

MỤC LỤC

MỤC LỤC.....	i
BẢNG CÁC KÝ HIỆU, TỪ VIẾT TẮT.....	v
DANH MỤC CÁC BẢNG.....	vi
DANH MỤC HÌNH VẼ.....	viii
MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ HỆ THÔNG TIN VÀ PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH THEO TIẾP CẬN TẬP THÔ DUNG SAI.....	8
1.1. Mở đầu.....	8
1.2. Các khái niệm cơ bản về hệ thông tin.....	8
1.2.1. <i>Hệ thông tin đầy đủ và mô hình tập thô truyền thống.....</i>	8
1.2.2. <i>Hệ thông tin không đầy đủ và mô hình tập thô dung sai.....</i>	12
1.3. Phương pháp rút gọn thuộc tính theo tiếp cận tập thô dung sai.....	14
1.3.2. <i>Phương pháp rút gọn thuộc tính theo tiếp cận lai ghép lọc - đóng gói</i>	
17	
1.3.3. <i>Bài toán phân lớp trong khai phá dữ liệu.....</i>	18
1.4. Các nghiên cứu liên quan và các vấn đề còn tồn tại.....	21
1.4.1. <i>Các nghiên cứu liên quan đến rút gọn thuộc tính trong bảng quyết định không đầy đủ.....</i>	21
1.4.2. <i>Các nghiên cứu liên quan đến rút gọn thuộc tính trong bảng quyết định thay đổi.....</i>	22
1.4.3. <i>Các vấn đề còn tồn tại và mục tiêu nghiên cứu của luận án.....</i>	26
1.5. Bộ dữ liệu thực nghiệm.....	27
1.6. Kết luận chương 1.....	27

CHƯƠNG 2. PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH TRONG BẢNG QUYẾT ĐỊNH KHÔNG ĐẦY ĐỦ KHI TẬP ĐỐI TƯỢNG THAY ĐỔI ..	28
2.1. Mở đầu.....	28
2.2. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung, loại bỏ tập đối tượng.....	29
2.2.1. <i>Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định trong trường hợp bổ sung tập đối tượng.....</i>	30
2.2.2. <i>Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định trong trường hợp loại bỏ tập đối tượng.....</i>	37
2.3. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi tập đối tượng thay đổi giá trị.....	43
2.3.1. <i>Công thức gia tăng tính khoảng cách khi tập đối tượng thay đổi giá trị</i>	43
2.3.2. <i>Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi tập đối tượng thay đổi giá trị.....</i>	48
2.3.3. <i>Thực nghiệm, đánh giá thuật toán FWIA_U_Obj.....</i>	52
2.3.4. <i>Đánh giá thuật toán FWIA_U_Obj so với việc thực hiện gián tiếp hai thuật toán IDS_IFW_DO và IDS_IFW_AO.....</i>	58
2.4. Kết luận chương 2.....	61
CHƯƠNG 3. PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH TRONG BẢNG QUYẾT ĐỊNH KHÔNG ĐẦY ĐỦ KHI TẬP THUỘC TÍNH THAY ĐỔI	62
3.1. Mở đầu.....	62
3.2. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung tập thuộc tính.....	63
3.2.1. <i>Công thức cập nhật khoảng cách khi bổ sung tập thuộc tính.....</i>	63
3.2.2. <i>Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung tập thuộc tính.....</i>	67

3.2.3. Thực nghiệm, đánh giá thuật toán FWIA_AA.....	69
3.3. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi loại bỏ tập thuộc tính.....	74
3.3.1. Công thức gia tăng cập nhật khoảng cách khi loại bỏ tập thuộc tính.	74
3.3.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi loại bỏ tập thuộc tính.....	76
3.3.3. Thực nghiệm, đánh giá thuật toán FWIA_DA.....	79
3.4. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi tập thuộc tính thay đổi giá trị.....	84
3.4.1. Công thức gia tăng tính khoảng cách khi tập thuộc tính thay đổi giá trị	84
3.4.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi tập thuộc tính thay đổi giá trị.....	88
3.4.3. Thực nghiệm, đánh giá thuật toán FWIA_U_Attr.....	91
3.4.4. Thực nghiệm, đánh giá thuật toán FWIA_U_Attr so với việc thực hiện gián tiếp hai thuật toán FWIA_DA và FWIA_AA.....	96
3.5. Kết luận chương 3.....	99
KẾT LUẬN.....	100
DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA LUẬN ÁN.....	102
TÀI LIỆU THAM KHẢO.....	103

DANH MỤC CÁC THUẬT NGỮ

STT	TÊN TIẾNG ANH	TÊN TIẾNG VIỆT
1	Rough Set	Tập thô
2	Rough set theory	Lý thuyết Tập thô
3	Tolerance Rough Set	Tập thô dung sai
4	Tolerance Relation	Quan hệ dung sai
5	Tolerance Matrix	Ma trận dung sai
6	Information System	Hệ thông tin
7	Complete Information System	Hệ thông tin đầy đủ
8	Incomplete Information System	Hệ thông tin không đầy đủ
9	Decision Table	Bảng quyết định
10	Complete Decision Table	Bảng quyết định đầy đủ
11	Incomplete Decision Table	Bảng quyết định không đầy đủ
12	Indiscernibility Relation	Quan hệ bất khả phân
13	Attribute Reduction	Rút gọn thuộc tính
14	Extraction Reduction	Rút trích thuộc tính
15	Selection Reduction	Lựa chọn thuộc tính
16	Reduct/Core	Tập rút gọn/Tập lõi
17	Core Attribute	Thuộc tính lõi
18	Reductive Attribute	Thuộc tính rút gọn
19	Redundant Attribute	Thuộc tính dư thừa
20	Disposable/Indispensable	Thuộc tính cần thiết/không cần thiết
21	Distance	Khoảng cách
22	Positive Region	Miền dương
23	Classification quality	Chất lượng phân lớp.
24	Incremental Methods	Phương pháp gia tăng
25	Filter	Lọc
26	Wrapper	Đóng gói
27	Filter - Wrapper	Lọc - Đóng gói

BẢNG CÁC KÝ HIỆU, TỪ VIẾT TẮT

STT	Ký hiệu	Điễn giải
1	$IS = (U, A, V, f)$	Hệ thông tin
2	$IIS = (U, A, V, f)$	Hệ quyết định không đầy đủ
3	$DS = (U, C \cup D)$	Bảng quyết định
4	$IDS = (U, C \cup D)$	Bảng quyết định không đầy đủ
5	U	Số đối tượng
6	C	Số thuộc tính điều kiện trong bảng quyết định
7	$u(a)$	Giá trị của đối tượng u tại thuộc tính a
8	$IND(P)$	Quan hệ P -không phân biệt được
9	U/P	Phân hoạch của U trên P
10	$[u]_P$	Lớp tương đương chứa u của phân hoạch U/P
11	$SIM(P)$	Quan hệ dung sai trên P
12	$S_P(u)$	Lớp dung sai của u trên P
13	$M(C) = [c_{ij}]_{n \times n}$	Ma trận dung sai trên C

14	$D(C, C \cup \{d\})$	Khoảng cách giữa hai tập thuộc tính C và $C \cup \{d\}$
----	----------------------	--

DANH MỤC CÁC BẢNG

Số hiệu bảng	Tên bảng	Trang
Bảng 1.1	Các bộ dữ liệu sử dụng trong thực nghiệm	27
Bảng 2.1	Các bộ dữ liệu sử dụng trong thực nghiệm khi bổ sung và loại bỏ tập đối tượng	34
Bảng 2.2	Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M	35
Bảng 2.3	Thời gian thực hiện của ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M (tính theo giây)	36
Bảng 2.4	Số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp của ba thuật toán IDS_IFW_DO, IARM-E và KGIRA-M	41
Bảng 2.5	Thời gian thực hiện của ba thuật toán: IDS_IFW_DO, IARM-E và KGIRD-M (tính theo giây)	42
Bảng 2.6(a)	Biểu diễn thông tin về các ô tô	45
Bảng 2.6(b)	Biểu diễn thông tin về các ô tô sau khi đã thay đổi giá trị.	46
Bảng 2.7	Các bộ dữ liệu sử dụng trong thực nghiệm khi tập đối tượng thay đổi giá trị	53
Bảng 2.8	Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của ba thuật toán FWIA_U_Obj, FSMV và Object-R	55
Bảng 2.9	Thời gian thực hiện của ba thuật toán FWIA_U_Obj, FSMV và Object-R (tính bằng giây)	57
Bảng 2.10	Số lượng tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Obj so với 2 thuật toán IDS_IFW_DO và IDS_IFW_AO	59

Bảng 2.11	Thời gian thực hiện của thuật toán FWIA_U_Obj so với 2 thuật toán IDS_IFW_DO và IDS_IFW_AO (tính bằng giây)	60
-----------	---	----

Bảng 3.1	Biểu diễn thông tin về các tivi	65
Bảng 3.2	Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_AA	70
Bảng 3.3	Số thuộc tính tập rút gọn và độ chính xác phân lớp của 3 thuật toán FWIA_AA, UARA và IDRA	71
Bảng 3.4	Thời gian thực hiện ba thuật toán FWIA_AA, UARA, IDRA (tính bằng giây)	73
Bảng 3.5	Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_DA	79
Bảng 3.6	Số thuộc tính tập rút gọn và độ chính xác phân lớp của hai thuật toán FWIA_DA và UARD	78
Bảng 3.7	Thời gian thực hiện hai thuật toán FWIA_DA và UARD (tính bằng giây)	81
Bảng 3.8	Biểu diễn thông tin về các tivi khi thay đổi giá trị	86
Bảng 3.9	Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_U_Attr	91
Bảng 3.10	Số thuộc tính tập rút gọn và độ chính xác phân lớp của hai thuật toán FWIA_U_Attr và Attribute-R	93
Bảng 3.11	Thời gian thực hiện hai thuật toán FWIA_U_Attr và Attribute-R (tính bằng giây)	95
Bảng 3.12	Số lượng tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Attr và 2 thuật toán FWIA_DA và FWIA_AA.	97
Bảng 3.13	Thời gian thực hiện của thuật toán FWIA_U_Attr và 2 thuật toán FWIA_DA và FWIA_AA (tính bằng giây)	98

DANH MỤC HÌNH VẼ

Số hiệu hình vẽ	Tên hình vẽ	Trang
Hình 1.1	Quá trình lựa chọn thuộc tính	15
Hình 1.2	Mô hình phương pháp tìm tập rút gọn	17
Hình 2.1(a)	Số lượng thuộc tính tập rút gọn của ba thuật toán FWIA_U_Obj, FSMV và Object-R	56
Hình 2.1(b)	Độ chính xác phân lớp của ba thuật toán FWIA_U_Obj, FSMV và Object-R	56
Hình 3.1	Sơ đồ khối của thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp loại bỏ tập thuộc tính	76
Hình 3.2(a)	Số thuộc tính tập rút gọn của hai thuật toán FWIA_DA và UARD	82
Hình 3.2(b)	Độ chính xác phân lớp của hai thuật toán FWIA_DA và UARD	82
Hình 3.3(a)	Số lượng thuộc tính tập rút gọn của hai thuật toán FWIA_U_Attr và Attribute-R	94
Hình 3.3(b)	Độ chính xác phân lớp của hai thuật toán FWIA_U_Attr và Attribute-R	94

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Ngày nay, với xu hướng phát triển của cuộc cách mạng công nghiệp lần thứ 4, việc thu thập, lưu trữ, phân tích và xử lý thông tin từ tập dữ liệu lớn là yêu cầu cấp thiết đặt ra. Các tập dữ liệu ngày càng lớn về dung lượng, phức tạp, không đầy đủ, không chắc chắn. Việc thực thi các mô hình khai phá dữ liệu ngày càng trở nên thách thức. Do đó, bài toán rút gọn thuộc tính là bài toán cấp thiết đặt ra nhằm nâng cao hiệu quả của các mô hình khai phá dữ liệu. Rút gọn thuộc tính nằm trong giai đoạn tiền xử lý dữ liệu với nhằm loại bỏ các thuộc tính dư thừa nhằm nâng cao hiệu quả các mô hình khai phá dữ liệu. Quá trình rút gọn thuộc tính có thể thực hiện bởi phương pháp rút trích (extraction) hoặc lựa chọn (selection) thuộc tính. Hai phương pháp đều có mục tiêu tối giản tập thuộc tính sao cho lượng thông tin chứa trong tập thuộc tính rút gọn bảo toàn ở mức cao nhất. Lựa chọn thuộc tính được ứng dụng rất thường xuyên trong các tác vụ phân lớp, phân cụm và hồi quy. Giải pháp thông thường nhất của trích chọn thuộc tính là sử dụng phương pháp vét cạn để tìm tập thuộc tính con tốt nhất cho mỗi mô hình phân tích dữ liệu nhất định. Hiện nhiên giải pháp này cần được cải tiến để đáp ứng tiêu chí phân tích dữ liệu hiệu quả và nhanh. Vì vậy, rất nhiều nghiên cứu đã được thực hiện và các chiến lược nhằm giảm kích thước tập thuộc tính được đề xuất cũng rất phong phú. Nói chung một chiến lược trích chọn thuộc tính thường gồm bốn bước: Tạo tập thuộc tính con; Đánh giá tập thuộc tính được tạo; Kiểm tra tiêu chuẩn dừng lựa chọn; Kiểm tra đánh giá tập rút gọn kết quả.

Rút gọn thuộc tính là bài toán quan trọng trong bước tiền xử lý dữ liệu của quá trình khai thác dữ liệu [71, 93]. Mục tiêu của việc rút gọn thuộc tính là tìm tập con của tập thuộc tính, được gọi là tập rút gọn, để nâng cao hiệu quả của mô hình khai phá dữ liệu [46]. Lý thuyết tập thô do Pawlak [61] đề xuất được xem là công cụ hiệu quả giải quyết bài toán rút gọn thuộc tính trên bảng quyết định đầy đủ, đã và đang thu hút sự quan tâm của các nhà nghiên

cứu trong suốt bốn thập kỷ qua. Trong thực tế, các bảng quyết định thường thiếu giá trị trên miền giá trị của tập thuộc tính, gọi là *bảng quyết định không đầy đủ*. Để giải quyết bài toán rút gọn thuộc tính và trích lọc luật trực tiếp trên bảng quyết định không đầy đủ mà không qua bước tiền xử lý giá trị thiếu, Kryszkiewicz[38] mở rộng quan hệ tương đương trong lý thuyết tập thô truyền thống thành quan hệ dung sai và xây dựng mô hình tập thô dung sai. Dựa trên mô hình tập thô dung sai, nhiều thuật toán rút gọn thuộc tính trong bảng quyết định không đầy đủ đã được đề xuất trên cơ sở mở rộng các kết quả nghiên cứu về rút gọn thuộc tính theo tiếp cập tập thô truyền thống. Các thuật toán điển hình có thể kể đến là: các thuật toán dựa trên miền dương [25, 54, 58], các thuật toán dựa trên hàm ma trận phân biệt [17, 57], các thuật toán dựa trên hàm ma trận phân biệt mở rộng [56], các thuật toán dựa trên tập xấp xỉ thô [14, 21], các thuật toán dựa trên entropy thông tin [26, 64, 72], các thuật toán dựa trên lượng thông tin [18, 22]; các thuật toán dựa trên độ đo khoảng cách [1, 19], thuật toán dựa trên hệ số tương quan [85], thuật toán dựa trên thuộc tính thuộc [75].

