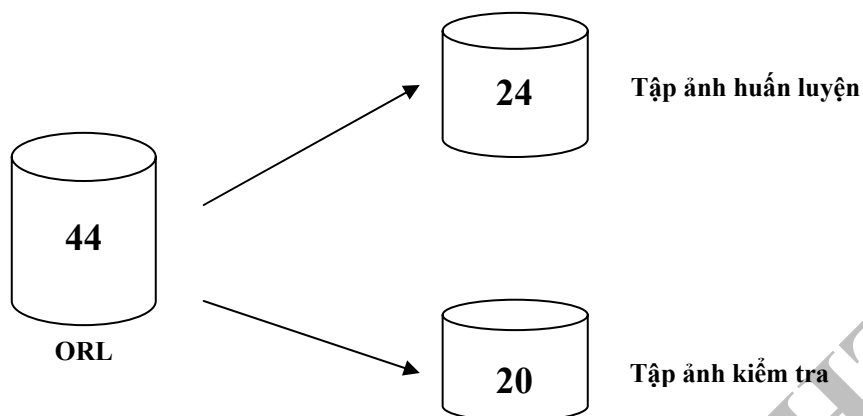
Sau khi đã đưa d vào tập R , tất cả các đối tượng đều bền vững, $k = threshold$ nên thuật toán kết thúc. Vậy tập rút gọn nhận được là $\{b, d\}$. \square

3.3. Mô hình thử nghiệm

Phần này trình bày về mô hình được sử dụng để thử nghiệm tác dụng của tập thô trong lựa chọn đặc trưng cho bài toán nhận dạng mặt người. So với mô hình tổng quát của một hệ nhận dạng mặt người hoàn chỉnh nêu ra trong chương II, ở đây chúng ta chỉ cài đặt chức năng của một bộ nhận dạng với hai giai đoạn huấn luyện và kiểm tra.

3.3.1. Tập dữ liệu

Trong toàn bộ quá trình thử nghiệm, chúng ta sử dụng tập dữ liệu mặt người ORL được cung cấp miễn phí trên mạng Internet tại địa chỉ <http://www.uk.research.att.com/facedatabase.html>. Tập dữ liệu này có tất cả 40 người, mỗi người có 10 ảnh kích thước $(ngang, doc) = (92, 112)$ với nhiều trạng thái cảm xúc, mang kính / không mang kính, ... Nền ảnh là màu đen đồng nhất. Sử dụng tập dữ liệu này là phù hợp với mục tiêu đề ra : kiểm nghiệm tác dụng của tập thô vào quá trình lựa chọn đặc trưng. Để làm cho tập dữ liệu phong phú hơn, các ảnh mặt sẽ được biến đổi bởi các thao tác thêm nhiễu, co giãn, làm mờ, ... Tập dữ liệu dùng huấn luyện sẽ có 24 ảnh / người, kiểm tra có 20 ảnh / người.



Hình 3- 2 : Phân chia tập dữ liệu huấn luyện và kiểm tra



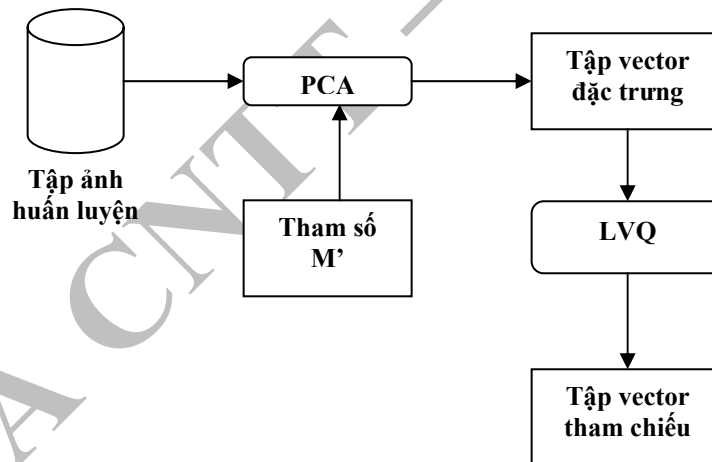
Hình 3- 3 : Ảnh của 10 người đầu tiên trong tập dữ liệu ORL

3.3.2. Mô hình 1

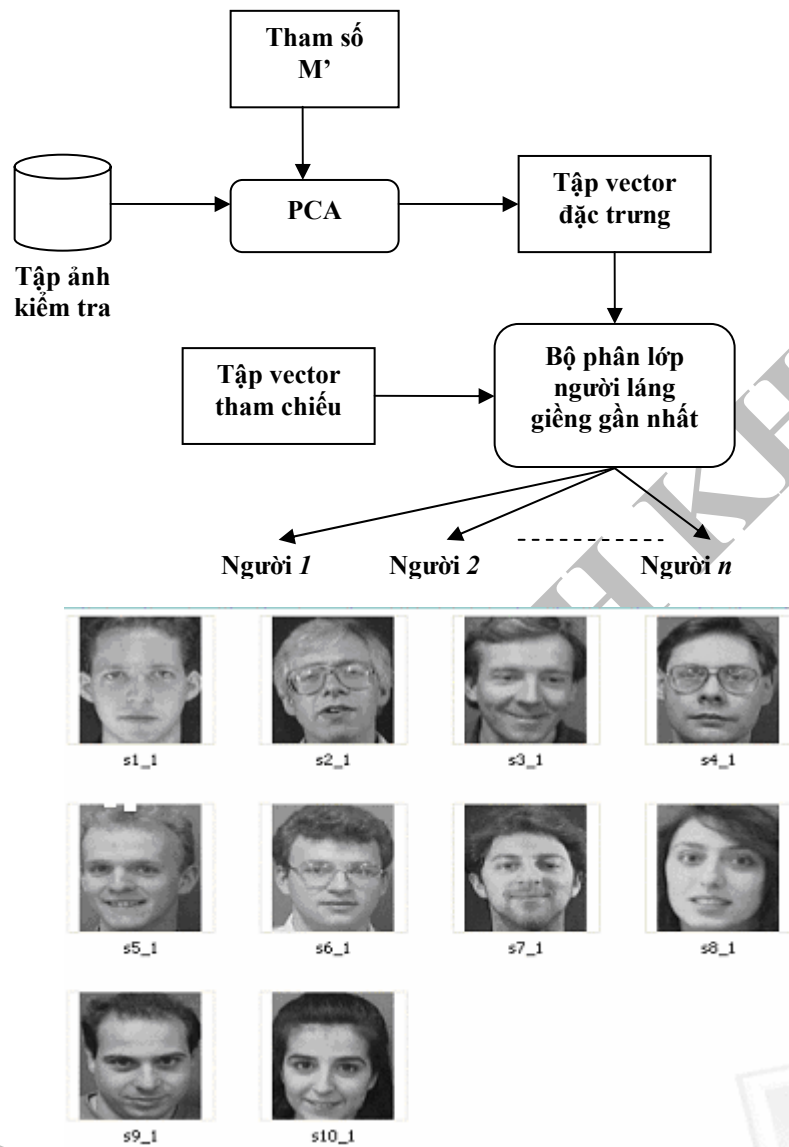
Đây là mô hình dùng kiểm tra khả năng của bộ rút trích đặc trưng bằng phương pháp phân tích thành phần chính *PCA* và khả năng phân loại bằng thuật toán lượng hoá vector *LVQ*.

- Giai đoạn huấn luyện : Tập ảnh huấn luyện được đưa vào bộ phận phân tích thành phần chính, kết quả tạo ra là tập các vector đặc trưng. Tập vector đặc trưng này làm đầu vào cho thuật toán LVQ để tạo ra tập vector tham chiếu. Tập vector tham chiếu được lưu trữ lại cho quá trình phân lớp.
- Giai đoạn phân lớp : Rút trích đặc trưng của tập ảnh kiểm tra bằng bộ phận phân tích PCA. Tập vector đặc trưng này cùng tập vector tham chiếu được đưa vào bộ phận lớp bằng thuật toán người láng giềng gần nhất để thực hiện phân loại.

Hai giai đoạn huấn luyện và phân lớp lần lượt được thể hiện trong Hình 3-4 và Hình 3-5. Lưu ý rằng trong các mô hình này, tham số M' là số thành phần tốt nhất, tương ứng với các trị riêng lớn nhất, trong phân tích thành phần chính. Giá trị tham số này do người dùng cung cấp.