Với tốc độ phát triển nhanh chóng của dữ liệu, các bảng quyết định không đầy đủ trong các bài toán thực tế thường có kích thước rất lớn và luôn luôn thay đổi, cập nhật, khi đó bảng quyết định không đầy đủ được gọi là *bảng quyết định không đầy đủ thay đổi* (*nghĩa là dữ liệu thay đổi trong trường hợp: (i) bổ sung, loại bỏ tập đối tượng; (ii) bổ sung, loại bỏ tập thuộc tính và (iii) tập đối tượng, tập thuộc tính thay đổi giá trị*). Ví dụ, một số bảng quyết định trong dữ liệu tin sinh học có hàng triệu thuộc tính. Hơn nữa, chúng luôn được thay đổi hoặc cập nhật theo thời gian [80], đặc biệt là trong các trường hợp thay đổi thuộc tính hoặc kích thước [9].

Trường hợp các bảng quyết định không đầy đủ thay đổi, các thuật toán rút gọn thuộc tính phải tính toán lại tập rút gọn trên toàn bộ bảng quyết định sau khi thay đổi nên chi phí về thời gian tính toán tăng lên đáng kể. Trường hợp bảng quyết định có kích thước lớn, việc thực hiện thuật toán trên toàn bộ

bảng quyết định sẽ gặp khó khăn về thời gian thực hiện. Do đó, các nhà nghiên cứu đề xuất *phương pháp gia tăng* tìm tập rút gọn. Các thuật toán gia tăng có khả năng giảm thiểu thời gian thực hiện và có khả năng thực hiện trên các bảng quyết định không đầy đủ kích thước lớn bằng giải pháp chia nhỏ bảng quyết định.

Theo tiếp cận *tập thô truyền thống* và *các mô hình tập thô mở rộng*, cho đến nay nhiều thuật toán gia tăng tìm tập rút gọn đã được đề xuất dựa trên tập thô truyền thống và một số tập thô mở rộng. Các nhà nghiên cứu đã đề xuất các thuật toán gia tăng tìm tập rút gọn trong trường hợp: bổ sung và loại bỏ tập đối tượng [10, 23, 46, 52, 56, 59, 67, 68, 92], bổ sung và loại bỏ tập thuộc tính [12, 56, 59, 83], tập đối tượng thay đổi giá trị [10, 92], tập thuộc tính thay đổi giá trị [11, 36, 41]. Ngoài ra, một số công bố đề xuất các thuật toán gia tăng tìm các tập xấp xỉ trong các trường hợp: bổ sung và loại bỏ tập đối tượng [43, 51], bổ sung và loại bỏ tập thuộc tính [24], tập đối tượng thay đổi giá trị [96], tập thuộc tính thay đổi giá trị [91].

Theo tiếp cận *mô hình tập thô dung sai*, trong mấy năm gần đây một số thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ đã được đề xuất với các trường hợp: bổ sung và loại bỏ tập đối tượng [45, 66, 69, 94, 98, 99], bổ sung và loại bỏ tập thuộc tính [12, 70]. Các thuật toán gia tăng này đều theo hướng tiếp cận lọc (*filter*) truyền thống. Với cách tiếp cận này, tập rút gọn tìm được là tập thuộc tính tối thiểu bảo toàn độ đo được định nghĩa. Việc đánh giá độ chính xác phân lớp được thực hiện sau khi tìm được tập rút gọn. Nhằm giảm thiểu số thuộc tính tập rút gọn và nâng cao hiệu quả độ chính xác của mô hình phân lớp, gần đây các tác giả trong [1, 2, 7] đã đề xuất các thuật toán gia tăng tìm tập rút gọn theo tiếp cận lọc - đóng gói (*filter - wrapper*) sử dụng độ đo khoảng cách. Với cách tiếp cận này, giai đoạn lọc tìm các ứng viên của tập rút gọn. Giai đoạn đóng gói tìm tập rút gọn có độ chính xác phân lớp cao nhất. Cụ thể, các tác giả trong [7] đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung tập đối tượng. Các tác

giả trong [2] đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung tập thuộc tính. Trong [1], tác giả đã xem xét đến trường hợp bổ sung, loại bỏ tập đối tượng, tập thuộc tính và đã xây dựng các công thức gia tăng tìm khoảng cách trong các trường hợp này.

Với các bảng quyết định thay đổi, ngoài các kịch bản bổ sung, loại bỏ tập đối tượng và tập thuộc tính, kịch bản tập đối tượng, tập thuộc tính thay đổi giá trị xuất hiện phổ biến trong các bài toán thực tế do dữ liệu trên các hệ thống luôn luôn thay đổi, cập nhật, đặc biệt là trên các hệ thống trực tuyến, các hệ thống dữ liệu thay đổi theo thời gian. Với kịch bản tập đối tượng, tập thuộc tính thay đổi giá trị này, trên bảng quyết định đầy đủ, một số công trình nghiên cứu đã đề xuất các thuật toán gia tăng tìm theo tiếp cận tập thô truyền thống [35, 47, 77, 84, 92], mô hình tập thô bao phủ [10, 11, 41], mô hình tập thô mờ [96].

Trên bảng quyết định không đầy đủ, một số công trình đã công bố các thuật toán gia tăng tìm tập rút gọn trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị. Các tác giả trong [69] xây dựng công thức cập nhật miền dương trong trường hợp tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng FSMV cập nhật tập rút gọn. Các tác giả trong [86] xây dựng công thức cập nhật độ đo không nhất quán trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị, trên cơ sở đó đề xuất hai thuật toán: thuật toán Object-R cập nhật tập rút gọn trong trường hợp tập đối tượng thay đổi giá trị và thuật toán Attribute-R trong trường hợp tập thuộc tính thay đổi giá trị. Tuy nhiên, các thuật toán này (FSMV, Object-R, Attribute-R) đều theo hướng tiếp cận lọc truyền thống.

Do đó, *mục đích nghiên cứu* của luận án là nghiên cứu, đề xuất các thuật toán gia tăng tìm tập rút gọn theo hướng tiếp cận lọc - đóng gói sử dụng khoảng cách nhằm giảm thiểu số lượng thuộc tính tập rút gọn, từ đó nâng cao hiệu quả của mô hình phân lớp.

2. Mục tiêu nghiên cứu

Mục tiêu nghiên cứu của luận án tập trung nghiên cứu hai vấn đề chính:

1) *Thứ nhất*: Nghiên cứu tập đối tượng thay đổi

- Nghiên cứu các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập đối tượng.

- Nghiên cứu, đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi trong trường hợp *tập đối tượng thay đổi giá trị*.

Các thuật toán nghiên cứu, đề xuất nhằm mục tiêu giảm thiểu số lượng thuộc tính tập rút gọn và cải thiện độ chính xác phân lớp, từ đó nâng cao hiệu quả mô hình phân lớp.

Trong trường hợp tập đối tượng thay đổi giá trị, luận án so sánh hướng tiếp cận rút gọn thuộc tính trực tiếp với hướng tiếp cận gián tiếp thực hiện đồng thời khi loại bỏ sau đó bổ sung tập đối tượng.

2) *Thứ hai*: Nghiên cứu tập thuộc tính thay đổi

- Nghiên cứu, xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập thuộc tính.

- Nghiên cứu, đề xuất các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi trong trường hợp *tập thuộc tính thay đổi giá trị*.

Các thuật toán nghiên cứu, đề xuất nhằm mục tiêu giảm thiểu số lượng thuộc tính tập rút gọn và cải thiện độ chính xác phân lớp, từ đó nâng cao hiệu quả mô hình phân lớp.

Trong trường hợp tập thuộc tính thay đổi giá trị, luận án so sánh hướng tiếp cận rút gọn thuộc tính trực tiếp với hướng tiếp cận gián tiếp thực hiện đồng thời khi loại bỏ sau đó bổ sung tập thuộc tính.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của luận án là các bảng quyết định không đầy đủ thay đổi trong trường hợp bổ sung, loại bỏ tập đối tượng, tập thuộc tính và tập đối tượng, tập thuộc tính thay đổi giá trị.

Phạm vi nghiên cứu của luận án là các phương pháp rút gọn thuộc tính của bảng quyết định không đầy đủ theo tiếp cận tập thô dung sai. Rút gọn thuộc tính cho bài toán phân lớp dữ liệu.

4. Phương pháp nghiên cứu

Phương pháp nghiên cứu của luận án là nghiên cứu lý thuyết và nghiên cứu thực nghiệm.

1) *Nghiên cứu lý thuyết:* Nghiên cứu các thuật toán rút gọn thuộc tính theo tiếp cận tập thô đã công bố, phân tích ưu điểm, nhược điểm và các vấn đề còn tồn tại của các nghiên cứu liên quan. Trên cơ sở đó, đề xuất các độ đo cải tiến và các thuật toán theo hướng tiếp cận lai ghép lọc - đóng gói. Các đề xuất, cải tiến được chứng minh chặt chẽ về lý thuyết bởi các định lý, mệnh đề.

2) *Nghiên cứu thực nghiệm:* Các thuật toán đề xuất được cài đặt, chạy thực nghiệm, so sánh, đánh giá với các thuật toán khác trên các bộ số liệu mẫu từ kho dữ liệu UCI nhằm minh chứng về tính hiệu quả của các nghiên cứu về lý thuyết.

5. Nội dung nghiên cứu

1) Nghiên cứu các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi trong trường hợp bổ sung, loại bỏ tập đối tượng, tập thuộc tính và tập đối tượng, tập thuộc tính thay đổi giá trị.

2) Thực nghiệm, cài đặt, so sánh, đánh giá các thuật toán đề xuất với các thuật toán khác đã công bố trên cùng môi trường thực nghiệm, cùng các bộ số liệu mẫu từ kho dữ liệu UCI.

6. Ý nghĩa khoa học và thực tiễn

Kết quả nghiên cứu của luận án cung cấp thêm cơ sở khoa học giúp các nghiên cứu toàn diện về tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi trong tất cả các trường hợp về tập đối tượng, tập thuộc tính thay đổi.

Với mục tiêu đặt ra, luận án đạt được 03 kết quả chính như sau:

1) Xây dựng công thức gia tăng cập nhật khoảng cách trong các trường hợp bổ sung, loại bỏ tập thuộc tính, trên cơ sở đó xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trên bảng quyết định không đầy đủ trong trường hợp *bổ sung, loại bỏ tập thuộc tính*.

2) Đề xuất công thức gia tăng cập nhật khoảng cách khi tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp *tập đối tượng thay đổi giá trị*.

3) Đề xuất công thức gia tăng cập nhật khoảng cách khi tập thuộc tính thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp *tập thuộc tính thay đổi giá trị*.

7. Bố cục của luận án

Bố cục của luận án gồm phần mở đầu và ba chương nội dung, phần kết luận và danh mục các tài liệu tham khảo. *Chương 1* trình bày các khái niệm cơ bản về mô hình tập thô truyền thống, mô hình tập thô dung sai và tổng quan về rút gọn thuộc tính theo tiếp cận tập thô dung sai; các nghiên cứu liên quan. Từ đó, phân tích các vấn đề còn tồn tại và nêu rõ các mục tiêu nghiên cứu cùng với tóm tắt các kết quả đạt được. *Chương 2* trình bày về nghiên cứu về tập đối tượng thay đổi trong trường hợp bổ sung, loại bỏ tập đối tượng và tập đối tượng thay đổi giá trị. *Chương 3* trình bày về nghiên cứu về tập đối tượng thay đổi trong trường hợp bổ sung, loại bỏ tập thuộc tính và tập thuộc

tính thay đổi giá trị. Cuối cùng, phần kết luận nêu những đóng góp của luận án, hướng phát triển và những vấn đề quan tâm của tác giả.

CHƯƠNG 1. TỔNG QUAN VỀ HỆ THÔNG TIN VÀ PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH THEO TIẾP CẬN TẬP THÔ DUNG SAI

1.1. Mở đầu

Chương này trình bày một số khái niệm cơ bản về lý thuyết tập thô, mô hình tập thô truyền thông trên hệ thông tin đầy đủ, mô hình tập thô dung sai trên hệ thông tin không đầy đủ. Chương 1 cũng trình bày tổng quan về hướng tiếp cận lọc, tiếp cận lọc - đóng gói trong rút gọn thuộc tính, các nghiên cứu liên quan đến rút gọn thuộc tính theo tiếp cận tập thô dung sai, các nghiên cứu liên quan đến các phương pháp gia tăng rút gọn thuộc tính theo tiếp cận tập thô dung sai. Trên cơ sở đó, chương 1 phân tích các vấn đề còn tồn tại của các nghiên cứu trước đây, từ đó đưa ra các mục tiêu nghiên cứu của luận án.

1.2. Các khái niệm cơ bản về hệ thông tin

1.2.1. Hệ thông tin đầy đủ và mô hình tập thô truyền thông

1.2.1.1- Hệ thông tin đầy đủ

Hệ thông tin là công cụ biểu diễn tri thức dưới dạng một bảng dữ liệu gồm p cột tương ứng với p thuộc tính và n hàng tương ứng với n đối tượng. Hệ thông tin được định nghĩa như sau:

Hệ thông tin là một bộ tứ $IS = (U, A, V, f)$, trong đó:

(1) U là tập hữu hạn, khác rỗng các đối tượng;

(2) A là tập hữu hạn, khác rỗng các thuộc tính;

$$(3) \quad \begin{array}{c} \bigcup \\ V \\ = \end{array} \quad \begin{array}{c} \text{với } V_a \\ \text{a} \in A \end{array} \quad \begin{array}{l} \text{là tập giá trị của} \\ \text{thuộc tính} \end{array} \quad \begin{array}{l} a \in A; \end{array}$$

$$(4) \quad f: U \times A \rightarrow V_a \quad \begin{array}{l} \text{là hàm thông tin, } \forall a \in A, u \in U, \end{array}$$

$f(u, a) \in V_a$. Với mọi $u \in U, a \in A$, ta ký hiệu thay vì $f(u, a)$.

giá trị thuộc tính a tại
đối tượng u là $a(u)$

Xét hệ thông tin $IS = (U, A, V, f)$, mỗi tập $P \subseteq A$ xác định
con các thuộc tính

một quan hệ hai ngôi trên U , ký hiệu $IND(P)$, được xác định như sau:
là

$$IND(P) = \{a(u) = a(v)\} \quad (1.1)$$

$$\{(u, v) \in U$$

$$\times_U \forall a \in \\ P,$$

Khi đó $IND(P)$ là quan hệ P -không phân biệt được.

Dễ thấy rằng $IND(P)$ là một quan hệ tương đương trên U . Nếu $(u, v) \in IND(P)$ thì hai đối tượng u và v không phân biệt được bởi các thuộc tính trong P .

Quan hệ tương đương $IND(P)$ xác định một phân hoạch trên U , ký hiệu là $U / IND(P)$ hay U / P . Ký hiệu lớp tương đương trong phân hoạch U / P chứa đối

, khi đó:
tượng u là $[u]_P$

$$[u]_P = \{v \in U \mid (u, v) \in IND(P)\}.$$

1.2.1.2. Mô hình tập thô truyền thông

Cho hệ thông tin $IS = (U, A, V, f)$ và tập đối tượng $X \subseteq U$. Với một tập thuộc tính $B \subseteq A$ cho trước, chúng ta có các lớp tương đương của phân hoạch U / B , thế thì một tập đối tượng X có thể biểu diễn thông qua các lớp tương đương này như thế nào?

Trong lý thuyết tập thô, để biểu diễn X thông qua các lớp tương đương của U / B người ta xấp xỉ X bởi hợp của một số hữu hạn các lớp tương đương

của U / B . Có hai cách xấp xỉ tập đối tượng X thông qua tập thuộc tính B , được gọi là B -xấp xỉ dưới và B -xấp xỉ trên của X , ký hiệu là lượt là \underline{BX} và BX , được

xác định như sau:

$$\underline{BX} = \left\{ u \in U \mid [u]_B \subseteq X \right\}, \quad BX = \left\{ u \in U \mid [u]_B \cap X \neq \emptyset \right\}$$

$\emptyset \}$

Tập \underline{BX} bao gồm tất cả các phần tử của U chắc chắn thuộc vào X , còn tập BX bao gồm các phần tử của U có thể thuộc vào X dựa trên tập thuộc tính B . Với tập X cho trước, tập xấp xỉ dưới \underline{BX} và xấp xỉ trên BX luôn đi cùng nhau và được sử dụng để xấp xỉ tập hợp trong các bài toán cụ thể.

Từ hai tập xấp xỉ nêu trên, ta định nghĩa các tập:

$$BN_B(X) = BX - \underline{BX}: \text{B-miền biên của } X, \quad U - BX: \text{B-miền ngoài của } X.$$

B-miền biên của X là tập chứa các đối tượng có thể thuộc hoặc không thuộc X , còn B-miền ngoài của X chứa các đối tượng chắc chắn không thuộc X .

U Sử dụng các lớp của phân hoạch U/B , các xấp xỉ dưới và trên của X có
thể viết lại: $\underline{BX} = \bigcup_{Y \subseteq B} Y \subseteq X$ = $B \cap X \neq \emptyset\}$.

thì X được gọi là **tập chính**
Trong trường hợp $BN_B(X) = \emptyset$

$$\Leftrightarrow BX = X = \underline{BX}$$

xác (exact set), ngược lại X được gọi là **tập thô** (rough set).

Với $B \subseteq A$, ta gọi B -miền dương của D là tập được xác định như sau:

U $POS_B(D) = \bigcup_{v \in U} v \cap D$ là tập tất cả các đối tượng u sao cho với mọi $v \in U$ mà

Rõ ràng $POS_B(D)$

$u(B) = v(B)$ ta đều có $u(D) = v(D)$. $POS_B(D) = \{u \in U \mid [u] \subseteq [u]_D\}$
Nói cách khác,

1.2.1.3. Bảng quyết định và tập rút gọn

Một lớp đặc biệt của các hệ thông tin có vai trò quan trọng trong nhiều ứng dụng là bảng quyết định. Bảng quyết định với tập thuộc tính A được chia thành hai tập khác rỗng rời nhau C và D , lần lượt được gọi là tập thuộc tính điều kiện và thuộc tính quyết định, nghĩa là $DS = (U, C \cup D)$ với $C \cap D = \emptyset$.

Trong bảng quyết định, các thuộc tính điều kiện được phân thành thuộc

tính lõi và thuộc tính không cần thiết. Thuộc tính lõi là thuộc tính cốt yếu, là thuộc tính có trong tất cả các tập rút gọn của bảng quyết định và dùng để xây dựng tập rút gọn, mà tập rút gọn liên quan đến phân lớp. Thuộc tính không cần thiết là thuộc tính dư thừa mà việc loại bỏ thuộc tính này không ảnh hưởng đến việc phân lớp dữ liệu. Các thuộc tính không cần thiết được phân thành hai nhóm: Thuộc tính dư thừa thực sự và thuộc tính rút gọn. Thuộc tính dư thừa thực sự là những thuộc tính dư thừa mà việc loại bỏ tất cả các thuộc tính như vậy không ảnh hưởng đến việc phân lớp dữ liệu. Thuộc tính rút gọn, với một tổ hợp thuộc tính nào đó, nó là thuộc tính dư thừa và với một tổ hợp các thuộc tính khác nó có thể là thuộc tính lõi.

Định nghĩa 1.1 [62] (*Độ quan trọng của thuộc tính dựa trên miền dương*)

Cho bảng quyết định tính a được $DS = (U, C \cup D)$, $P \subseteq C$, $a \in P$, độ quan trọng xác định:

$$sig(a, P) = POS_P(D) - POS_{P-\{a\}}(D)$$

$$\begin{matrix} \\ \\ \vdots \\ 2 \end{matrix}$$

Nếu $sig(a, P) > 0$ thì thuộc tính a được gọi là thuộc tính cần thiết. Nếu

$sig(a, P) = 0$ thì thuộc tính a được gọi là thuộc tính không cần thiết (dư thừa).

Định nghĩa 1.2 [62] (*Tập rút gọn dựa trên miền dương*)

Cho bảng quyết định $DS = (U, C \cup D)$. Tập $R \subseteq C$ thỏa mãn các điều kiện:
định

1) $POS_R(D) = POS_C(D)$

2) $\forall r \in R, POS_{R-\{r\}}(D) \neq POS_C(D)$
 $POS_C(D)$ hoặc

thì R là một *tập rút gọn* của C dựa trên miền dương.

Trong định nghĩa này, điều kiện 1) là điều kiện tập rút gọn R bảo toàn độ chắc chắn của các luật phân lớp như tập thuộc tính gốc C ; điều kiện 2) đảm bảo để trong tập rút gọn R không chứa thuộc tính nào dư thừa.

Tập rút gọn định nghĩa như trên còn được gọi là tập rút gọn Pawlak.

Trong một bảng quyết định có thể có nhiều tập rút gọn, ký hiệu là họ

tất cả các tập rút gọn Pawlak của C . Tập tất cả các thuộc tính cần thiết trong DS được gọi là tập lõi dựa trên miền dương và được ký hiệu là

$$PCORE \cap$$

$$(C) =$$

$$R$$

$$_{R \in PRED(C)}$$

Định nghĩa 1.3 [62] (*Thuộc tính rút gọn dựa trên miền dương*)

Cho bảng quyết định $DS = (U, C \cup D)$, với $a \in C$ ta nói rằng a là thuộc tính rút gọn của DS nếu tồn tại một tập rút gọn

$R \in PRED(C)$ sao cho $a \in R$.

1.2.2. Hệ thông tin không đầy đủ và mô hình tập thô dung sai

Nhằm giải quyết bài toán rút gọn thuộc tính trên các hệ quyết định không đầy đủ, Marzena Kryszkiewicz[38] đã mở rộng quan hệ tương đương trong lý thuyết tập thô truyền thống thành quan hệ dung sai và xây dựng mô hình tập thô mở rộng dựa trên quan hệ dung sai gọi là mô hình tập thô dung sai.

1.2.2.1. Hệ thông tin không đầy đủ

Cho $IS = (U, A, V, f)$ và $a \in A$
 hệ thông tin, nếu tồn tại $u \in U$
 sao cho $a(u)$

thiểu giá trị thì IS được gọi là hệ thông tin không đầy đủ. Ta biểu diễn giá trị thiểu là '*' và hệ thông tin không đầy đủ là $IIS = (U, A, V, f)$. Xét hệ thông tin không

đầy đủ $IIS = (U, A, V, f)$ với tập thuộc tính $P \subseteq A$, ta
 định nghĩa một quan hệ nhị

phân trên U
 như sau: $\forall a \in P, a(u) = a(v) \vee a(u) = '*' \vee a(v) = '*' \}$

(1.3)

$SIM(P) = \{(u, v) \in U \times U$
 Quan hệ $SIM(P)$ không phải là quan hệ tương
 đương (vì chúng có tính phản

xq, đối xứng nhưng không có tính bắc $SIM(P)$ được gọi là quan
 cầu). Quan hệ

\cap
 hệ dung sai (tolerance relation) trên U . Theo [38],

$$SIM(P) = \bigcup_{a \in P} SIM(\{a\}).$$

Đặt $S_P(u) = \left\{ v \mid (u, v) \in SIM(P) \text{ khi đó } S_P(v) \right\}$ được gọi là một lớp

dung sai. $S_P(u)$

là tập lớn nhất các đối tượng không có khả năng phân biệt với

không có khả năng phân biệt với u, hay u
u trên tập thuộc tính P (tức là $\forall v \in U$
và v có quan hệ dung sai với nhau).

Ký hiệu tập tất cả các lớp dung sai sinh bởi quan hệ $SIM(P)$ trên U là
 không phải là một phân
 $U / SIM(P)$, khi đó các lớp dung sai

trong $U / SIM(P)$

hoạch của U mà hình thành một phủ của U vì chúng có thể giao nhau và

$$\bigcup_{u \in U} S_P(u) = U.$$

Tập tất cả các phủ của U sinh bởi được ký
các tập con thuộc tính $P \subseteq A$

hiệu là $COVER(U)$

Các tập P -xấp xỉ dưới và P -xấp xỉ trên của X trong hệ thông tin không
đầy đủ, ký hiệu lần lượt là $\underline{P}X$ và PX , được xác định như sau:

$$\underline{P}X = \left\{ u \in U \mid S_P(u) \subseteq X \right\} \text{ và } PX = \left\{ u \in U \mid S_P(u) \cap X \neq \emptyset \right\}$$

Với các tập xấp xỉ nêu trên, ta gọi P -miền biên của X là tập
 $BN_P(X) = PX - \underline{P}X$, và P -miền ngoài của X là tập $U - \underline{P}X$.

1.2.2.2. Bảng quyết định không đầy đủ

Cho bảng quyết định $DS = (U, C \cup D)$, nếu tồn tại $c \in C$ sao cho
định $\underset{u \in U}{c(u)}$

thiếu giá trị thì DS được gọi là bảng quyết
định không đầy đủ. Ta biểu
 $c(u)$

diễn giá trị thiếu là '*' và bảng quyết $IDS = (U, C \cup D)$
định không đầy đủ là

với $\forall d \in D, * \notin V_d$. Theo [38] thì $D = \{d\}$ tức là D chỉ gồm một thuộc tính quyết
định duy nhất, khi đó bảng quyết định $IDS = (U, C \cup \{d\})$
không đầy đủ ký hiệu

Định nghĩa 1.4 [38] Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$

với $U = \{u_1, u_2, \dots, u_n\}$ và $P \subseteq C$. Khi đó, $SIM(P)$, ma trận dung sai của quan hệ

ký hiệu

$M(P) = [\lfloor p_{ij} \rfloor]$, được định nghĩa như sau:

$$M(P) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix} \quad (1.4)$$

trong đó $p_{ij} \in \{0, 1\}$. $p_{ij} = 1$ nếu $u_j \in S_P$ $p_{ij} = 0$ nếu $u_j \notin S_P$ (u_i) và (u_i) với $i, j = 1..n$

Với việc biểu diễn $SIM(P)$ quan hệ dung sai

bằng ma trận dung sai $M(P)$,

ta có mọi $u_i \in U$, $S_P(u_i) = \{u_j \in U | p_{ij} = 1\}$ và $S_P(u_i) = \sum_{j=1}^n p_{ij}$.

$j=1$

Với $P, Q \subseteq C, u \in U$

ta có

$$S_{P \cup Q}(u) = S_P \quad M(P) = \lfloor p_{ij} \rfloor,$$

$$(u) \cap S_Q(u).$$

Giả

sử

$M(Q) = \lfloor q_{ij} \rfloor_{n \times n}$ là hai ma trận dung sai của $SIM(P)$, $SIM(Q)$, khi đó ma trận

dung sai trên tập
thuộc tính $S = P \cup Q$ là: với

$$s_{ij} = p_{ij} \cdot q_{ij}$$

$$M(S)$$

=

$$M$$

$$(P$$

\cup

$$Q)$$

=

$$\lfloor s_{ij}$$

$$\rfloor_{n \times n}$$

Xét bảng $IDS = (U, C \cup D)$ với
quyết định không đầy đủ

$$U = \{u_1, u_2, \dots, u_n\},$$

$P \subseteq C, X \subseteq U$. Giả sử tập đối tượng X được biểu diễn bằng véc tơ một chiều

$X = (x_1, x_2, \dots, x_n)$ với $x_i = 1$ nếu $u_i \in X$ và $x_i = 0$ nếu

$u_i \notin X$. Khi đó,

$$\underline{P}X = \left\{ u_i \in U \mid p_{ij} \leq x_j, j = 1..n \right\} \text{ và } P^X = \left\{ u_i \in U \mid p_{ij} \cdot x_j \neq \emptyset, j = 1..n \right\}.$$

1.2.2.3. Tập rút gọn của bảng quyết định không đầy đủ

Trong [38], Marzena Kryszkiewicz định nghĩa tập rút gọn của bảng quyết định không đầy đủ, là tập con tối thiểu của tập thuộc tính điều kiện mà bảo toàn hàm quyết định suy rộng của tất cả các đối tượng.

Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\}, V, f)$. Với $B \subseteq C$, $u \in U$ gọi là hàm quyết định suy rộng, Theo [38], nếu thì IDS là *nhất quán*, trái lại $|\partial_B(u)| = 1$ với mọi $u \in U$ IDS là *không nhất quán*.

Định nghĩa 1.5. [38] Cho bảng quyết định không đầy đủ $IDS = (U, C \cup D, V, f)$

Tập thuộc tính $R \subseteq C$ thỏa mãn các điều kiện:

1) $\partial_R(u) = \partial_C(u)$ với mọi $u \in U$

2) với mọi $\forall R' \subset R$, tồn tại $u \in U$ $\partial_{R'}(u) \neq \partial_C(u)$ sao cho

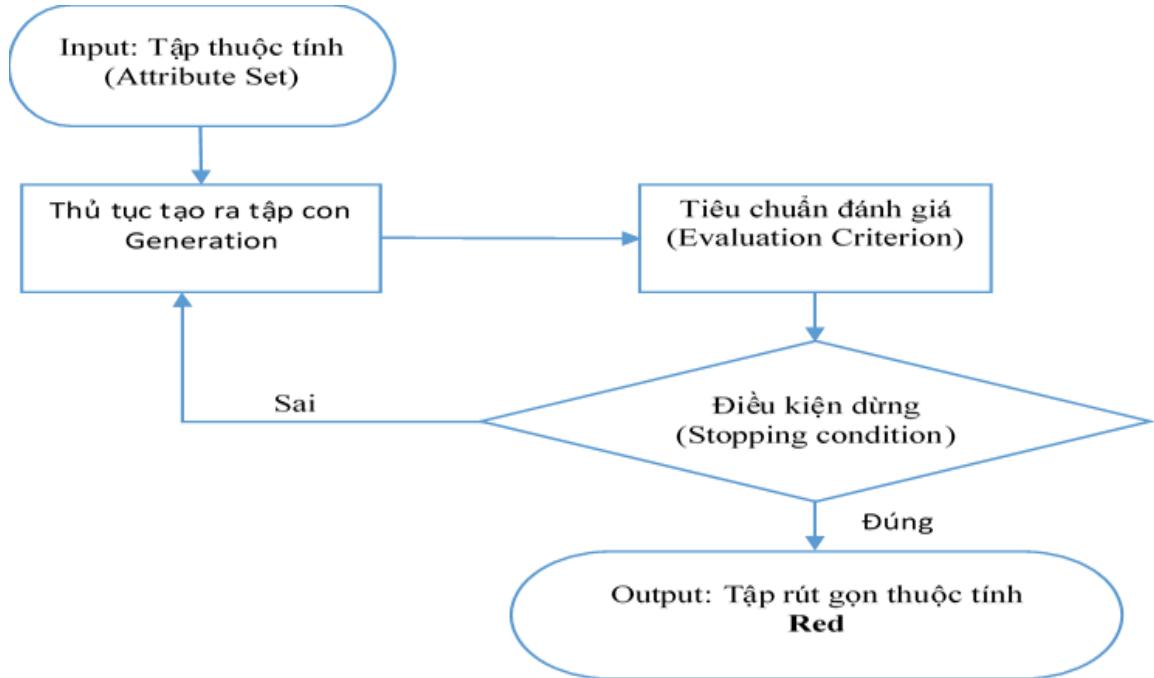
thì R được gọi là một tập rút gọn của C .