Hình 3- 4 : Giai đoạn huấn luyện tạo tập vector tham chiếu



Hình 3- 5 : Giai đoạn phân lớp tập ảnh kiểm tra

3.3.3. Mô hình 2

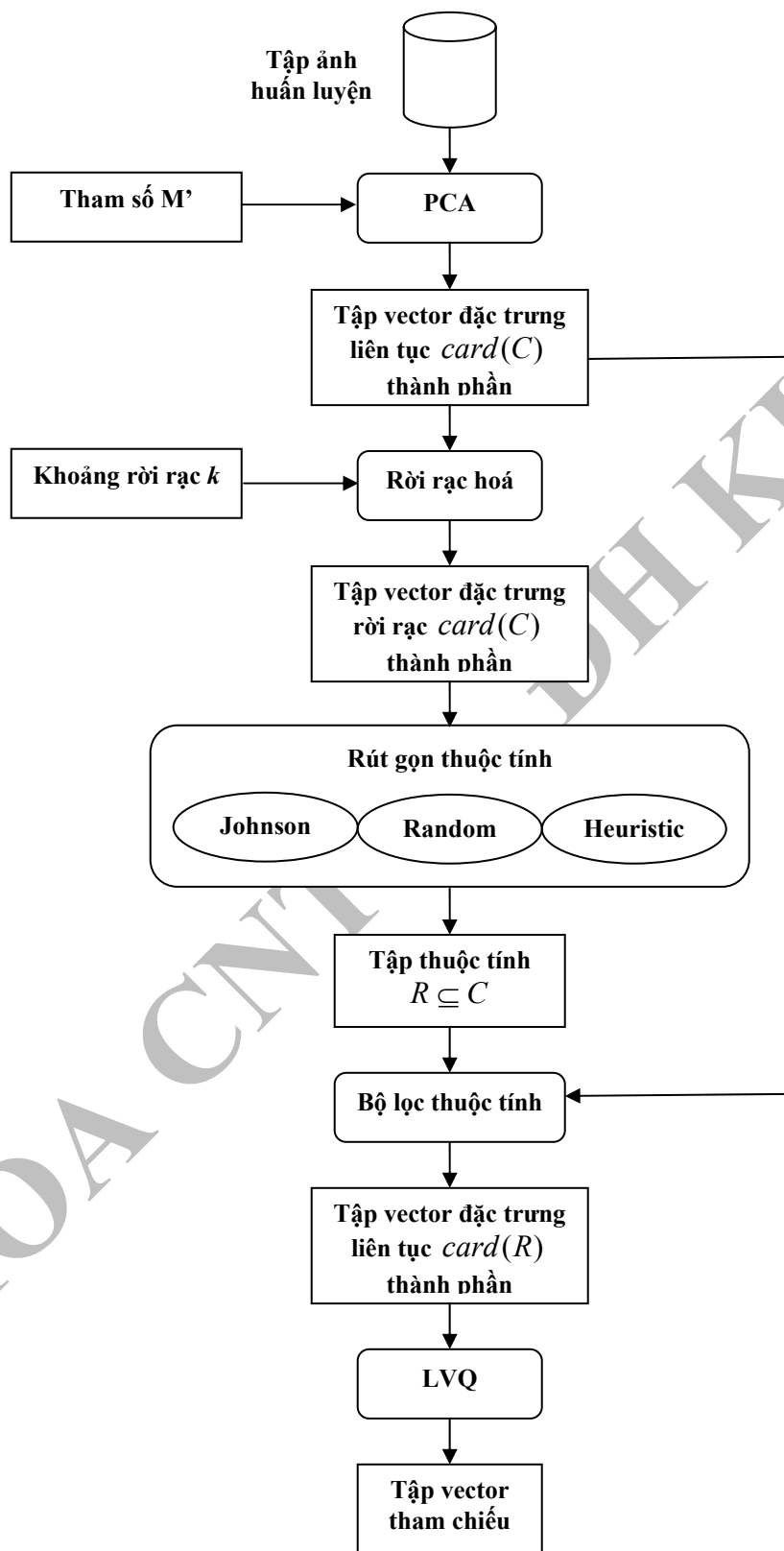
Mô hình này là mở rộng của mô hình 1 bằng cách sử dụng lý thuyết tập thô để tìm rút gọn cho tập các thuộc tính đặc trưng rút ra trong giai đoạn huấn luyện. Lưu ý rằng do các khái niệm liên quan đến tập thô chỉ làm việc trên thuộc tính rời rạc nên ta cần sử dụng thêm bước rời rạc hoá tập thuộc tính trước khi làm đầu vào cho các thuật toán tập thô.

- Giai đoạn huấn luyện :
 - Rút trích đặc trưng cho tập ảnh huấn luyện bằng phương pháp phân tích thành phần chính như mô hình 1, kết quả nhận được là tập các vector đặc trưng có giá trị thực. Gọi tập các thuộc tính của các vector này là C , $|C| = M'$.
 - Áp dụng thuật toán rời rạc hoá cho tập các vector đặc trưng này, kết quả nhận được là tập các vector đặc trưng nhận giá trị rời rạc gồm C thành phần. Ta sử dụng thuật toán rời rạc : chia miền giá trị của mỗi thuộc tính thành k khoảng bằng nhau, nếu giá trị thực tại một thuộc tính thuộc vào khoảng thứ $t \in \{0, 1, \dots, k-1\}$ thì giá trị rời rạc tương ứng là t . Cũng như M' , giá trị của khoảng rời rạc k do người dùng cung cấp (Xem phần 3.3.4).
 - Sử dụng lý thuyết tập thô để tìm tập thuộc tính rút gọn R của tập thuộc tính C ban đầu. Ta sử dụng ba thuật toán : Johnson, Random được trình bày trong chương 1, và heuristic vừa được trình bày ở phần trên (ta tạm gọi là phương pháp Heuristic).
 - Với tập vector đặc trưng giá trị thực $card(C)$ thành phần, giữ lại các thành phần tương ứng với các thuộc tính trong R . Ta được tập vector đặc trưng mới có giá trị thực gồm $card(R)$ thuộc tính.
 - Sử dụng thuật toán lượng hoá vector đối với tập vector đặc trưng giá trị thực $card(R)$ thành phần để tạo tập các vector tham chiếu.
- Giai đoạn phân lớp :
 - Rút trích đặc trưng cho tập ảnh huấn luyện bằng phương pháp phân tích thành phần chính như mô hình 1, kết quả nhận được là tập các vector đặc trưng có giá trị thực. Gọi tập các thuộc tính của các vector này là C .

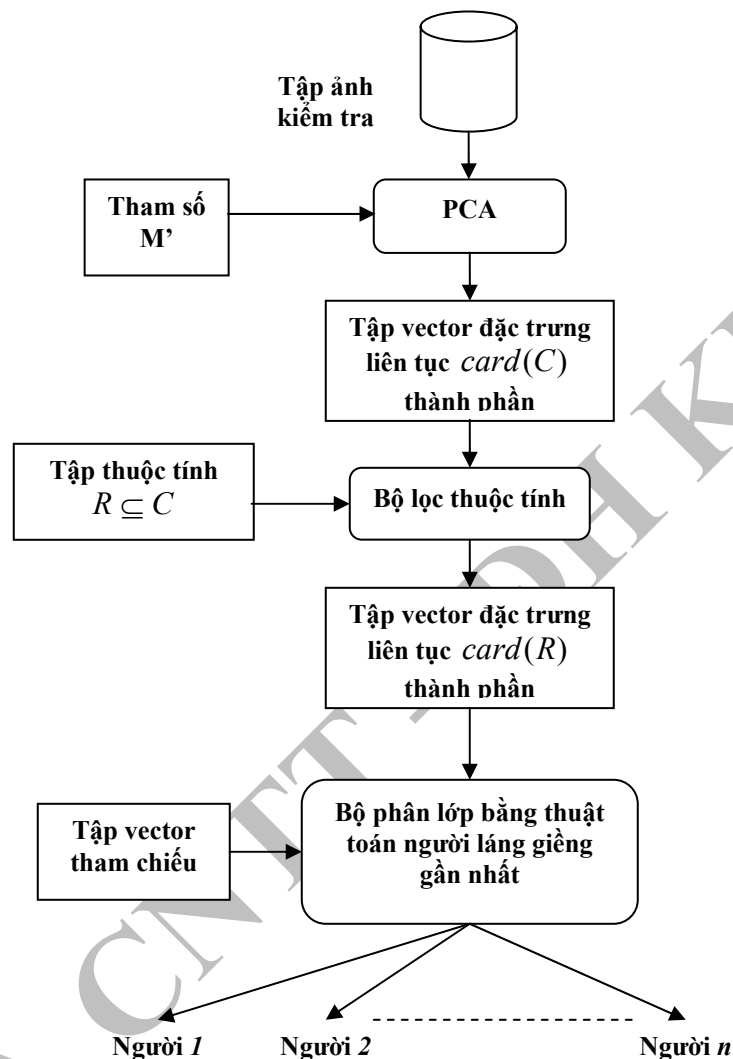
- Với tập vector đặc trưng giá trị thực $card(C)$ thành phần, giữ lại các thành phần tương ứng với các thuộc tính trong R . Ta được tập vector đặc trưng mới có giá trị thực gồm $card(R)$ thuộc tính.
- Sử dụng tập vector tham chiếu, phân lớp tập vector đặc trưng $card(R)$ thành phần bằng thuật toán người láng giềng gần nhất

Hai giai đoạn huấn luyện và phân lớp lần lượt được thể hiện trong Hình 3-6 và Hình 3-7.