Tập rút gọn định nghĩa như trên còn gọi là tập rút gọn Kryszkiewicz. ■

1.3. Phương pháp rút gọn thuộc tính theo tiếp cận tập thô dung sai

1.3.1. Phương pháp rút gọn thuộc tính theo tiếp cận lọc

Rút gọn thuộc tính dựa vào lý thuyết tập thô là một quá trình chọn lựa tập con của tập thuộc tính có số thuộc tính tối thiểu nhưng lượng thông tin hàm chứa tối đa gần như tập toàn bộ thuộc tính ban đầu. Để thiết kế một thuật toán rút gọn thuộc tính quá trình rút gọn thuộc tính dựa vào lý thuyết tập thô được mô tả trong sơ đồ khái [66] dưới đây:



Hình 1.1. Quá trình lựa chọn thuộc tính

Trong sơ đồ có 3 yếu tố cơ bản sau đây:

1- *Thủ tục tạo ra tập con (Generation)*: Để tạo ra các tập con ứng viên để đánh giá. Tạo lập tập con thuộc tính là quá trình tìm kiếm liên tiếp nhằm tạo ra các tập con để đánh giá, lựa chọn.

2- *Tiêu chuẩn đánh giá*: Để đánh giá tập con ứng viên. Tiêu chuẩn đánh giá tính toán phù hợp với tập con thuộc tính được tạo bởi thủ tục Generation.

3- *Điều kiện dừng*: Kiểm tra tiêu chuẩn dừng lựa chọn; Kiểm tra đánh giá tập rút gọn kết quả.

Tạo lập tập con thuộc tính là quá trình tìm kiếm liên tiếp nhằm tạo ra các tập con để đánh giá, lựa chọn. Giả sử có M thuộc tính trong tập dữ liệu ban

dầu,

khi đó số tất cả các tập con từ M thuộc tính sẽ là 2^M . Với số ứng viên này, việc tìm tập con tối ưu, ngay cả khi M không lớn lắm, cũng là một việc không thể. Vì vậy, phương pháp chung để tìm tập con thuộc tính tối ưu là lần lượt tạo ra các tập con để so sánh. Mỗi tập con sinh ra bởi một thủ tục sẽ được đánh giá theo một tiêu chuẩn nhất định và đem so sánh với tập con tốt nhất trước đó. Nếu tập con này tốt hơn, nó sẽ thay thế tập cũ.

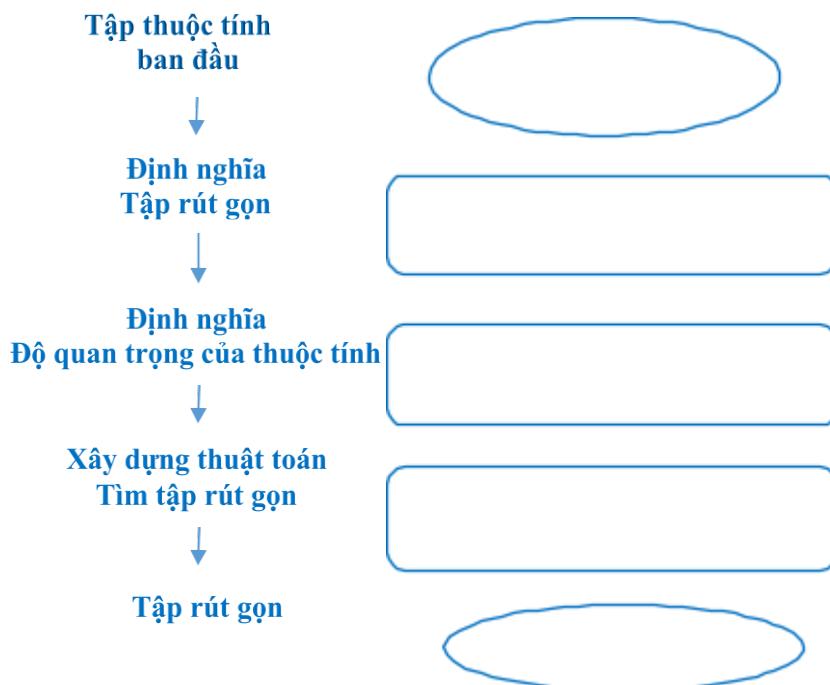
Quá trình tìm kiếm tập con thuộc tính tối ưu sẽ dừng khi một trong bốn điều kiện sau xảy ra: (a) Đã thu được số thuộc tính quy định, (b) Số bước lặp quy định cho quá trình lựa chọn đã hết, (c) Việc thêm vào hay loại bỏ một thuộc tính nào đó không cho một tập con tốt hơn, (d) Đã thu được tập con tối ưu theo tiêu chuẩn đánh giá.

Tập con tốt nhất cuối cùng phải được kiểm chứng thông qua việc tiến hành các phép kiểm định, so sánh các kết quả khai phá với tập thuộc tính “tốt nhất” này và tập thuộc tính ban đầu trên các tập dữ liệu thực hoặc nhân tạo khác nhau.

Từ sơ đồ trên, có thể thấy rằng các tiêu chuẩn đánh giá được sử dụng để đánh giá chất lượng của các thuộc tính ứng cử viên là một thành phần quan trọng, đã có một số lượng lớn các tiêu chuẩn đánh giá được thiết kế dựa trên lý thuyết tập thô và các tiêu chí khác để chọn thuộc tính ứng viên tốt nhất.

Theo lý thuyết tập thô [62], Pawlak đưa ra khái niệm tập rút gọn và xây dựng thuật toán tìm một tập rút gọn tốt nhất của bảng quyết định dựa trên tiêu chí đánh giá là độ quan trọng của thuộc tính. Phương pháp tìm một tập rút gọn tốt nhất bao gồm các bước: Định nghĩa tập rút gọn, định nghĩa độ quan trọng của thuộc tính và sau đó xây dựng thuật toán tìm một tập rút gọn.

Phương pháp rút gọn thuộc tính được mô hình hóa như sau [62]:



Hình 1.2-Mô hình phương pháp tìm tập rút gọn

Các thuật toán tìm tập rút gọn thường được xây dựng theo hai hướng tiếp cận khác nhau [62]: *Hướng tiếp cận từ dưới lên*: Xuất phát từ tập rỗng hoặc tập lõi, thêm dần các thuộc tính có độ quan trọng lớn nhất cho đến khi thu được tập rút gọn. Kiểm tra tính tối thiểu của tập rút gọn thu được; *Hướng tiếp cận từ trên xuống*: Xuất phát từ tập thuộc tính ban đầu, loại bỏ thuộc tính có độ quan trọng nhỏ nhất cho đến khi thu được tập rút gọn, kiểm tra tính tối thiểu của tập rút gọn thu được.

Tiêu chuẩn so sánh, đánh giá các phương pháp là số lượng thuộc tính của tập rút gọn, độ phức tạp của thuật toán tìm tập rút gọn và độ chính xác phân lớp của tập dữ liệu sau khi rút gọn.

1.3.2. Phương pháp rút gọn thuộc tính theo tiếp cận lai ghép lọc - đóng gói

Hiện nay, có hai cách tiếp cận chính đối với bài toán rút gọn thuộc tính đó là tiếp cận *lọc* và tiếp cận *đóng gói* [33]. Mỗi cách tiếp cận có những mục tiêu riêng về giảm thiểu số lượng thuộc tính hay nâng cao độ chính xác.

- Tiếp cận lọc: Cách tiếp cận lọc thực hiện việc rút gọn thuộc tính độc lập với thuật khai phá dữ liệu sử dụng sau này. Các thuộc tính được chọn chỉ dựa trên độ quan trọng của chúng trong việc mô tả dữ liệu.

- Tiếp cận đóng gói: Ngược lại với cách tiếp cận lọc, cách tiếp cận đóng gói tiến hành việc lựa chọn bằng cách áp dụng ngay thuật khai phá, độ chính xác của kết quả được lấy làm tiêu chuẩn để lựa chọn các tập con thuộc tính.

- *Rút gọn thuộc tính theo tiếp cận lai ghép lọc - đóng gói*: Kết hợp các ưu điểm của cả hai cách tiếp cận lọc và đóng gói [100] để tìm tập rút gọn tối ưu về số thuộc tính tối thiểu và độ chính xác phân lớp cao nhất.

Giai đoạn lọc chỉ thực hiện nhiệm vụ tìm các ứng viên của tập rút gọn, từ đó giai đoạn đóng gói thực hiện chạy bộ phân lớp và chọn trong các ứng viên có độ chính xác phân lớp cao nhất làm tập rút gọn.

1.3.3. Bài toán phân lớp trong khai phá dữ liệu

1.3.3.1. Phân lớp

Phân lớp là một hình thức học được giám sát tức là: tập dữ liệu huấn luyện (quan sát, thẩm định) đi đôi với những nhãn chỉ định lớp quan sát, những dữ liệu mới được phân lớp dựa trên tập huấn luyện. Mục đích của khai phá dữ liệu nhằm phát hiện các tri thức mà mỗi tri thức được khai phá đó sẽ được mô tả bằng các mẫu dữ liệu. Sự phân lớp là quá trình quan trọng trong khai phá dữ liệu, nó chính là việc tìm những đặc tính của đối tượng, nhằm mô tả một cách rõ ràng phạm trù mà các đối tượng đó thuộc về một lớp nào đó [81]. Quá trình phân lớp gồm có 02 tiến trình:

1. Xây dựng mô hình: với tập các lớp đã được định nghĩa trước, mỗi bộ mẫu phải được quyết định để thừa nhận vào một nhãn lớp. Tập các bộ dùng cho việc xây dựng mô hình gọi là tập dữ liệu huấn luyện, tập huấn luyện có thể được lấy ngẫu nhiên từ các cơ sở dữ liệu nghiệp vụ được lưu trữ.

2. Sử dụng mô hình: ước lượng độ chính xác của mô hình. Dùng một tập dữ liệu kiểm tra có nhãn lớp được xác định hoàn toàn độc lập với tập dữ liệu huấn luyện để đánh giá độ chính xác của mô hình. Khi độ chính xác của mô hình được chấp nhận, ta sẽ dùng mô hình để phân lớp các bộ hoặc các đối tượng trong tương lai mà nhãn lớp của nó chưa được xác định từ tập dữ liệu chưa biết.

1.3.3.2. Sinh luật quyết định trên tập rút gọn của bảng quyết định

Rút trích và đánh giá hiệu năng tập luật quyết định từ bảng quyết định là bước tiếp theo của rút gọn thuộc tính trong quá trình khai phá dữ liệu sử dụng lý thuyết tập thô. Qian Y. và các cộng sự [63] đã đề xuất ba độ đo mới nhằm khắc phục các nhược điểm của các độ đo cổ điển, đó là *độ chắc chắn*, *độ nhất quán* và *độ hỗ trợ* để đánh giá hiệu năng tập luật quyết định của bảng quyết định (gọi tắt là các độ đo đánh giá hiệu năng tập luật quyết định).

a) Luật quyết định và các độ đo cổ điển

Cho bảng quyết định $DS = (U, C \cup D)$, giả sử $U / C = \{X_1, X_2, \dots, X_m\}$ và $U / D = \{Y_1, Y_2, \dots, Y_n\}$. Với $X_i \in U / C$, $Y_j \in U / D$ và $X_i \cap Y_j \neq \emptyset$, ký hiệu $des(X_i)$ và $des(Y_j)$

lần lượt là các mô tả X_i và Y_j của các lớp tương đương trong bảng quyết định DS .

Một luật quyết định có dạng $Z_{ij} : des(X_i) \rightarrow des(Y_j)$.

Các độ đo đánh giá luật quyết định đơn Z_{ij} được đề xuất trong [63].

(1) Độ xác chắn: $\mu(Z_{ij}) = \frac{|X_i \cap Y_j|}{|X_i|}$,

(2) Độ hỗ trợ: $s(Z_{ij}) = \frac{|X_i \cap Y_j|}{|U|}$.

Các độ đo này chỉ sử dụng để đánh giá cho các luật quyết định đơn, không phù hợp cho việc đánh giá hiệu năng tập luật quyết định.

Độ chính xác của phân lớp: Giả sử $F = U / D = \{Y_1, Y_2, \dots, Y_n\}$ là một phân

hoạch của U theo D . Độ chính xác của phân lớp F bởi C , ký hiệu là $\alpha_C(F)$, được Pawlak [62] định nghĩa như sau:

$$\alpha_c(F) =$$

và độ nhất quán (hay độ phụ thuộc) $\gamma_c(D)$ được Pawlak [62] định nghĩa như sau:

$$\gamma_c(D) =$$

Trong một số trường hợp, $\alpha_c(F)$ được dùng để đo độ chắc chắn của bảng quyết định. Tuy nhiên, nhược điểm của độ đo này được Qian Y. và các cộng

sự chỉ ra trong [63].

cũng không biểu diễn tốt

Hơn nữa, độ nhất quán $\gamma_c(D)$

tính nhất quán của bảng quyết định vì chỉ xem xét các giá trị xấp xỉ dưới.

b) Các độ đo đánh giá hiệu năng tập luật quyết định

Nhằm khắc phục nhược điểm các độ đo cỗ điển, Qian Y. và cộng sự [63] đã đề xuất ba độ đo đánh giá hiệu năng tập luật quyết định: *độ chắc chắn* (certainty measure), *độ nhất quán* (consistency measure) và *độ hỗ trợ* (support measure).

Cho bảng quyết định $DS = (U, C \cup D)$ và $RULE = \{Z_{ij} : des(X_i) \rightarrow des(Y_j)\}$ với

$X_i \in U / C, Y_j \in U / D, i = 1..m, j = 1..n$.

Độ chắc chắn α của DS được định nghĩa như sau:

$$\alpha(DS) = \sum_{i=1}^m s(Z_{ij})^{\mu}(Z_{ij}) = \sum_{j=1}^n \cdot$$

 $i=1$

Độ nhất quán β của DS được định nghĩa như sau:

$$\beta(DS) = \sum \left[1 - \frac{4}{N_i} X_i Y_j \right] \mu(Z) (1 - \mu(Z))$$

$$\prod_{i=1}^{N_i} \prod_{j=1}^{N_i} \square_{ij}$$

với N_i là số luật quyết định sinh X_i .
bởi lớp tương đương

Độ hỗ trợ γ của DS được định nghĩa như sau:

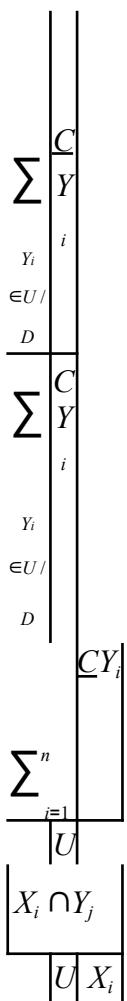
$$\gamma(DS) = \sum_{i=1}^m s^2(Z_{ij}) = \sum_{j=1}^n \prod_{i=1}^{N_i} \square_{ij} X_i \cap Y_j$$

 $i=1$

c) Công thức tính độ chính xác (accuracy)

Cách đánh giá này đơn giản tính tỉ lệ giữa số mẫu dự đoán đúng và tổng số mẫu trong tập dữ liệu. Công thức:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{số lượng mẫu}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$



Trong đó:

- Predicted (giá trị dự đoán): là kết quả dự đoán của mô hình
- Actual (giá trị thực): thu được bằng cách quan sát hoặc đo lường thực tế dữ liệu (luôn luôn đúng)
 - P (Positive) và N (Negative)
 - TP (True Positive): giá trị actual và predicted đều là positive
 - FP (False Positive): giá trị actual là negative nhưng predicted là positive
 - TN (True Negative): giá trị actual và predicted đều là negative
 - FN (False Negative): actual là negative nhưng predict là positive

Giả sử độ accuracy = 90% có nghĩa là trong số 100 mẫu thì có 90 mẫu được phân loại chính xác. Tuy nhiên đối với tập dữ liệu kiểm thử không cân bằng (nghĩa là số positive lớn hơn rất nhiều so với negative) thì đánh giá có thể gây hiềm nhầm.

1.4. Các nghiên cứu liên quan và các vấn đề còn tồn tại

1.4.1. Các nghiên cứu liên quan đến rút gọn thuộc tính trong bảng quyết định không đầy đủ

Trong những năm gần đây, các nghiên cứu liên quan đến rút gọn thuộc tính và trích lọc luật quyết định trong bảng quyết định không đầy đủ có định đã thu hút sự quan tâm của các nhà nghiên cứu. Nhiều thuật toán tìm tập rút gọn đã được đề xuất sử dụng các độ đo khác nhau trên cơ sở mở rộng các độ đo trong lý thuyết tập thô truyền thống. Các thuật toán điển hình có thể kể đến là:

các thuật toán dựa trên miền dương [25, 54, 58], các thuật toán dựa trên hàm ma trận phân biệt [17, 57], các thuật toán dựa trên hàm ma trận phân biệt mở rộng [56], các thuật toán dựa trên tập xấp xỉ thô [14, 21], các thuật toán dựa trên entropy thông tin [26, 64, 72], các thuật toán dựa trên lượng thông tin [18, 22], các thuật toán dựa trên độ đo khoảng cách [1, 19], thuật toán dựa trên hệ số tương quan [85], thuật toán dựa trên thuộc tính thuộc [75].

Các thuật toán đã đề xuất nêu trên đều có điểm chung là: xây dựng một độ đo đặc trưng cho độ quan trọng, hay khả năng phân lớp của thuộc tính; xây dựng thuật toán heuristic tìm một tập rút gọn tốt nhất dựa trên độ quan trọng của thuộc tính. Các thuật toán này đều theo hướng tiếp cận lọc, nghĩa là độ chính xác phân lớp được đánh giá sau khi tìm được tập rút gọn. Do đó, điểm hạn chế là tập rút gọn tìm được chưa tối ưu về số lượng thuộc tính cũng như độ chính xác phân lớp, vì độ đo được sử dụng trong nhiều trường hợp không đại diện cho độ chính xác phân lớp của mô hình.

Trong [3], các tác giả đã đề xuất thuật toán tìm tập rút gọn của bảng quyết định không đầy đủ theo tiếp cận lọc - đóng gói nhằm giảm thiểu số thuộc tính tập rút gọn, từ đó nâng cao hiệu quả của mô hình phân lớp.

1.4.2.Các nghiên cứu liên quan đến rút gọn thuộc tính trong bảng quyết định thay đổi

Việc áp dụng các thuật toán tìm tập rút gọn theo tiếp cận truyền thống đối với bảng quyết định có kích thước lớn và thay đổi, cập nhật gấp nhiều khó khăn, trong đó chủ yếu là hai khó khăn chính: Thứ nhất, trong các bảng quyết định lớn, các thuật toán này gặp khó khăn do không gian bộ nhớ và tốc độ tính toán bị hạn chế. Thứ hai, trong các bảng quyết định thay đổi và cập nhật, thuật toán phải tính toán lại trên toàn bộ bảng quyết định sau mỗi lần thay đổi, do đó thời gian tính toán tăng lên đáng kể. Để khắc phục những khó khăn thách thức đó, các nhà nghiên cứu đã đề xuất *phương pháp gia tăng* tìm tập rút gọn để giảm thời gian thực hiện của các thuật toán. Các thuật toán gia tăng

có khả năng giảm

thiểu thời gian thực hiện và có khả năng thực hiện trên các bảng quyết định không đầy đủ kích thước lớn bằng giải pháp chia nhỏ bảng quyết định.

Các nghiên cứu liên quan đến các thuật toán gia tăng tìm tập rút gọn trong bảng quyết định thay đổi đã và đang thu hút sự quan tâm của các nhà nghiên cứu trong mấy năm gần đây. Phần tiếp theo, luận án trình bày chi tiết các nghiên cứu liên quan đến các thuật toán gia tăng tìm tập rút gọn theo hai hướng: hướng thứ nhất là tiếp cận tập thô truyền thống và các mô hình tập thô mở rộng trên bảng quyết định đầy đủ; hướng thứ hai là mô hình tập thô dung sai trên *bảng quyết định không đầy đủ*, đây là hướng nghiên cứu của luận án.

1.4.2.1. Theo tiếp cận tập thô truyền thống và các mô hình tập thô mở rộng trên bảng quyết định đầy đủ

Trong các bảng quyết định thay đổi với dữ liệu đầy đủ, cho đến nay nhiều thuật toán gia tăng tìm tập rút gọn đã được đề xuất dựa trên tập thô truyền thống và một số tập thô mở rộng. Các nhà nghiên cứu đã đề xuất các thuật toán gia tăng tìm tập rút gọn dựa trên các phương pháp khác nhau trong các trường hợp: bổ sung và loại bỏ tập đối tượng; bổ sung và loại bỏ tập thuộc tính; tập đối tượng và tập thuộc tính thay đổi giá trị.

a) *Với trường hợp bổ sung và loại bỏ tập đối tượng:* các thuật toán gia tăng tìm tập rút gọn được đề xuất sử dụng các độ đo khác nhau như: khoảng cách [16, 30], miền dương [15, 23, 39], ma trận phân biệt [40, 46, 56, 82, 89], ma trận phân biệt mở rộng [40, 56, 84, 91], ma trận phân biệt đơn giản [92], hạt thông tin [35], entropy thông tin [68], hàm thành viên [67], quan hệ không xác định [59], hạt bao phủ [10], tập thô mờ [48, 90, 91].

b) *Với trường hợp bổ sung và loại bỏ tập thuộc tính:* các thuật toán gia tăng tìm tập rút gọn được đề xuất sử dụng các độ đo khác nhau như: hàm thuộc [53], entropy thông tin [78], khoảng cách [32], hạt thông tin [34], ma trận phân biệt [79], ma trận phân biệt nhị phân nén [56], ma trận phân biệt trong hệ quyết định nén [83], quan hệ không xác định [59], quan hệ rõ ràng

[12], tập thô mờ [95, 96].

c) *Với trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị:* các thuật toán gia tăng tìm tập rút gọn được đề xuất sử dụng các độ đo khác nhau như: ma trận phân biệt [84, 92], hạt thông tin [10, 35, 47], entropy thông tin [77], ma trận [11, 41], độ phụ thuộc mờ trong tập thô mờ [96].

Kết quả thực nghiệm của các thuật toán gia tăng cho thấy, các thuật toán gia tăng giảm thiểu đáng kể thời gian thực hiện so với các thuật toán không gia tăng. Do đó, chúng có thể thực thi hiệu quả trên các bảng quyết định không đầy đủ có kích thước lớn và thay đổi, cập nhật. Tuy nhiên, các thuật toán gia tăng trong các công bố nêu trên đều theo tiếp cận lọc truyền thống. Với mục tiêu giảm thiểu số lượng tập rút gọn, từ đó nâng cao hiệu năng của mô hình phân lớp, trong công trình [20], các tác giả xây dựng thuật toán gia tăng tìm tập rút gọn trong trường hợp bổ sung tập đối tượng sử dụng độ đo khoảng cách mờ theo tiếp cận lai ghép lọc - đóng gói. Kết quả thực nghiệm trong công trình [20] cho thấy, tập rút gọn thu được của các thuật toán lọc - đóng gói giảm thiểu đáng kể số thuộc tính tập rút gọn và cải thiện độ chính xác mô hình phân lớp.

1.4.2.2.Theo tiếp cận mô hình tập thô dung sai trên bảng quyết định không đầy đủ

Trong mấy năm gần đây, một số thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ đã được đề xuất bởi các nhóm nghiên cứu với các trường hợp: bổ sung và loại bỏ tập đối tượng [45, 66, 69, 94, 98, 99], bổ sung và loại bỏ tập thuộc tính [70], tập đối tượng và tập thuộc tính thay đổi giá trị [69, 85].

a) *Với trường hợp bổ sung và loại bỏ tập đối tượng:* Yang và các cộng sự [90] xây dựng các thuật toán gia tăng tìm tập rút gọn sử dụng hàm quyết định suy rộng. Liu và các cộng sự [45] giới thiệu các ma trận độ chính xác, ma trận hỗ trợ, ma trận bao phủ và phát triển các thuật toán gia tăng tìm tập rút gọn. Yu J. và các cộng sự [94] giới thiệu entropy dựa trên trọng số và phát triển các thuật toán gia tăng tìm tập rút gọn. Trong bài báo này, các tác giả có xét đến

thứ tự xuất hiện của các thuộc tính trong tập rút gọn. Shu và các cộng sự [66] xây dựng công thức gia tăng tính miền dương. Sử dụng phương pháp gia tăng tìm miền dương, các tác giả

đã trình bày hai thuật toán gia tăng tìm tập rút gọn IARM-I và IARM-E trong trường hợp bổ sung, loại bỏ tập đối tượng tương ứng. Gần đây, năm 2020, Zhang và các cộng sự [98] đã cải tiến các thuật toán gia tăng trong [97] để tìm tập rút gọn. Các tác giả xây dựng các công thức gia tăng độ đo hạt tri thức, dựa trên các cơ chế này, nhóm tác giả đã phát triển hai thuật toán gia tăng: KGIRA-M và KGIRD-M để cập nhật tập rút gọn khi bổ sung và loại bỏ tập đối tượng tương ứng. Xét đến thời gian tính toán và độ chính xác phân lớp, kết quả thực nghiệm trong

[98] cho thấy rằng các thuật toán đa đối tượng KGIRA-M và KGIRD-M hiệu quả hơn các thuật toán đơn đối tượng KGIRA và KGIRD trong [97] tương ứng.

b) *Với trường hợp bổ sung, loại bỏ tập thuộc tính:* Shu và các cộng sự [70] xây dựng các công thức gia tăng cập nhật miền dương, trên cơ sở đó đề xuất hai thuật toán: thuật toán UARA cập nhật tập rút gọn trong trường hợp bổ sung tập thuộc tính và thuật toán UARD cập nhật tập rút gọn trong trường hợp loại bỏ tập thuộc tính. Gần đây, Chen và cộng sự [12] đã đưa ra định nghĩa quan hệ phân biệt được (discernible relation) của một thuộc tính điều kiện đối với thuộc tính quyết định và xây dựng một thuật toán rút gọn thuộc tính dựa trên quan hệ phân biệt. Sau đó, các tác giả xây dựng cơ chế gia tăng để cập nhật quan hệ rõ ràng và đề xuất thuật toán gia tăng IDRA tìm tập rút gọn khi bổ sung tập thuộc tính.

c) *Với trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị:* Shu và các cộng sự trong [69] xây dựng công thức cập nhật miền dương trong trường hợp tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng FSMV cập nhật tập rút gọn. Xie và các cộng sự trong [86] xây dựng công thức cập nhật độ đo không nhất quán trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị, trên cơ sở đó đề xuất hai thuật toán: thuật toán Object-R cập nhật tập rút gọn trong trường hợp tập đối tượng thay đổi giá trị và Attribute-R trong trường hợp tập thuộc tính thay đổi giá trị. Tuy nhiên, các

thuật toán đề xuất nêu trên trong bảng quyết định không đầy đủ (FSMV, Object-R, Attribute-R) đều theo hướng tiếp cận lọc truyền thống.

1.4.3. Các vấn đề còn tồn tại và mục tiêu nghiên cứu của luận án

Thứ nhất: Với bảng quyết định không đầy đủ, tập đối tượng thay đổi trong trường hợp: bổ sung, loại bỏ tập đối tượng và tập đối tượng thay đổi giá trị. Trong [1, 7], các tác giả đã đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi bổ sung tập đối tượng, tuy đã thực nghiệm nhưng so sánh với thuật toán năm 2015, còn thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi loại bỏ tập đối tượng chưa tiến hành thực nghiệm. Vì vậy để hoàn thiện hơn về nghiên cứu, luận án bổ sung thực nghiệm so sánh với thuật toán công bố mới nhất trong trường hợp bổ sung tập đối tượng và tiến hành thực nghiệm trong trường hợp loại bỏ tập đối tượng để đánh giá tính hiệu quả. Với lớp bài toán tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp *tập đối tượng thay đổi giá trị*, các thuật toán đã đề xuất FSMV [69], Object-R[86] đều theo hướng tiếp cận lọc truyền thống. Do đó, luận án nghiên cứu, đề xuất thuật toán FWIA_U_Obj là thuật toán gia tăng theo hướng tiếp cận lọc - đóng gói nhằm giảm thiểu số lượng thuộc tính tập rút gọn so với các thuật toán theo tiếp cận lọc, từ đó nâng cao hiệu quả của mô hình phân lớp.

Thứ hai: Với bảng quyết định không đầy đủ, tập thuộc tính thay đổi trong trường hợp: bổ sung, loại bỏ tập thuộc tính và tập thuộc tính thay đổi giá trị. Trong [2], các tác giả chỉ xem xét trường hợp bổ sung tập thuộc tính, chưa nghiên cứu về trường hợp loại bỏ tập thuộc tính. Công thức tính khoảng cách trong [1] không sử dụng phương pháp gia tăng để xem xét phần thay đổi khi bổ sung, loại bỏ tập thuộc tính. Trong phần này, luận án xây dựng công thức cập nhật khoảng cách trong trường hợp bổ sung, loại bỏ tập thuộc tính có sử dụng phương pháp gia tăng và khác với công thức trong [1, 2], trên cơ sở đó xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập thuộc tính. Với lớp bài toán tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp *tập thuộc tính thay đổi giá trị*, thuật toán Attribute-R [86] theo hướng tiếp cận lọc truyền thống. Do đó, luận án nghiên cứu, đề xuất thuật toán FWIA_U_Attr là thuật toán gia tăng theo hướng tiếp cận lọc - đóng gói nhằm giảm thiểu số lượng thuộc tính tập rút gọn so với các thuật toán theo tiếp cận lọc, từ đó nâng cao hiệu quả của mô hình

phân lớp.