KHOA CNTT – ĐHQKHN



Hình 3- 6 : Giai đoạn huấn luyện tạo tập vector tham chiếu



Hình 3- 7 : Giai đoạn phân lớp tập ảnh kiểm tra

3.3.4. Vấn đề lựa chọn số khoảng rời rạc

Thuật toán rời rạc là một phần quan trọng trong mô hình thử nghiệm ở trên và góp một phần vào kết quả nhận dạng cuối cùng. Đối với thuật toán rời rạc chia đều khoảng cách được sử dụng trong mô hình, số lượng k các khoảng rời rạc của các thuộc tính nên được chọn thoả mãn điều kiện : $k < m$ với m là số lượng ảnh huấn luyện (hay số

lượng ảnh huấn luyện trung bình) của mỗi người trong tập huấn luyện. Heuristic này được đưa ra dựa trên hai lập luận sau đây :

1. Giả sử sau khi rút trích đặc trưng, một người nào đó có m vector đặc trưng v_1, v_2, \dots, v_m với tập thuộc tính (hay thành phần) là C . Gọi các giá trị của tập vector này tại thuộc tính $a \in C$ là $v_1[a], v_2[a], \dots, v_m[a]$ và giá trị $v_i[a] \in R$, sau quá trình rời rạc hóa, sẽ trở thành giá trị $d_i[a] \in \{0, 1, \dots, k-1\}, \forall i = 1, 2, \dots, m$. Theo nguyên lý Dirichlet [2], trong tập các giá trị rời rạc $\{d_0[a], d_1[a], \dots, d_{k-1}[a]\}$ tồn tại $\left\lceil \frac{m}{k} \right\rceil \geq 2$ (vì $k < m$) giá trị rời rạc bằng nhau, hay : $\exists i, j \in \{1, 2, \dots, m\}, i \neq j, d_i[a] = d_j[a]$. Đây cũng là điều khá hợp lý mà ta mong muốn : với cùng một người, tại mọi thuộc tính $a \in C$, luôn có ít nhất hai vector đặc trưng nhận giá trị như nhau tại a .
2. Khi số lượng khoảng rời rạc càng lớn thì xác suất để hai vector đặc trưng nhận giá trị như nhau tại mỗi thuộc tính sẽ càng nhỏ. Điều này cũng có nghĩa là khả năng phân biệt các đối tượng (vector đặc trưng) của các thuộc tính càng lớn, dẫn đến các rút gọn có kích thước càng nhỏ. Trong trường hợp xấu nhất, tồn tại một thuộc tính $a \in C$ mà tại đó tất cả các vector đặc trưng đều nhận giá trị khác nhau, và như vậy $R = \{a\}$ là một rút gọn của tập thuộc tính C . Rõ ràng việc nhận dạng với tập thuộc tính R như vậy sẽ cho kết quả không cao.

Heuristic chọn khoảng rời rạc vừa trình bày đã được kiểm nghiệm thực tế. Trong trường hợp số khoảng rời rạc quá lớn, tập thuộc tính rút gọn có kích thước rất nhỏ (2 hoặc 3) phần tử dẫn đến kết quả nhận dạng thấp.

Trong chương trình thử nghiệm, với tập dữ liệu huấn luyện 24 ảnh / người, ta có thể chọn số khoảng rời rạc là 10.

Chương 4

Cài Đặt Chương Trình Và Thử Nghiệm

-----oOo-----

4.1. Chương trình cài đặt

4.1.1. Ngôn ngữ và môi trường

Chương trình thử nghiệm được cài đặt bằng ngôn ngữ C++ trên hệ điều hành Microsoft Windows NT/2000 và Microsoft Windows XP, sử dụng môi trường lập trình Microsoft Visual C++ 6.0 IDE.

4.1.2. Tổ chức thư mục mã nguồn

Thư mục *Dib* : Chứa tập tin cho thư viện xử lý ảnh.

Thư mục *Newmat* : Thư viện ma trận.

Thư mục *Stefan Chekanov* : Chứa tập tin cho các lớp truy xuất cơ sở dữ liệu. Chức năng lưu trữ dữ liệu không được hỗ trợ trong chương trình cài đặt này.

Thư mục *My Classes* : Các lớp xử lý quan trọng.

Thư mục *Others* : Chứa tập tin cho các lớp tiện ích khác.

Các tập tin cho các lớp giao diện : nằm trong thư mục hiện hành *Face Recognition*.

4.1.3. Một số lớp quan trọng

1. Lớp bảng quyết định

- Tên lớp : CDecisionTable
- Tập tin :

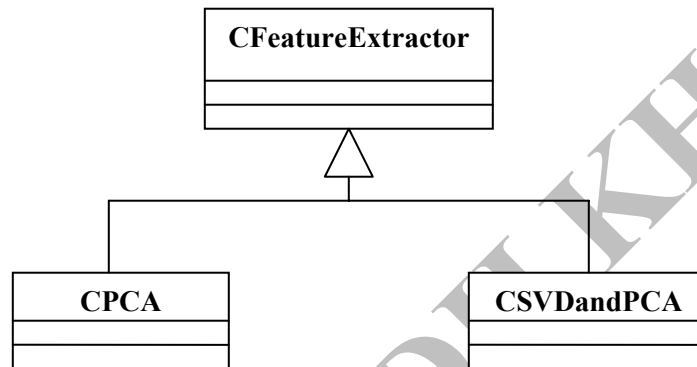
My Classes \ DecisionTable.h

My Classes \ DecisionTable.cpp

- Chức năng : Biểu diễn một hệ thông tin có một thuộc tính quyết định (bảng quyết định) và các thao tác tương ứng. Sử dụng để lưu tập vector đặc trưng của tập huấn luyện, kiểm tra.

2. Các lớp thực hiện rút trích đặc trưng

2.1. Sơ đồ lớp



Hình 4- 1 : Sơ đồ các lớp rút trích đặc trưng

2.2. Lớp rút trích đặc trưng cơ sở

- Tên lớp : CFeatureExtractor
- Tập tin :
My Classes \ FeatureExtractor.h
My Classes \ FeatureExtractor.cpp
- Chức năng : Lớp cơ sở cho các lớp rút trích đặc trưng

2.3. Lớp phân tích thành phần chính (PCA)

- Tên lớp : CPCA
- Tập tin :
My Classes \ PCA.h
My Classes \ PCA.cpp
- Chức năng : Thực hiện rút trích đặc trưng tập ảnh bằng phương pháp phân tích thành phần chính.

2.4. Lớp rút trích đặc trưng bằng phương pháp SVD và PCA

- Tên lớp : CSVDandPCA

- Tập tin :

My Classes \ SVDandPCA.h

My Classes \ SVDandPCA.cpp

- Chức năng : Rút trích đặc trưng ảnh bằng phương pháp SVD kết hợp PCA.

Lưu ý : Trong chương trình cài đặt có thêm hai mô hình 3 và 4 kết hợp phép biến đổi SVD và PCA để rút trích đặc trưng ảnh mặt. Tuy nhiên do chưa có tài liệu nào chứng minh được hiệu quả của SVD trong nhận dạng, và kết quả kiểm nghiệm trên dữ liệu ORL cho thấy tác dụng rút trích đặc trưng của SVD chưa cao nên đây chỉ là 2 cài đặt phụ, không được sử dụng để minh họa hay thống kê trong luận văn này. Về phép biến đổi SVD, xin xem [1].

3. Lớp rời rạc hoá

- Tên lớp : CEqualWidthDiscretizer

- Tập tin :

My Classes \ EqualWidthDiscretizer.h

My Classes \ EqualWidthDiscretizer.cpp

- Chức năng : Rời rạc dữ liệu bằng phương pháp chia đều miền giá trị.

4. Lớp thuật toán tập thô

- Tên lớp : CRoughSetAlgorithm

- Tập tin :

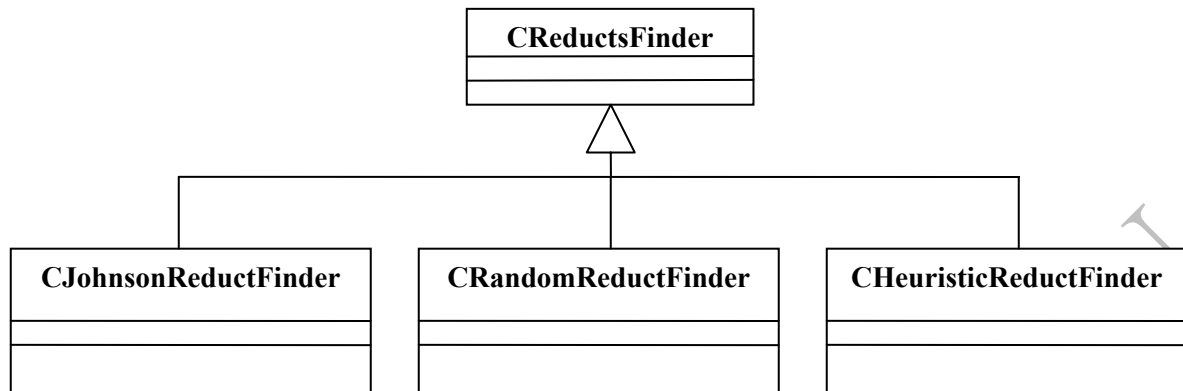
My Classes \ RoughSetAlgorithm.h

My Classes \ RoughSetAlgorithm.cpp

- Chức năng : Thực hiện các thuật toán trong lý thuyết tập thô.