1.5. Bộ dữ liệu thực nghiệm

Số liệu thực nghiệm: Tiến hành thực nghiệm trên 06 bộ dữ liệu được lấy trong kho dữ liệu UCI [73] như mô tả ở bảng 1.1.

Trong đó: Các cột $|O|$, $|A|$, $|k|$ được ký hiệu tương ứng là: Số đối tượng; Số thuộc tính điều kiện; Số lớp quyết định.

Bảng 1.1- Các bộ dữ liệu sử dụng trong thực nghiệm

STT	Tập dữ liệu	$ O $	$ A $	$ k $
1	Audiology.data	226	69	24
2	Soybean-laarge.data	307	35	2
3	house-votes-84.data	435	16	2
4	Arrhythmia.data	452	279	16
5	Anneal.data	798	38	6
6	Ad.data	3279	1558	2

1.6. Kết luận chương 1

Nhu vậy chương 1 đã trình bày các khái niệm về mô hình tập thô truyền thống trên bảng quyết định đầy đủ, mô hình tập thô dung sai trên bảng quyết định không đầy đủ. Chương này cũng trình bày tổng quan về hướng tiếp cận lọc, đóng gói trong rút gọn thuộc tính. Nhằm đưa ra bức tranh tổng thể về rút gọn thuộc tính theo tiếp cận tập thô, đồng thời chương 1 trình bày tổng quan các nghiên cứu liên quan đến rút gọn thuộc tính theo tiếp cận tập thô dung sai, các thuật toán gia tăng tìm tập rút gọn trong bảng quyết định theo tiếp cận tập thô truyền thống và các mô hình mở rộng, các thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ theo tiếp cận tập thô dung sai. Trên cơ sở đó, chương 1 phân tích các vấn đề còn tồn tại của các thuật toán trên lớp bài toán luận án giải quyết. Từ đó, chương 1 đưa ra các mục tiêu luận án cần giải quyết.

CHƯƠNG 2. PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH TRONG BẢNG QUYẾT ĐỊNH KHÔNG ĐẦY ĐỦ KHI TẬP ĐỐI TƯỢNG THAY ĐỔI

2.1. Mở đầu

Khi xử lý các bảng dữ liệu có kích thước lớn, thay đổi, việc áp dụng các thuật toán tìm tập thuộc tính rút gọn để xây dựng mô hình phân lớp, dự báo hiệu quả theo cách tiếp cận truyền thống gặp rất nhiều thách thức lớn. Chẳng hạn, trong các hệ thống trực tuyến (online), các bảng dữ liệu có kích thước rất lớn, dữ liệu mới liên tục được bổ sung vào và dữ liệu cũ không ngừng được xóa đi các đối tượng, đồng thời các đối tượng cũng liên tục thay đổi giá trị.

Để xây dựng các thuật toán hiệu quả tìm tập rút gọn trên các bảng quyết định thay đổi, các nhà nghiên cứu đề xuất các thuật toán *gia tăng* nhằm giảm thiểu thời gian thực hiện. Đầu tiên, xuất phát từ nghiên cứu thuật toán gia tăng tìm tập rút gọn trên bảng quyết định không đầy đủ có dữ liệu cố định, sau đó nghiên cứu các thuật toán gia tăng tìm tập rút gọn trong các trường hợp bổ xung, loại bỏ tập đối tượng. Việc nghiên cứu, thực nghiệm các thuật toán nêu trên nhằm hoàn thiện hơn nữa các thuật toán đã công bố, trên cơ sở đó làm tiền đề cho việc xây dựng, đề xuất các thuật toán mới giải quyết trường hợp còn lại của bảng quyết định không đầy đủ thay đổi xuất hiện phổ biến trong các bài toán thực tiễn: trường hợp tập đối tượng thay đổi giá trị.

Cụ thể chương này trình bày nghiên cứu như sau:

- 1) Nghiên cứu, hoàn thiện thực nghiệm thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập đối tượng so sánh với các thuật toán công bố mới nhất.
- 2) Xây dựng công thức cập nhật khoảng cách trong trường hợp *tập đối tượng thay đổi giá trị*, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập đối tượng thay đổi giá trị. Thực nghiệm so sánh hướng tiếp cận rút gọn thuộc tính trực tiếp với hướng tiếp cận rút gọn thuộc tính gián tiếp khi thực hiện đồng thời loại bỏ sau đó bổ sung tập đối tượng.

2.2. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung, loại bỏ tập đối tượng

Trong [19], các tác giả đã đưa ra công thức tính khoảng cách, như sau:

Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$

và U / SIM là hai phủ sinh bởi $P, Q \subseteq C$. Khi đó:

$$(P),$$

$D(P, Q) = \frac{1}{n^2} \sum_{i=1}^n \left(|S(u_i) \cup S(u_i) - S(u_i) \cap S(u_i)| \right)$ là một khoảng cách giữa hai phủ

$$U / SIM \text{ và } U / SIM$$

$$(P) \quad (Q). \text{ Giả sử } M(C) = \lfloor c_{ij} \rfloor, \quad M(\{d\}) = \lfloor d_{ij} \rfloor$$

tương ứng

là ma trận dung sai trên C và d . Khi đó, khoảng cách giữa hai tập thuộc tính C

và $C \cup \{d\}$ được xác định như sau:

$$\sum_{i=1}^n (|S(u_i) - S(u_i) \cap S(u_i)|) = \sum_{i=1}^n (c_{ij} - d_{ij})$$

$$\sum_{i=1}^n$$

(2.I)
 $c.d)$

$$n^2 \underset{i=1}{\dots} \quad C \quad i \quad \{d\} \quad \overset{\text{---}}{i} \quad n^2 \underset{i=1}{\dots} \underset{j=1}{\dots} \quad ij \quad ij \quad ij$$

Dựa trên khoảng cách được xây dựng, các tác giả trong [19] định nghĩa tập rút gọn và độ quan trọng của thuộc tính dựa trên khoảng cách.

Định nghĩa 2.1.[19] Cho bảng $IDS = (U, C \cup \{d\})$
 quyết định không đầy đủ với $B \subseteq C$.

Nếu:

$$1) D(B, B \cup \{d\}) = D(C, C \cup \{d\})$$

$$2) \forall b \in B, \quad D(B - \{b\}, \{B - \{b\}\} \cup \{d\}) \neq D(C \cup \{d\})$$

thì B là một tập rút gọn của C dựa trên $IDS = (U, C \cup \{d\})$

Định nghĩa 2.2.[19] Cho bảng
 quyết định không đầy đủ
 với $B \subseteq C$
 nghĩa bởi:

và $b \in C - B$. Độ quan trọng của thuộc tính b
 đối với B được định

$$SIG_B(b) = D(B, B \cup \{d\}) - D(B \cup \{b\}, B$$

$$\cup \{b\} \cup \{d\})$$

Độ quan trọng $SIG_B(b)$

đặc trưng cho chất lượng phân lớp của thuộc tính

b đối với thuộc tính quyết định d và được sử dụng làm tiêu chuẩn rút gọn thuộc tính cho thuật toán heuristic tìm tập rút gọn.

2.2.1. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định trong trường hợp bổ sung tập đối tượng

2.2.1.1. Mô tả thuật toán

Trong [1, 7], các tác giả xây dựng công thức gia tăng cập nhật khoảng cách cho bởi công thức sau đây:

Cho bảng $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$.
 quyết định
 không đầy đủ

Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_{n+1}, u_{n+2}, \dots, u_{n+s}\}$ được bổ sung vào U

với $s \geq 1$, đặt $M_{U \cup \Delta U}(C) = \begin{bmatrix} c_{ij} \end{bmatrix}_{(n+s) \times (n+s)}$ và $M_{U \cup \Delta U}(\{d\}) =$

$[d_{ij}]$

tương ứng là

mã trận dung sai trên C và $\{d\}$. Khi đó, công thức tính gia tăng khoảng cách như sau:

$$D(\{d\}) \leftarrow C^2C - \frac{1}{c} \left(C, C \cup \{d\} \right) + \frac{2}{c \cdot d} \quad (2.2)$$

$$\left| \begin{array}{c} n+s \\ \hline n+s \end{array} \right| \quad \sum \sum_{i=n+1, j=1}^{(n+s)^2} ij - ij$$

$$U$$

Từ công thức (2.2), các tác giả trong [1,7] đưa ra kết quả sau đây:

Cho bảng $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$,
 quyết định không đầy đủ

$R \subseteq C$ là tập rút gọn C của IDS. Giả ΔU được sử tập đối tượng gồm s phần tử

bổ sung vào U với $s \geq 1$. Khi đó chúng ta có:

Nếu $S_R(u_{n+i}) \subseteq S_{\{d\}}(u_{n+i})$ với $i=1, \dots, s$ thì R là tập rút gọn của C trên

$$IDS_1 = (U \cup \Delta U, C \cup \{d\})$$

Từ đó, thuật toán gia tăng lọc - đóng gói tìm tập rút gọn sử dụng độ đo khoảng cách trong trường hợp bổ sung tập đối tượng [1,7] được mô tả như sau:

Thuật toán
IDS_IFW_AO

$$IDS = (U, C \cup \{d\})$$

với

Đầu vào: Cho bảng quyết định không đầy đủ

$$U = \{u_1, u_2, \dots, u_n\}, C$$

$$= \{c_1, c_2, \dots, c_n\}; \text{Tập rút}$$

$$\text{gọn } R \subseteq C.$$

- Ma trận dung $M_U(C)$ và $M_U(\{d\})$.
 sai: $M_U(R)$,

- Tập đối tượng bổ sung $\Delta U = \{u_{n+1}, u_{n+2}, \dots, u_{n+s}\}$ với s phần tử.

Đầu ra: Tìm tập rút gọn R_{best} trên $IDS_1 = (U \cup \Delta U, C \cup \{d\})$

Bước 1: Khởi tạo và kiểm tra

1. $T := \emptyset$; // T chứa các ứng viên của tập rút gọn.

2. Tính $M_{U \cup \Delta U}(R)$, $M_{U \cup \Delta U}(C)$ và $M_{U \cup \Delta U}(\{d\})$;
 các ma trận

3. $i := 1$;

4. Repeat

5. If $S_R(u_{n+i}) \subseteq S_{\{d\}}(u_{n+i})$ then $\Delta U = \Delta U - \{u_{n+i}\};$

6. $i := i + 1;$

7. Until $i = s;$

8. If $\Delta U = \emptyset$ then Return $R;$

Bước 2: Tìm tập rút gọn

9. Tính $D_U(R, R \cup \{d\}), D_U(C, C \cup \{d\});$

10. Tính

$D_{U \cup \Delta U}(R, R \cup \{d\}), D_{U \cup \Delta U}(C, C \cup \{d\})$ sử dụng công thức (2.2)

//Giai đoạn lọc, tìm các ứng viên cho tập rút gọn

11. $j := 0;$

12. Repeat

13. $j := j + 1;$

- For each $r \in C \cap R$
- Tính $SIG_R[r]$;

1 Chọn $r_m \in C \cap R$ sao cho $SIG_R[r_m] \sqsupseteq \max_{r \in C \cap R}$
6 $R := R \setminus \{r_m\};$
1 $T_j :=$
⋮
1 $T := T \cup T_j; // T là tập hợp mà mỗi phần tử của nó là một tập$
⋮

20. Until $D_{U \cup U} \subseteq R, R \subseteq d \subseteq D_{U \cup U} \subseteq C, C \subseteq d \subseteq$

//Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất

21. For $i = 1$ to i

2 Tính độ chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp kiểm tra chéo 10-fold;

- $R_{best} \subseteq T_k //$ với $T_k (1 \leq k \leq t)$ có độ chính xác phân lớp cao nhất.
- Return $R_{best}.$

Theo [1,7], độ phức tạp của thuật toán IDS_IFW_AO là $O(|C|^2 * (|U| + |\Delta U|)^2) + O(|C| * f(n))$ với $f(n)$ là thời gian tính bộ phân lớp.

2.2.1.2. Thực nghiệm, đánh giá thuật toán IDS_IFW_AO

Trong các công trình [1, 7], các tác giả chỉ mới thực nghiệm thuật toán IDS_IFW_AO với thuật toán IARM-I [66] năm 2015. Trong phần này, luận án hoàn thiện thực nghiệm thuật toán IDS_IFW_AO ở hai điểm: *thứ nhất* là bổ sung so sánh, đánh giá với thuật toán mới nhất KGIRA-M [98] năm 2020, *thứ hai* là kết quả thực nghiệm được đánh giá qua 10 lần chạy thực nghiệm.

a) Mục tiêu thực nghiệm

Đánh giá tính hiệu quả của thuật toán dựa trên các tiêu chí: *Số lượng thuộc tính trong tập rút gọn, độ chính xác phân lớp và thời gian thực hiện*.

Thuật toán IDS_IFW_AO được so sánh với hai thuật toán tìm tập rút gọn trong bảng quyết định không đầy đủ theo tiếp cận lọc trong trường hợp bổ sung tập đối tượng, đó là thuật toán IARM-I[66] (IARM-I là thuật toán gia tăng tìm tập rút gọn sử dụng miền dương) và thuật toán KGIRA-M[98] (KGIRA-M là thuật toán gia tăng tìm tập rút gọn sử dụng độ đo hạt tri thức)

b) Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Tiến hành thực nghiệm trên 06 bộ dữ liệu được lấy trong kho dữ liệu UCI [73] như mô tả ở bảng 2.1.

Mỗi tập dữ liệu được chia thành hai phần xấp xỉ bằng nhau: Tập dữ liệu ban đầu được ký hiệu là O_{ori} và tập dữ liệu gia tăng được ký hiệu là O_{inc} . Tiếp theo, tập dữ liệu gia tăng O_{inc} được chia thành năm phần bằng nhau được ký hiệu lần lượt là O_1, O_2, O_3, O_4, O_5 .

Trong bảng 2.1, các cột $|O|$, $|O_{ori}|$, $|O_{inc}|$, $|A|$, $|k|$ được ký hiệu tương ứng là: Số đối tượng; Số đối tượng trong O_{ori} ; Số đối tượng trong O_{inc} ; Số thuộc tính điều kiện; Số lớp quyết định.

Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold, nghĩa là bộ dữ

liệu được chia thành 10 phần xấp xỉ bằng nhau, lấy lần lượt 1 phần làm bộ dữ liệu kiểm tra, 9 phần còn lại làm dữ liệu huấn luyện. Quá trình được lặp lại 10 lần. Độ chính xác phân lớp được biểu diễn bởi $v \pm \sigma$ trong đó v là giá trị độ chính xác trung bình (mean) của 10 lần lặp và σ là sai số chuẩn (standard error).

Bảng 2.1. Các bộ dữ liệu sử dụng trong thực nghiệm khi bổ sung và loại bỏ tập đối tượng

STT	Tập dữ liệu	$ O $	$ O_{ori} $	$ O_{inc} $	$ A $	$ k $
1	Audiolgy.data	226	116	110	69	24
2	Soybean-laarge.data	307	157	150	35	2
3	house-votes-84.data	435	220	215	16	2
4	Arrhythmia.data	452	222	230	279	16
5	Anneal.data	798	393	405	38	6
6	Ad.data	3279	1644	1635	1558	2

Môi trường thực nghiệm: Thực nghiệm được thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel® Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

c) Kịch bản thực nghiệm

Để đánh giá hiệu suất của thuật toán IDS_IFW_AO, trước hết cả ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M được thực hiện trên $|O_{ori}|$. Sau đó, cả ba thuật toán này được chạy bằng cách bổ sung lần lượt từ O_1 đến O_5 của O_{inc} .

d) Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 2.2 trình bày kết quả về số thuộc tính trong tập rút gọn và độ chính xác phân lớp của các thuật toán IDS_IFW_AO, IARM-I và KGIRA-M. Trong đó cột

$|R|$ là số thuộc tính trong tập rút gọn và cột Acc là độ chính xác phân lớp.

Từ bảng 2.2 nhận thấy rằng: độ chính xác phân lớp và số lượng thuộc tính của thuật toán IDS_IFW_AO tối ưu hơn so với IARM-I, KGIRA-M. Thuật toán KGIRA-M hiệu quả hơn thuật toán IARM-I về cả độ chính xác của phân lớp và số lượng thuộc tính trong tập rút gọn.

Bảng 2.2. Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M

STT	Tập dữ liệu	Tập dữ liệu bổ sung	IDS_IFW_AO		IARM-I		KGIRA-M	
			R	Acc	R	Acc	R	Acc
1	Audiology	O_{ori}	5	76.18 ± 0.25	8	74.29 ± 0.32	8	74.82 ± 0.28
		O_1	5	76.18 ± 0.18	9	75.12 ± 0.28	9	75.26 ± 0.16
		O_2	6	81.26 ± 0.46	12	78.26 ± 0.34	11	78.89 ± 0.29
		O_3	6	81.26 ± 0.09	12	78.26 ± 0.16	11	78.89 ± 0.07
		O_4	7	78.84 ± 0.32	14	78.17 ± 0.26	13	78.26 ± 0.18
		O_5	7	78.84 ± 0.48	15	76.64 ± 0.41	14	77.45 ± 0.26
2	Soybean-large	O_{ori}	5	96.12 ± 0.16	7	95.46 ± 0.28	7	95.52 ± 0.54
		O_1	5	96.12 ± 0.24	7	95.46 ± 0.17	7	95.52 ± 0.18
		O_2	6	96.72 ± 0.48	9	95.04 ± 0.35	9	95.04 ± 0.46
		O_3	7	95.18 ± 0.26	9	95.04 ± 0.42	9	95.04 ± 0.15
		O_4	7	95.18 ± 0.18	10	94.19 ± 0.26	10	94.26 ± 0.24
		O_5	8	94.58 ± 0.08	11	94.28 ± 0.19	11	94.18 ± 0.08
3	house-votes-84	O_{ori}	4	92.48 ± 0.42	9	91.17 ± 0.36	8	92.16 ± 0.12
		O_1	5	92.76 ± 0.36	10	91.45 ± 0.25	9	91.68 ± 0.53
		O_2	7	94.48 ± 0.25	14	92.28 ± 0.38	12	93.04 ± 0.17
		O_3	7	94.48 ± 0.25	14	92.28 ± 0.38	13	93.82 ± 0.15
		O_4	9	94.12 ± 0.18	16	92.06 ± 0.22	14	93.68 ± 0.26
		O_5	9	94.12 ± 0.12	17	92.88 ± 0.24	16	94.01 ± 0.12
4	Arrhythmia	O_{ori}	6	70.08 ± 0.34	14	69.16 ± 0.49	12	70.06 ± 0.04
		O_1	7	72.45 ± 0.19	17	72.05 ± 0.26	14	72.28 ± 0.18

		O_2	7	72.45 ± 0.22	17	72.05 ± 0.35	16	72.39 ± 0.06
		O_3	8	74.18 ± 0.41	21	73.23 ± 0.38	18	73.88 ± 0.24
		O_4	8	74.18 ± 0.35	21	73.23 ± 0.16	19	74.06 ± 0.08
		O_5	9	76.04 ± 0.16	24	73.08 ± 0.08	21	75.08 ± 0.32
5	Anneal	O_{ori}	4	84.18 ± 0.48	8	84.06 ± 0.24	7	83.69 ± 0.28
		O_1	5	89.06 ± 0.54	8	84.06 ± 0.54	8	86.24 ± 0.65
		O_2	5	89.06 ± 0.17	8	84.06 ± 0.28	9	87.03 ± 0.54
		O_3	6	91.28 ± 0.26	9	88.48 ± 0.35	10	89.28 ± 0.38
		O_4	6	91.28 ± 0.31	9	88.48 ± 0.64	10	89.28 ± 0.12
		O_5	6	91.28 ± 0.22	10	90.06 ± 0.34	11	90.68 ± 0.26
6	Ad	O_{ori}	12	93.01 ± 0.24	23	92.16 ± 0.36	20	91.86 ± 0.42
		O_1	14	91.18 ± 0.62	28	90.48 ± 0.55	25	90.09 ± 0.18
		O_2	14	91.18 ± 0.58	28	90.48 ± 0.27	26	90.78 ± 0.12
		O_3	17	91.65 ± 0.39	32	91.17 ± 0.14	32	91.05 ± 0.25
		O_4	18	92.82 ± 0.24	36	92.06 ± 0.08	34	92.46 ± 0.14
		O_5	19	92.90 ± 0.32	45	92.46 ± 0.28	42	92.58 ± 0.16

Do đó, mô hình phân lớp dựa trên tập rút gọn của thuật toán IDS_IFW_AO hiệu quả hơn mô hình phân lớp của thuật toán IARM-I và thuật toán KGIRA-M về chất lượng phân lớp và độ phức tạp của mô hình.

e) Đánh giá thời gian thực hiện

Thời gian thực hiện của ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M được trình bày trong bảng 2.3.

Bảng 2.3. Thời gian thực hiện của ba thuật toán IDS_IFW_AO, IARM-I và KGIRA-M (tính theo giây)

STT	Tập dữ liệu	Tập dữ liệu Bổ sung	IDS_IFW_AO		IARM-I		KGIRA-M	
			Thời gian thực hiện	Tổng Thời gian thực hiện	Thời gian thực hiện	Tổng Thời gian thực hiện	Thời gian thực hiện	Tổng Thời gian thực hiện
1	Audiology	O_{ori}	6.08	6.08	5.82	5.82	5.78	5.78
		O_1	0.61	6.69	0.51	6.33	0.46	6.24
		O_2	0.35	7.04	0.26	6.59	0.38	6.62
		O_3	0.64	7.68	0.42	7.01	0.50	7.12
		O_4	0.34	8.02	0.28	7.29	0.20	7.32
		O_5	0.44	8.46	0.35	7.64	0.41	7.73
2	Soybean-large	O_{ori}	3.04	3.04	2.86	2.86	2.74	2.74
		O_1	0.64	3.68	0.42	3.28	0.45	3.19
		O_2	0.34	4.02	0.22	3.52	0.49	3.68
		O_3	0.73	4.75	0.54	4.06	0.47	4.15
		O_4	0.43	5.18	0.34	4.40	0.27	4.42
		O_5	0.68	5.86	0.40	4.80	0.34	4.76
3	house-votes-8	O_{ori}	5.86	5.86	5.03	5.03	4.98	4.98
		O_1	0.56	6.42	0.39	5.42	0.38	5.36
		O_2	0.61	7.03	0.46	5.88	0.38	5.74
		O_3	0.53	7.56	0.37	6.25	0.43	6.17
		O_4	0.47	8.03	0.31	6.56	0.31	6.48
		O_5	0.55	8.58	0.32	6.88	0.37	6.85
4	Arrhythmia	O_{ori}	35.48	35.48	28.72	28.72	26.78	26.78
		O_1	1.58	37.06	1.42	30.14	1.46	28.24
		O_2	3.12	40.18	2.26	32.40	2.62	30.86
		O_3	2.50	42.68	2.03	34.43	1.92	32.78
		O_4	1.36	44.04	1.15	35.58	1.02	33.80
		O_5	2.14	46.18	1.84	37.42	1.66	35.46
5	Anneal	O_{ori}	7.48	7.48	6.05	6.05	6.12	6.12
		O_1	0.58	8.06	0.38	6.43	0.34	6.46
		O_2	0.81	8.95	0.63	7.06	0.52	6.98
		O_3	0.53	9.48	0.34	7.40	0.31	7.29
		O_4	0.77	10.25	0.56	7.96	0.48	7.77
		O_5	0.80	11.05	0.59	8.55	0.62	8.39
6	Ad	O_{ori}	96.74	96.74	82.05	82.05	80.72	80.72
		O_1	5.69	102.43	4.84	86.89	4.26	84.98
		O_2	6.13	108.56	5.18	92.07	4.98	89.96
		O_3	5.70	114.26	4.26	96.33	4.65	94.61
		O_4	3.86	118.12	2.54	98.87	2.86	97.47
		O_5	4.74	122.86	2.98	101.85	2.84	100.31

Trên tất cả các tập dữ liệu trong bảng 2.3, thuật toán IDS_IFW_AO có thời gian thực hiện cao hơn thuật toán IARM-I và thuật toán KGIRA-M vì thuật toán IDS_IFW_AO cần nhiều thời gian hơn để thực hiện phân lớp trong giai đoạn đóng gói. Trong khi đó thời gian thực hiện của thuật toán IARM-I xấp xỉ bằng thuật toán KGIRA-M.

2.2.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định trong trường hợp loại bỏ tập đối tượng

2.2.2.1. Mô tả thuật toán

Trong [1], tác giả đã xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp loại bỏ tập đối tượng, thuật toán IDS_IFW_DO. Tuy nhiên, chưa tiến thành thực nghiệm để đánh giá tính hiệu quả của thuật toán trên các bộ dữ liệu mẫu. Trong mục này, luận án tiến hành thực nghiệm thuật toán IDS_IFW_DO nhằm đánh giá tính hiệu quả so với các thuật toán đã công bố với kịch bản loại bỏ tập đối tượng. Trước hết, luận án trình bày thuật toán IDS_IFW_DO.

Cho bảng quyết định không đầy $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$ và

$$M_U(\{d\}) = [[d_{i,j}]]_{n \times n} \quad \text{tương ứng là ma trận dung sai của } C \text{ và } d.$$

$$M_U(C) = [[c_{i,j}]]_{n \times n},$$

Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ bị loại khỏi U , $s < n$

Khi đó, công thức cập nhật khoảng cách được xây dựng trong [1] như sau:

$$\frac{D^2_D(u_i, C \cup \{d\})}{(n-1)} \quad (2.3)$$

$$c.d - \left(\frac{c^{k+s-1} d^{i-1}}{c} - c.d \right)$$

Từ công thức cập nhật khoảng cách (2.3) và định nghĩa 2.1 về tập rút gọn dựa trên khoảng cách ta có kết quả sau đây [1]:

$$\text{Cho bảng quyết định không đầy đủ } IDS = (U, C \cup \{d\}) \quad \text{với } U = \{u_1, u_2, \dots, u_n\}$$

và $B \subseteq cl_{\Delta}$ là tập rút gọn dựa trên khoảng cách. Giả sử tập đối tượng gồm s phần

từ $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ bị loại khỏi U , $s < n$. Khi đó ta có: Nếu $S_B(u_i) \subseteq S_{\{d\}}(u_i)$
 thì B là tập rút gọn của $IDS_1 = (U - \Delta U, C \cup \{d\})$
 với mọi $i = k..(k + s - 1)$

Từ đó, thuật toán gia tăng lọc - đóng gói IDS_IFW_DO tìm tập rút gọn trong trường hợp loại bỏ tập đối tượng như sau:

Thuật toán

IDS_IFW_DO

Đầu vào: Cho bảng quyết định không đầy đủ

$IDS \subseteq U, C$

v

$U \subseteq u_1, u_2, \dots, u_n$; Tập rút gọn $R \subseteq C$;

$\{d\}$

ó

- Các ma trận dung sai: $M_U(B) \subseteq b_{i,j} \in \mathbb{R}^{n \times n}$, $M_U(C) \subseteq c_{i,j} \in \mathbb{R}^{n \times n}$, $M_U(\{d\}) \subseteq d_{i,j} \in \mathbb{R}^{n \times n}$;

- Tập đối tượng loại bỏ $U \setminus u_k, u_{k+1}, \dots, u_{k+s-1}$ gồm s đối tượng với $0 < s < n$;

Đầu ra: Tìm tập rút gọn R_{best} của C trên $IDS_1 \subseteq U \subseteq U, C \cup \{d\}$

Bước 1: Khởi tạo và kiểm tra

1. $T \subseteq // T$ chứa các ứng viên của tập rút gọn.

2. $i := k$;

· Repeat

· If $S_R(u_i) \subseteq S_{\{d\}}(u_i)$ then $U \subseteq U \setminus \{u_i\}$;

5. $i := i + 1$;

· Until $i = k+s-1$;

· If $U \subseteq$ then Return R ; //Tập rút gọn không thay đổi

Bước 2: Cập nhật tập rút gọn

// Giai đoạn lọc, tìm các ứng viên cho tập rút gọn

11. j:=0;

12. Repeat 13.

j:=j+1;

- For mỗi $r \in R$
- Tính $SIG_R(r)$;
- Chọn $r_m \in R$ sao cho $SIG(r_m) = \min\{SIG_{R \setminus \{r\}}(r), \forall r \in R\}$

17. $R \leftarrow R \setminus \{r_m\}$;

· $T_j \leftarrow R$;

· $T \leftarrow T \cup T_j$;

20. Until $D_{U \cup U} \subseteq R, R \subseteq d \subseteq D_{U \cup U} \subseteq C, C \subseteq d \subseteq$;

// Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất

- For $i \in I$ to j
- Tính chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp kiểm tra chéo 10-fold;
- $R_{best} \leftarrow T_k$; //với $T_k (I \leq k \leq t)$ có độ chính xác phân lớp cao nhất;
- Return R_{best} .

Theo [1], độ phức tạp của thuật toán IDS_IFW_DO là với $f(n)$ là thời gian tính bộ phân lớp.

$$O \left(C^2 * (U - \Delta U)^2 \right) + O(C * f(n))$$

2.2.2.2. Thực nghiệm, đánh giá thuật toán IDS_IFW_DO

a) Mục tiêu thực nghiệm

Trong phần này, luận án tiến hành thực nghiệm để đánh giá tính hiệu quả của thuật toán dựa trên các tiêu chí: *số lượng thuộc tính trong tập rút gọn, độ chính xác phân lớp và thời gian thực hiện*. Thuật toán IDS_IFW_DO được so sánh với hai thuật toán IARM-E[66] và thuật toán KGIRD-M[98]. Thuật toán IARM-E là thuật toán cập nhật tập rút gọn trong bảng quyết định không đầy đủ theo tiếp cận lọc trong trường hợp loại bỏ tập đối tượng sử dụng miền dương và thuật toán KGIRD-M là thuật toán cập nhật tập rút gọn trong bảng quyết định không đầy đủ theo tiếp cận lọc trong trường hợp loại bỏ tập đối tượng sử dụng độ đo hạt tri thức.

b) Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Thực nghiệm này vẫn sử dụng 06 tập dữ liệu trong bảng 2.1. Trong đó, tập dữ liệu ban đầu, ký hiệu là O , được chia thành 10 phần xấp xỉ bằng nhau theo số lượng của tập đối tượng. Chọn ngẫu nhiên 4 phần để loại bỏ các tập đối tượng, ký hiệu lần lượt là O_1, O_2, O_3, O_4 .