5. Các lớp rút gọn thuộc tính

5.1. Sơ đồ lớp



Hình 4- 2 : Sơ đồ các lớp thực hiện rút gọn đặc trưng

5.2. Lớp cơ sở

- Tên lớp : CReductsFinder
- Tập tin :
My Classes \ ReductsFinder.h
My Classes \ ReductsFinder.cpp
- Chức năng : Lớp cơ sở cho các thuật toán tìm tập rút gọn.

5.3. Lớp chiến lược Johnson

- Tên lớp : CJohnsonReductFinder
- Tập tin :
My Classes \ JohnsonReductFinder.h
My Classes \ JohnsonReductFinder.cpp
- Chức năng : Tìm tập thuộc tính rút gọn theo chiến lược Johnson.

5.4. Lớp chiến lược ngẫu nhiên

- Tên lớp : CRandomReductFinder
- Tập tin :
My Classes \ RandomReductFinder.h
My Classes \ RandomReductFinder.cpp
- Chức năng : Tìm tập thuộc tính rút gọn theo chiến lược ngẫu nhiên xác suất.

5.5. Lớp Heuristic

- Tên lớp : CRSReductFinder
- Tập tin :
 My Classes \ RSReductFinder.h
 My Classes \ RSReductFinder.cpp
- Chức năng : Tìm tập thuộc tính rút gọn dựa theo tiêu chuẩn đưa ra trong hệ *GDT – RS*.

6. Lớp mạng lượng hoá vector (LVQ)

- Tên lớp : CLVQNet
- Tập tin :
 My Classes \ LVQNet.h
 My Classes \ LVQNet.cpp
- Chức năng : Khởi tạo và huấn luyện tập các vector tham chiếu.

7. Lớp thuật toán phân loại người láng giềng gần nhất

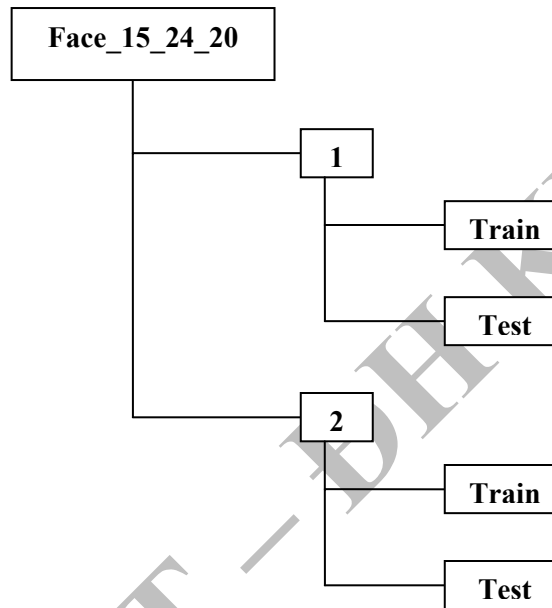
- Tên lớp : CNearestNeighbor
- Tập tin :
 My Classes \ NearestNeighbor.h
 My Classes \ NearestNeighbor.cpp
- Chức năng : Thực hiện phân lớp tập vector đặc trưng của dữ liệu kiểm tra theo tập các vector tham chiếu tạo bởi mạng lượng hoá vector.

4.2. Tổ chức dữ liệu thử nghiệm

- Các tập dữ liệu huấn luyện, kiểm tra lần lượt được đặt trong thư mục *Train* và *Test*.
- Các tập có cùng số người và số ảnh huấn luyện / người được đặt vào cùng một thư mục có tên :

Face_<số người>_<số ảnh huấn luyện mỗi người>_<số ảnh kiểm tra mỗi người>.

Ví dụ : Thư mục Face_15_24_20 chứa các tập huấn luyện, kiểm tra với 15 người, mỗi người có 24 ảnh huấn luyện và 20 ảnh kiểm tra. Các thư mục nằm trong Face_15_24_20 là các thư mục chứa từng tập huấn luyện, kiểm tra. Một sơ đồ cây minh hoạ việc tổ chức này :

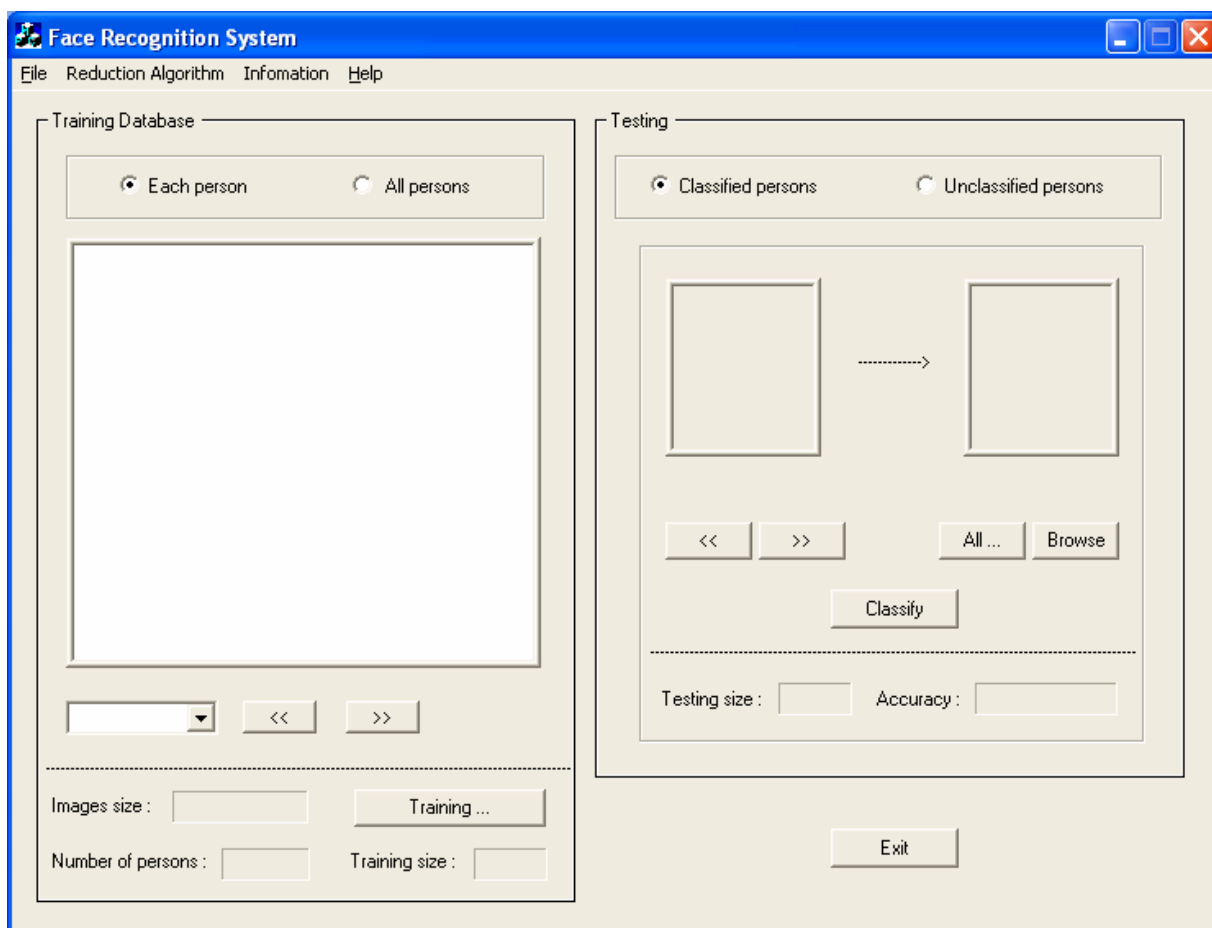


Hình 4- 3 : Minh hoạ tổ chức tập dữ liệu

- Trong các thư mục *Train* và *Test*, ảnh của mỗi người được gom vào từng thư mục mang tên là các số tự nhiên. Các số tự nhiên này là bắt buộc và sẽ được sử dụng như tên tham chiếu tới người tương ứng (tức là chúng ta không sử dụng thư mục mang tên của từng người).

4.3. Hướng dẫn và minh hoạ sử dụng chương trình

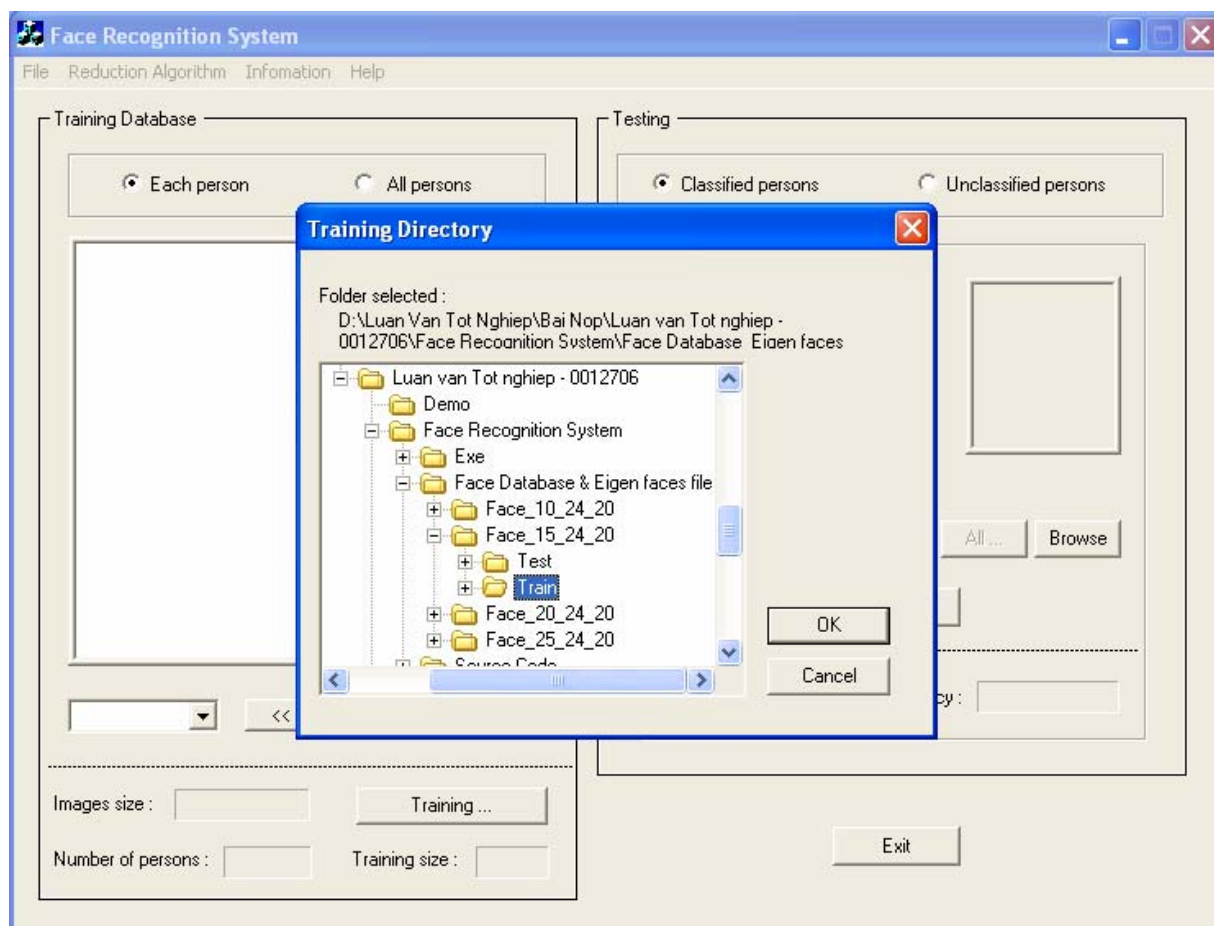
4.3.1. Màn hình chính



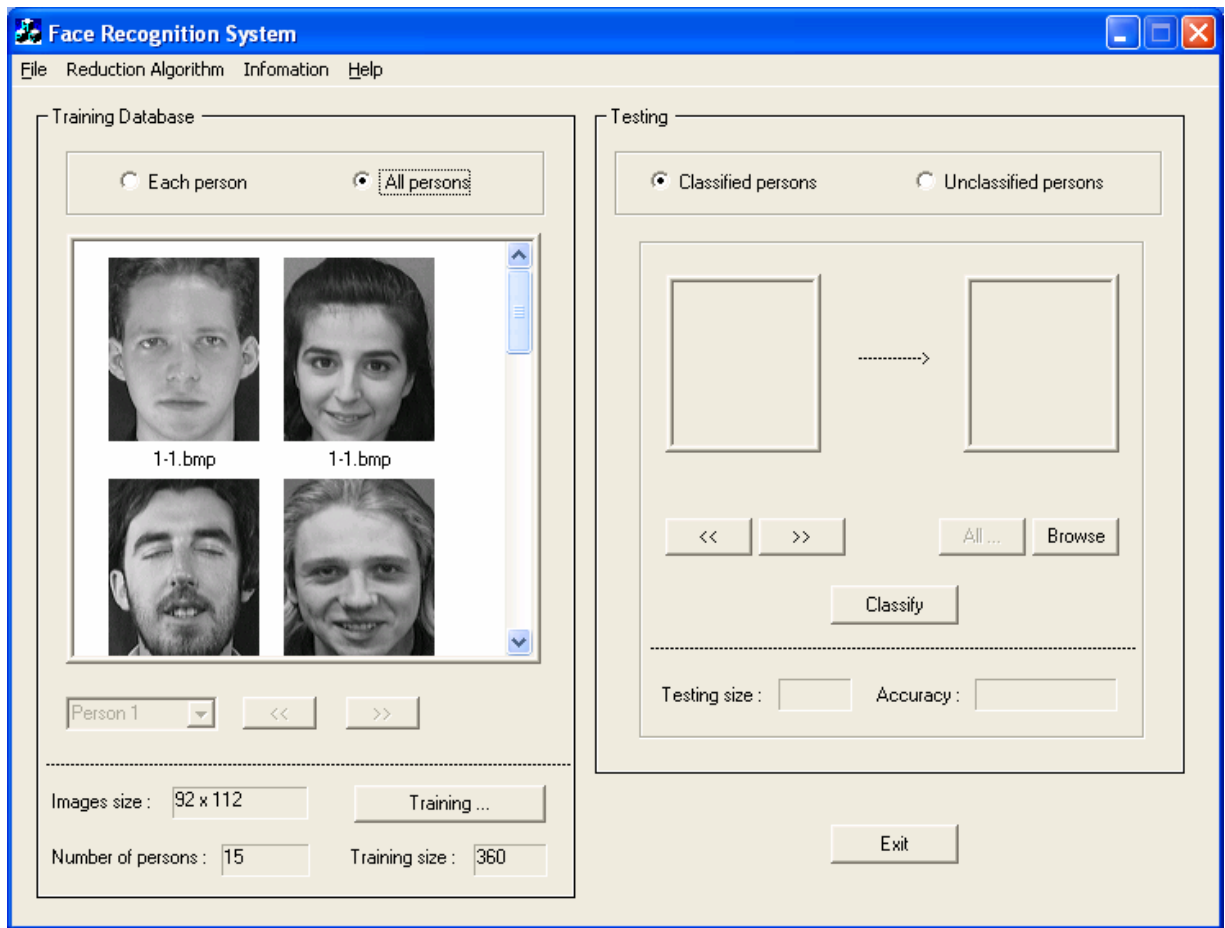
Hình 4- 4 : Màn hình chính của chương trình

4.3.2. Nhập tập ảnh huấn luyện

- Menu : *File / New training database ...*
- Chọn thư mục chứa tập ảnh huấn luyện, trong Hình 4-5 thư mục này là *Train*.
- Hình ảnh :



Hình 4- 5 : Chọn tập ảnh huấn luyện



Hình 4- 6 : Sau khi chọn tập ảnh huấn luyện

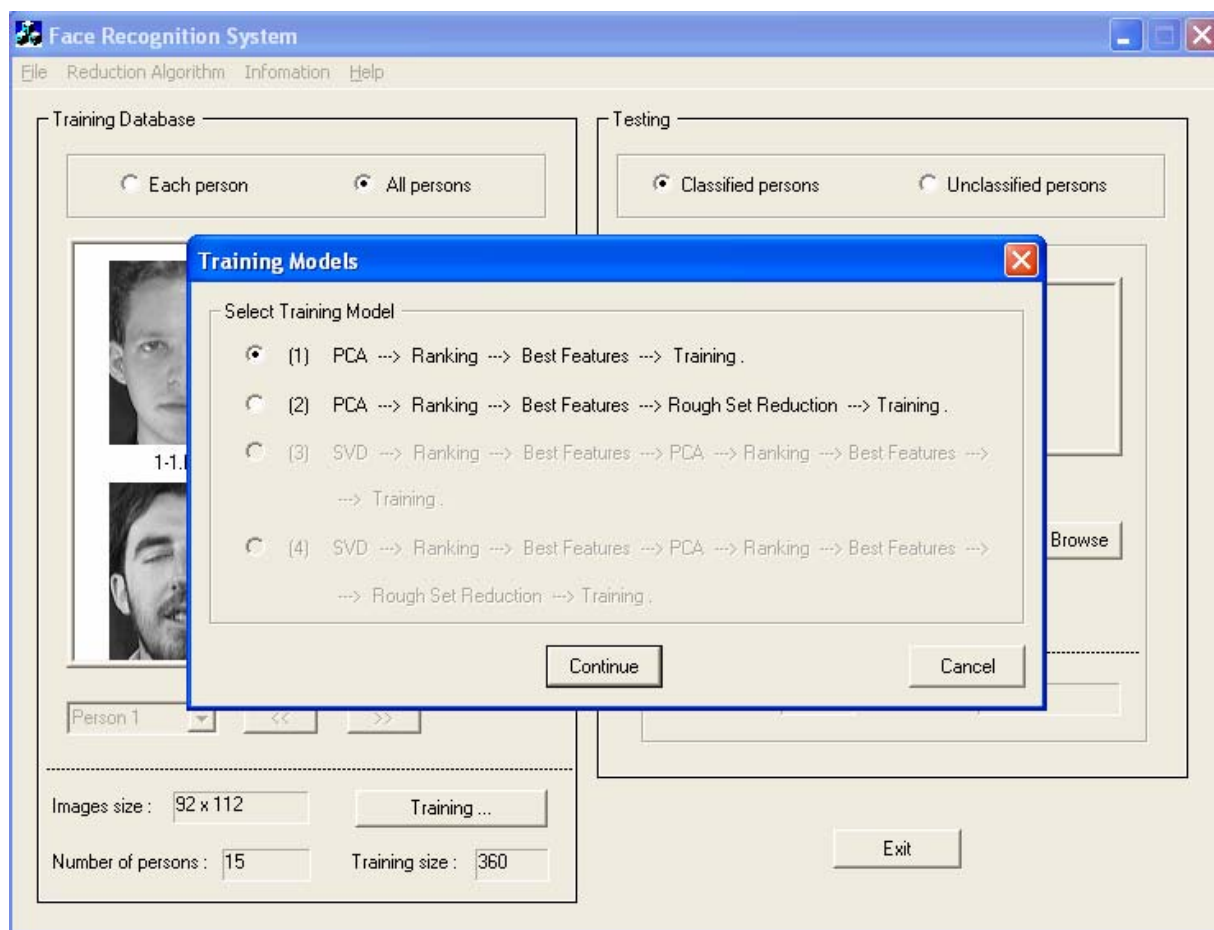
4.3.3. Chọn thuật toán rút gọn thuộc tính

- Menu : *Reduction Algorithm*. Thuật toán được chọn sẽ được áp dụng vào các mô hình 2 và 4. Thuật toán khởi tạo là Johnson.

4.3.4. Quá trình huấn luyện

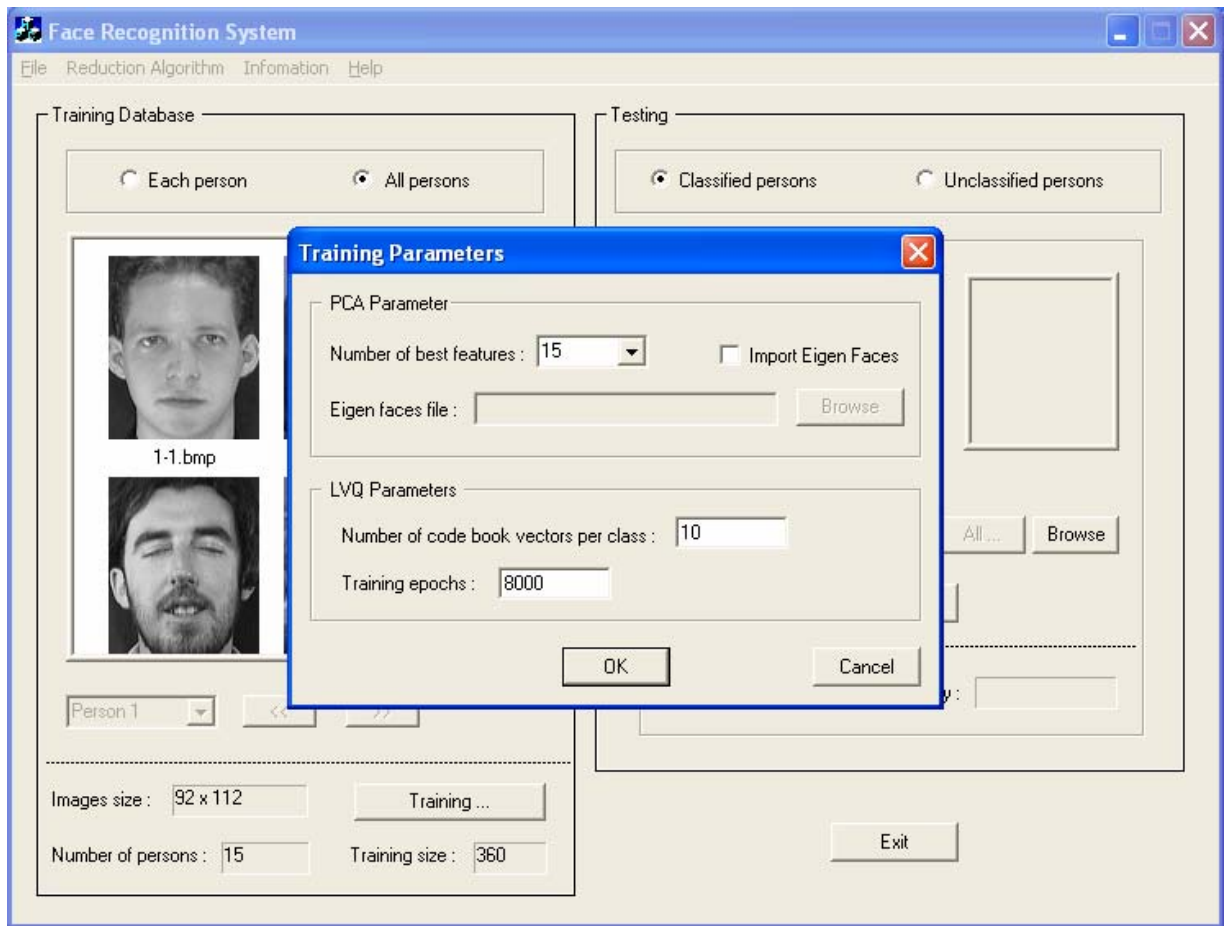
- Nút nhấn : *Training ...*
- Chọn mô hình sử dụng.
- Hình ảnh :

Chương 4 – Cài đặt chương trình và thử nghiệm



Hình 4- 7 : Chọn mô hình huấn luyện

- Chọn tham số của mô hình tương ứng.
- Hình ảnh :



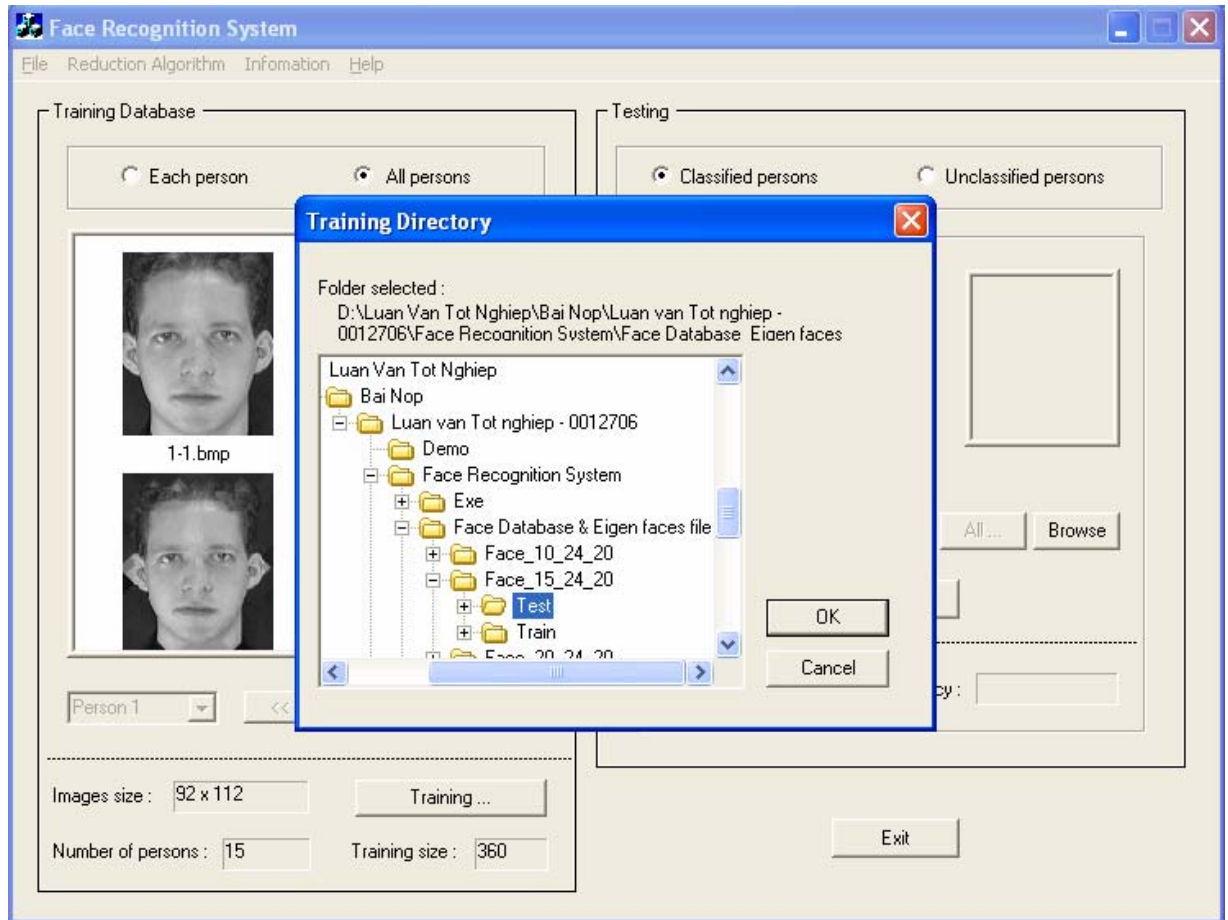
Hình 4- 8 : Chọn tham số cho mô hình

4.3.5. Quá trình phân lớp

- Có 2 chế độ :
 - Phân lớp tập ảnh có giám sát : Tập ảnh đưa vào đã được phân vào các thư mục tương ứng với từng người (xem phần mô tả dữ liệu 4.2). Chẳng hạn như các thư mục *Test* nói trong 4.2 .
 - Phân lớp tập ảnh không giám sát : Tất cả các ảnh kiểm tra nằm trực tiếp trong một thư mục, không thuộc các thư mục của từng người.
- Chọn chế độ :
 - *Classified persons*
 - *Unclassified persons*

Chương 4 – Cài đặt chương trình và thử nghiệm

- Mở thư mục : nút nhấn *Browse*.
- Hình ảnh :



Hình 4- 9 : Chọn thư mục phân lớp trong chế độ giám sát

- Nhấn nút *Classify* để bắt đầu phân lớp.
- Kết quả phân lớp :
 - Chế độ có giám sát : Tỷ lệ phần trăm số ảnh được phân lớp đúng.
 - Chế độ không giám sát : *Unknown*.

4.3.6. Xem thông tin

- Các mục trong menu *Information* :
 - *Real – attribute training table* : Bảng huấn luyện ở trạng thái giá trị thực.

Chương 4 – Cài đặt chương trình và thử nghiệm

- *Discrete – attribute training table* : Bảng huấn luyện ở trạng thái giá trị rời rạc.
- *Real – attribute testing table* : Bảng kiểm tra ở trạng thái giá trị thực.
- *Discrete – attribute testing table* : Bảng kiểm tra ở trạng thái giá trị rời rạc.
- *Set of attributes* : Tập thuộc tính sử dụng để phân lớp. Đối với mô hình 2 và 4 đây là tập các thuộc tính rút gọn, với mô hình 1 và 3 đây là tập tất cả các thuộc tính đặc trưng.
- *Training type* : Xem các thông tin tham số, thuật toán của mô hình vừa sử dụng.

4.4. Một số kết quả

4.4.1. Thư mục Face_10_24_20

- Số lượng người : 10
- Số ảnh huấn luyện / người : 24
- Số ảnh kiểm tra / người : 20
- Số vector tham chiếu (LVQ) : 10
- Chu kỳ huấn luyện : 5000
- Số khoảng rời rạc (bước rời rạc hoá) : 3
- Bảng số liệu

STT	Tham số PCA	Kết quả mô hình 1 (%)	Mô hình 2					
			Johnson		Heuristic		Random	
			Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)
1	5	87.5	5	87.5	5	87.5	5	87.5
2	6	93.5	6	93.5	6	93.5	6	93.5

Chương 4 – Cài đặt chương trình và thử nghiệm

3	7	97.5	7	97.5	6	94.5	6	94
4	8	97	8	97	6	87	7	94.5
5	9	97.5	8	97	6	94	8	93.5
6	10	98	8	97	6	82.5	8	95.5
7	11	98	8	97	6	82.5	8	98
8	12	99	7	94	6	82.5	7	97
9	13	99	7	94	6	82.5	9	85
10	14	99.5	7	94	6	82.5	8	93
11	15	100	7	94	6	82.5	8	95.5

Bảng 4- 1 : Kết quả huấn luyện, kiểm tra tập Face_10_24_20

4.4.2. Thư mục Face_15_24_20

- Số lượng người : 15
- Số ảnh huấn luyện / người : 24
- Số ảnh kiểm tra / người : 20
- Số vector tham chiếu (LVQ) : 10
- Chu kỳ huấn luyện : 8000
- Số khoảng rời rạc (bước rời rạc hoá) : 3

STT	Tham số PCA	Kết quả mô hình 1 (%)	Mô hình 2					
			Johnson		Heuristic		Random	
			Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)
1	5	85	5	85	5	85	5	85
2	6	87	6	87	6	87	6	87
3	7	89	7	89	7	89	7	89

Chương 4 – Cài đặt chương trình và thử nghiệm

4	8	90.33	8	90.33	8	90.33	8	90.33
5	9	90.67	9	90.67	8	90.33	9	90.67
6	10	91.33	8	89.67	6	94.33	8	97.67
7	11	91.33	8	89.67	6	94.33	9	90.67
8	12	92	8	89.67	7	97.67	8	90.33
9	13	92.33	8	89.67	7	97.67	9	90.67
10	14	92.33	9	91.67	7	97.67	10	88.67
11	15	92.33	9	91.67	7	97.67	8	89.67

Bảng 4- 2 : Kết quả huấn luyện, kiểm tra tập Face_15_24_20

4.4.3. Thư mục Face_20_24_20

- Số lượng người : 20
- Số ảnh huấn luyện / người : 24
- Số ảnh kiểm tra / người : 20
- Số vector tham chiếu (LVQ) : 10
- Chu kỳ huấn luyện : 10000
- Số khoảng rời rạc (bước rời rạc hoá) : 3

STT	Tham số PCA	Kết quả mô hình 1 (%)	Mô hình 2					
			Johnson		Heuristic		Random	
			Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)
1	5	76.5	5	76.5	5	76.5	5	76.5
2	6	82	6	82	6	82	6	82
3	7	87	7	87	7	87	7	87
4	8	82.75	8	82.75	8	82.75	8	82.75

Chương 4 – Cài đặt chương trình và thử nghiệm

5	9	82	9	82	9	82	9	82
6	10	85.25	10	85.25	10	85.25	10	85.25
7	11	86	11	86	11	86	11	86
8	12	86.25	10	85.5	9	88.5	11	90.33
9	13	92.33	10	85.5	8	89.25	10	90.67
10	14	87.25	11	81.25	9	94.5	11	88.67
11	15	86.5	11	81.25	10	95.5	13	89.67
12	16	86.5	11	81.25	10	95.5	12	82
13	17	86.5	10	81.75	10	95.5	11	89.5
14	18	87	10	81.75	11	90.25	13	91.5
15	19	87.25	11	81.75	11	90.25	12	84.5
16	20	87.25	11	81.75	11	90.25	12	79.5

Bảng 4- 3 : Kết quả huấn luyện, kiểm tra tập Face_20_24_20

4.4.4. Thư mục Face_25_24_20

- Số lượng người : 25
- Số ảnh huấn luyện / người : 24
- Số ảnh kiểm tra / người : 20
- Số vector tham chiếu (LVQ) : 10
- Chu kỳ huấn luyện : 12500
- Số khoảng rời rạc (bước rời rạc hoá) : 3

STT	Tham số PCA	Kết quả mô hình 1 (%)	Mô hình 2					
			Johnson		Heuristic		Random	
			Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)	Số thuộc tính	Kết quả (%)

Chương 4 – Cài đặt chương trình và thử nghiệm

1	10	83.8	10	83.8	10	83.8	10	83.8
2	11	86.6	11	86.6	11	86.6	10	83.8
3	12	88.2	12	88.2	11	84.4	12	88.2
4	13	84.6	12	84.6	10	77.8	11	86.6
5	14	85.4	11	90.2	9	80.4	13	84.6
6	15	85	10	83.2	10	88.8	13	89
7	16	84.8	10	83.2	10	88.8	12	80.6
8	17	84.8	10	83.2	10	96.4	12	84.8
9	18	85	10	83.2	10	86.2	14	84.2
10	19	84.8	10	83.2	10	86.2	14	79.8
11	20	85.6	10	83.2	10	86.2	12	84.6
12	21	86.2	10	83.2	10	86.2	14	84.8
13	22	85.6	10	83.2	10	86.2	14	76
14	23	86.4	10	83.2	10	91	12	74.4
15	24	85.8	10	83.2	10	91	13	81.8
16	25	85.8	11	81.75	10	91	14	79.2

Bảng 4- 4 : K ết quả huấn luyện, kiểm tra tập Face_25_24_20

4.5. Nhận xét kết quả

Từ các kết quả thử nghiệm ở trên ta có một số nhận xét :

1. Trong 3 thuật toán tìm rút gọn tập thuộc tính, chiến lược Heuristic tỏ ra vượt trội hơn và chiến lược ngẫu nhiên xác suất là thấp nhất về khả năng nhận dạng.
2. Trong các bảng thống kê trên, những kết quả in đậm thể hiện được lợi ích của tiếp cận tập thô trong lựa chọn và rút gọn đặc trưng. Thứ nhất, những phần này phải tương ứng với kết quả nhận dạng từ 90% trở lên. Thứ hai, có hai cách so sánh :

- So sánh theo chiều ngang : kết quả cho thấy chất lượng nhận dạng sau khi rút gọn đặc trưng sẽ tăng lên. Ví dụ : với tập Face_15_24_20, nếu giữ nguyên 10 đặc trưng thì kết quả nhận dạng là 91.33, trong khi kết quả sau khi rút gọn còn 6 đặc trưng là 94.33.
 - So sánh theo chiều xiên : với 2 tập đặc trưng cùng kích thước, tập đặc trưng nhận được từ quá trình rút gọn cho chất lượng tốt hơn tập đặc trưng tương ứng với các thành phần tốt nhất của phân tích thành phần chính. Ví dụ : với tập Face_15_24_20, tập đặc trưng rút gọn kích thước 6 cho kết quả nhận dạng 94.33%, trong khi tập tương ứng với 6 thành phần tốt nhất của phân tích thành phần chính cho kết quả nhận dạng 87%.
3. Trong những thử nghiệm trên chúng ta sử dụng thuật toán phân lớp người láng giềng gần nhất. Điều này cũng có thể là nguyên nhân chất lượng phân loại không cao : chưa bao giờ chúng ta đạt được kết quả nhận dạng 100 %.

Chương 5

Tự Đánh Giá Và Hướng Phát Triển Đề Nghị

-----oOo-----

5.1. Tự đánh giá

Với những trình bày lý thuyết cũng như kết quả thực nghiệm, luận văn đã đạt được một số thành quả sau đây :

- ❖ Trình bày rõ ràng và có hệ thống lý thuyết tập thô, minh hoạ đầy đủ các khái niệm bằng nhiều ví dụ.
- ❖ Tìm hiểu một số vấn đề tổng quát trong bài toán nhận dạng mặt người : mô hình chung, các khó khăn gặp phải.
- ❖ Nghiên cứu phương pháp phân tích thành phần chính trong bài toán nhận dạng, một số thuật toán học lượng hoá vector ứng dụng trong phân lớp dữ liệu.
- ❖ Cài đặt chương trình ứng dụng nhận dạng mặt người, trong đó kết hợp lý thuyết tập thô vào giai đoạn lựa chọn và rút gọn đặc trưng.
- ❖ Đưa ra nhận xét từ số liệu thống kê và biểu đồ để thấy được mặt mạnh, yếu của phương pháp lựa chọn và rút gọn đặc trưng bằng lý thuyết tập thô. Đưa ra một số lý giải ban đầu cho các kết quả.

Tuy vậy, luận văn vẫn còn chưa hoàn thiện ở các điểm sau :

- ❖ Chưa đưa vào các phương pháp rút trích đặc trưng khác để có điều kiện đánh giá chính xác.
- ❖ Chưa so sánh được hiệu quả áp dụng lý thuyết tập thô trong bài toán lựa chọn và rút gọn đặc trưng với các tiếp cận khác.
- ❖ Thuật toán phân lớp sử dụng còn quá đơn giản.

- ❖ Tập dữ liệu chưa phong phú.

5.2. Hướng phát triển đề nghị

Dựa trên một số điểm tự đánh giá trên, sau đây là một số hướng để phát triển vấn đề được nêu trong luận văn này :

- ❖ Nghiên cứu các thuật toán rút trích đặc trưng khác như : ánh xạ tự tổ chức (SOM), phép biến đổi Cosine rời rạc (DCT),...và so sánh với phương pháp phân tích thành phần chính.
- ❖ Nghiên cứu các thuật toán sử dụng trong phân lớp như Support Vector Machine, mô hình Markov ẩn,...Việc sử dụng các mô hình này chắc chắn sẽ mang lại kết quả nhận dạng cao hơn bộ phân lớp người láng giềng gần nhất được sử dụng trong luận văn.
- ❖ Nghiên cứu các phương pháp lựa chọn và rút gọn đặc trưng khác và so sánh với tiếp cận bằng lý thuyết tập thô.

Tài Liệu Tham Khảo

Chương 1 :

- [1] Jan Komorowski, Lech Polkowski, Andrzej Skowron : *Rough Sets : A Tutorial*.
- [2] Roman W. Swiniarski (2001) : *Rough set methods in feature reduction and classification* – Int. J. Appl. Math. Comput. Sci., 2001, Vol. 11, No. 3, 565 – 582.
- [3] Zhenghong Yang, Tianning Li, Feng Jin, Shuyun Xu (2002) : *Rough Set in Data Mining*.
- [4] Ron Kohavi, Brian Frasca (1994) : *Useful Feature Subsets and Rough Set Reducts*.
- [5] Nguyễn Hoàng Phương, Nadipuram R. Prasad, Lê Linh Phong : *Nhập môn Trí tuệ tính toán*, Nhà xuất bản Khoa học Kỹ thuật (2002).
- [6] Ning Zhong, Juzhen Dong, Setsuo Ohsuga (2001) : *Using Rough Sets with Heuristic for Feature Selection*. Journal of Intelligent Information System, 16, 199 – 214, 2001.
- [7] Nguyen Sinh Hoa, Nguyen Hung Son – Institut of Computer Science, Wasaw University, Poland : *Some efficient algorithms for rough set methods*.

Chương 2 :

- [1] Matthew Turk, Alex Pentland (1991) : *Eigenfaces for Recognition*.
- [2] Ilker Atalay (M. Sc. Thesis - 1996) : *Face Recognition using eigenfaces*.
- [3] Raphael Cendrillon (1999) : *Real time face recognition using eigenfaces*.
- [4] Trần Phước Long, Nguyễn Văn Lượng (2003) : *Nhận dạng người dựa vào thông tin khuôn mặt xuất hiện trên ảnh*. Luận văn tốt nghiệp, Khoa Công nghệ Thông tin, Đại học Khoa học Tự nhiên Tp. HCM.

- [5] Shang – Hun Lin, Ph.D (IC Media Corporation) (2000) : *An Introduction to Face Recognition Technology*. Informing Science Special Issue on Multimedia Informing Technology – Part 2, Volume 3 No 1, 2000.
- [6] Sezin Kaymak (2003) : *Face Detection, Recognition and Rescontruction using Eigenfaces*.
- [7] *LVQ_Pak*, Neural Network Research Centre - Laboratory of Computer and Information Science – Helsinki University Of Technology.
http://www.cis.hut.fi/research/lvq_pak/.
- [8] Dr. *Justin D. Wang* (2003) : Neural Network, Department Science and Computer Engineering, La Trobe University, Australia.
- [9] Linsay I Smith (2002) : *A tutorial on Principle Components Analysis*.

Chương 3 :

- [1] Ning Zhong, Juzhen Dong, Setsuo Ohsuga (2001) : *Using Rough Sets with Heuristic for Feature Selection*. Journal of Intelligent Information System, 16, 199 – 214, 2001.
- [2] Kenneth H. Rosen : *Discrete Mathematics and Its Application*, McGraw – Hill, 1994. Bản dịch tiếng Việt : *Toán học rời rạc ứng dụng trong Tin học*, Nhà xuất bản Khoa học Kỹ thuật, 1998.
- [3] Roman W. Swiniarski (2001) : *Rough set methods in feature reduction and classification* – Int. J. Appl. Math. Comput. Sci., 2001, Vol. 11, No. 3, 565 – 582.

Chương 4 :

- [1] Roman W. Swiniarski (2001) : *Rough set methods in feature reduction and classification* – Int. J. Appl. Math. Comput. Sci., 2001, Vol. 11, No. 3, 565 – 582.
- [2] Nguyen Sinh Hoa, Nguyen Hung Son – Institut of Computer Science,

Wasaw University, Poland : *Some efficient algorithms for rough set methods.*

KHOA CNTT – ĐH KHTN