Các thuật toán IDS_IFW_DO, IARM-E và KGIRD-M thực hiện loại bỏ lần lượt từng phần một: O_1, O_2, O_3, O_4 . Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold.

Độ chính xác phân lớp được biểu diễn bởi $\nu \pm \sigma$ trong đó ν là giá trị độ chính xác trung bình (mean) của 10 lần lặp và σ là sai số chuẩn (standard error).

Môi trường thực nghiệm: Thực nghiệm được thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel® Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

c) Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 2.4 trình bày kết quả về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của các thuật toán IDS_IFW_DO, IARM-E và KGIRD-M.

Trong đó cột

$|R|$ và Acc lần lượt là số thuộc tính trong tập rút gọn và độ chính xác phân lớp.

Bảng 2.4. Số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp của ba thuật toán IDS_IFW_DO, IARM-E và KGIRD-M

STT	Tập dữ liệu	Tập dữ liệu loại bỏ	IDS_IFW_DO		IARM-E		KGIRD-M	
			$ R $	Acc	$ R $	Acc	$ R $	Acc
1	Audiology	O_1	7	78.22 ± 0.26	14	78.06 ± 0.12	12	78.14 ± 0.23
		O_2	6	80.86 ± 0.43	12	79.92 ± 0.18	11	79.74 ± 0.28
		O_3	6	81.12 ± 0.18	11	79.08 ± 0.44	10	80.28 ± 0.32
		O_4	5	77.24 ± 0.34	9	77.02 ± 0.16	8	76.94 ± 0.18
2	Soybean-large	O_1	9	94.82 ± 0.52	12	94.26 ± 0.26	13	94.72 ± 0.09
		O_2	10	94.96 ± 0.35	14	94.43 ± 0.18	14	94.83 ± 0.16
		O_3	7	96.28 ± 0.26	9	95.14 ± 0.41	9	96.25 ± 0.06
		O_4	8	96.72 ± 0.38	10	95.28 ± 0.25	10	96.43 ± 0.12
3	house-votes-84	O_1	9	93.68 ± 0.18	16	93.02 ± 0.09	18	92.98 ± 0.16
		O_2	10	94.76 ± 0.26	18	93.58 ± 0.15	19	93.24 ± 0.25
		O_3	7	92.68 ± 0.45	15	91.75 ± 0.34	15	92.42 ± 0.17
		O_4	6	91.53 ± 0.29	12	91.26 ± 0.17	13	91.18 ± 0.09
4	Arrhythmia	O_1	8	73.68 ± 0.46	19	72.69 ± 0.32	20	73.14 ± 0.28
		O_2	9	74.06 ± 0.54	21	73.82 ± 0.19	21	73.46 ± 0.35
		O_3	11	72.96 ± 0.32	23	72.62 ± 0.14	22	71.84 ± 0.28

		O_4	7	70.54 ± 0.25	18	70.26 ± 0.08	19	70.18 ± 0.16
5	Anneal	O_1	6	92.62 ± 0.15	10	92.46 ± 0.09	11	91.86 ± 0.23
		O_2	8	93.17 ± 0.52	13	93.06 ± 0.08	12	92.68 ± 0.18
		O_3	9	93.98 ± 0.38	14	93.14 ± 0.15	13	92.72 ± 0.35
		O_4	7	91.06 ± 0.48	9	89.75 ± 0.32	10	90.68 ± 0.23
6	Ad	O_1	7	91.46 ± 0.26	9	90.72 ± 0.14	11	90.26 ± 0.34
		O_2	6	92.75 ± 0.65	8	91.67 ± 0.42	9	91.45 ± 0.18
		O_3	8	89.26 ± 0.34	11	88.64 ± 0.18	10	89.05 ± 0.14
		O_4	5	88.76 ± 0.18	8	88.13 ± 0.12	9	87.58 ± 0.25

Dựa trên kết quả trong bảng 2.4, nhận thấy rằng: Độ chính xác phân lớp của thuật toán IDS_IFW_DO cao hơn một chút so với thuật toán IARM-E và

thuật toán KGIRD-M. Hơn nữa, số lượng thuộc tính trong tập rút gọn của thuật toán IDS_IFW_DO nhỏ hơn nhiều so với hai thuật toán IARM-E và KGIRD-

M. Do đó, chất lượng phân lớp của thuật toán IDS_IFW_DO tốt hơn so với thuật toán IARM-E và thuật toán KGIRD-M.

d) Đánh giá thời gian thực hiện

Thời gian thực hiện của các thuật toán IDS_IFW_DO, IARM-E và KGIRD-M. trình bày trong bảng 2.5 dưới đây.

Bảng 2.5. Thời gian thực hiện của ba thuật toán: IDS_IFW_DO, IARM-E và KGIRD-M (tính theo giây)

STT	Tập dữ liệu	Tập dữ liệu loại bỏ	IDS_IFW_DO	IARM-E	KGIRD-M
1	Audiology	O_1	0.78	0.62	0.58
		O_2	1.36	1.08	0.96
		O_3	2.24	1.86	1.35
		O_4	3.46	2.98	2.76
2	Soybean-large	O_1	0.82	0.69	0.62
		O_2	1.88	1.54	1.48
		O_3	2.96	2.32	2.26
		O_4	4.16	3.64	3.58
3	house-votes-84	O_1	0.52	0.38	0.36
		O_2	1.18	0.85	0.86
		O_3	1.82	1.38	1.32
		O_4	2.54	2.06	2.18
4	Arrhythmia	O_1	1.62	1.24	1.19
		O_2	3.18	2.63	2.58
		O_3	5.06	4.45	4.39
		O_4	6.87	5.98	5.87
5	Anneal	O_1	0.64	0.42	0.44
		O_2	1.38	1.02	0.96
		O_3	2.18	1.75	1.64
		O_4	2.84	2.23	2.06
6	Ad	O_1	6.84	5.96	5.28
		O_2	11.85	10.02	9.76
		O_3	19.76	16.34	14.84
		O_4	26.64	22.48	20.36

Nhìn vào bảng 2.5, kết quả thời gian thực hiện của thuật toán IDS_IFW_DO

cao hơn thuật toán IARM-E và thuật toán KGIRD-M trên tất cả

các tập dữ liệu vì IDS_IFW_DO cần nhiều thời gian hơn để thực hiện trình phân lớp trong giai đoạn đóng gói.

Kết quả thời gian thực hiện của thuật toán IARM-E xấp xỉ bằng thuật toán KGIRD-M.

2.3. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi tập đối tượng thay đổi giá trị

Trong các bài toán thực tế, không chỉ xem xét tập đối tượng thay đổi như bổ sung và loại bỏ mà còn xảy ra các trường hợp thay đổi giá trị của đối tượng tại các thuộc tính của nó. Trong mục này, luận án đề xuất công thức gia tăng tính khoảng cách trong trường hợp tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập đối tượng thay đổi giá trị.

2.3.1. Công thức gia tăng tính khoảng cách khi tập đối tượng thay đổi giá trị

Cho bảng quyết định $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$ và

$$M(C) = [c_{ij}]_{n \times n}, \quad M(\{d\}) = [d_{ij}]$$

tương ứng là ma trận dung sai trên C và $\{d\}$.

Theo [19], khoảng cách giữa hai tập thuộc tính C và $C \cup \{d\}$ được xác định như sau:

$$\sum^C$$

$$c.d) \quad \begin{pmatrix} & & \\ & 2 & \\ & \vdots & \\ & 4 & \\ U & i=j & ij & ij \\ n^2 & & & \end{pmatrix}$$

Từ đó, luận án xây dựng công thức gia tăng tính khoảng cách trong trường hợp tập đối tượng thay đổi giá trị bởi mệnh đề 2.1 dưới đây.

Mệnh đề 2.1. Cho $IDS = (U, C \cup \{d\})$ với
bảng quyết định
không đầy đủ

$U = \{u_1, u_2, \dots, u_n\}$. Không mất tính tổng quát, giả sử tập đối tượng gồm s phần tử
 $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị thành

$\Delta' = \{u'_1, \dots, u'_n\}$. Với $M_{U'}(C) = [c'_{ij}]_{n \times n}$ và $M(C) = [c_{ij}]_{n \times n}$ ($\{d\} = [d_j]_{1 \times n}$) tương ứng là ma

$\{u'_1, \dots, u'_n\}$

trận dung sai trên C và $c'_{i,k}, \dots, c'_{i,k+s-1}$ bị thay đổi giá trị $\{d\}$, khi đó các phần tử

thành c^1, \dots, c^k với $i = k..(k+s-1)$. $(C, C \cup \{d\})$ là khoảng cách
Giả sử D

$$i,k \quad i,k+1-s \quad U'$$

sau khi cập nhật tập đối tượng ΔU và $D_U(C, C \cup \{d\})$ là công thức khoảng cách

trước khi cập nhật. Khi đó, công thức tính gia tăng khoảng cách như sau:

$$\begin{aligned} D \cdot (C, C \cup \{d\}) &= D(C, C \cup \{d\}) + \\ &\cup \{d\} + \\ &\left(c^1 - c_{i=k, j=1}^{2^{k+s-1} n} \right. \\ &\quad \left. \sum_{U}^{ij} ij \right) \quad (2.5) \end{aligned}$$

n^2

Chứng minh

Theo công thức tính khoảng cách (2.4), ta có:

$$\begin{aligned} \frac{D}{L} \cdot (C, C \cup \{d\}) &= \left(c \cdot d \right) + \dots + \left(c - c \cdot d \right) \\ &\quad \sum_{U}^{ij} ij - \sum_{j=I}^{n^2} ij \\ &= I \left(\sum_{j=I}^{k+s-1} \sum_{j=I}^{n^2} ij \right) \\ &= \overline{\sum_{j=I}^{n^2} ij} \left(c_{II} - c_{II} \cdot d_{II} \right) + \dots + \sum_{j=k}^{k+s-1} \left(c_{Ij} - c_{Ij} \cdot d_{Ij} \right) + \dots + \left(c_{In} - c_{In} \cdot d_{In} \right) + \dots + \\ &\quad + \sum_{j=1}^k \sum_{j=1}^{n^2} ij \quad \square \\ &\quad + \sum_{j=1}^{-1} \sum_{j=1}^{n^2} ij \quad \square \end{aligned}$$

$$\begin{aligned}
& c' \cdot d \\
& \left(c' \right) + \left(c \cdot d \right) + \dots + \left(c^{k+s-1} \right) + \dots + \left(c - c \cdot d \right) \\
& \left(c' \right) = \sum_{i=k}^n \sum_{j=1}^{n-1} \left| \begin{array}{ccccccccc} ij & & ij & & nI & & nI & & nI \\ & & & & nj & & nj & & nj \\ & & & & nn & & nn & & nn \end{array} \right| \\
& = \sum_{i=k}^n \sum_{j=1}^{n-1} \left| \begin{array}{ccccccccc} i=k & j=1 & & & j=k & & & & k+s-1 \\ & & & & & & & & \\ & & & & k+s-1 & & & & k+s-1 \\ & & & & k+s-1 & & & & k+s-1 \end{array} \right| \\
& = \sum_{i=k}^n \sum_{j=1}^{n-1} \left(c_{II} - c_{II} \cdot d_{II} \right) + \dots + \sum_{j=k}^{n-1} \left(c_{IJ} - c_{IJ} \cdot d_{IJ} \right) + \dots + \left(c_{In} - c_{In} \cdot d_{In} \right) + \sum_{j=k}^{n-1} \left(c_{Ij} - c_{Ij} \cdot d_{Ij} \right) - \sum_{j=k}^{n-1} \left(c_{IJ} - c_{IJ} \cdot d_{IJ} \right) \\
& + \sum_{i=k}^n \sum_{j=1}^{n-1} \left| \begin{array}{ccccccccc} c \cdot d & & & & c \cdot d & & & & c \cdot d \\ \square & \square & & & \square & & & & \square \\ \square & \square & & & \square & & & & \square \\ \vdots & \vdots & & & \vdots & & & & \vdots \end{array} \right| + \dots + \left(c' \right) - \sum_{i=k}^n \sum_{j=1}^{n-1} \left| \begin{array}{ccccccccc} c & & & & c & & & & c \\ \square & \square & & & \square & & & & \square \\ \square & \square & & & \square & & & & \square \\ \vdots & \vdots & & & \vdots & & & & \vdots \end{array} \right| \\
& \sum^n \left(c \right)
\end{aligned}$$

$$\sum_{n^2} \sum_{ij}^{ij}$$

$$_{i=I\,j=I}^{i=k\,\,\,j=l}$$

$$= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \left(\begin{array}{c} c \\ ij \\ ij \\ ij \end{array} \right)^{c.d} + \frac{2^{k+s-l-n}}{n^2} \left(\begin{array}{c} c' \\ ij \\ ij \\ ij \end{array} \right)^{-c} (1-d)$$

$$= D(C, C \cup \{d\}) + \frac{2^{k+s-l-n}}{n^2} \left(\begin{array}{c} c' - c \\ ij \\ ij \\ ij \end{array} \right) (1-d)$$

Mệnh đề đã được chứng minh.

Ví dụ 2.1. Xét bảng quyết định không đầy đủ

$$U = \{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9\}, \quad C = \{c_1, c_2, c_3, c_4\}$$

được biểu diễn thông tin ở bảng

, 2.6(a) dưới đây.

đây.

Tập đối tượng thay đổi giá trị $\Delta U = \{u_7, u_8, u_9\}$, các giá trị của đối tượng

thay đổi thành các giá trị mới như sau:

$$u_7(c_3) = \text{"Đầy đủ"},$$

$$u_8(c_2) = \text{"Cao"} \text{ và}$$

$$u_9(c_4) = \text{"Cao"}.$$

Bảng 2.6(a). Biểu diễn thông tin về các ô tô

Ô tô	Đơn giá	Km đã đi	Kích thước	Tốc độ tối đa	Gia tốc
	c_1	c_2	c_3	c_4	d
u_1	Cao	Cao	Đầy đủ	Thấp	Tốt
u_2	Thấp	*	Đầy đủ	Thấp	Tốt
u_3	*	*	Gọn nhẹ	Cao	Xấu
u_4	Cao	*	Đầy đủ	Cao	Tốt
u_5	*	*	Đầy đủ	Cao	Tuyệt hảo
u_6	Thấp	Cao	Đầy đủ	*	Tốt
u_7	Cao	Thấp	Gọn nhẹ	Cao	Rất tốt
u_8	Cao	Trung bình	Đầy đủ	Thấp	Tuyệt hảo
u_9	Cao	Cao	Đầy đủ	Trung bình	Tuyệt hảo

- Tính ma trận $M_U(C)$ và

$$M(\{d\})$$

$$\begin{array}{c} |1 \\ |0 \end{array} \quad \begin{array}{c} 0 \\ 0 \end{array} \quad \begin{array}{c} |1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ |1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{array}{c} | \\ |0 \\ |0 \end{array} \quad \begin{array}{c} | \\ 0 \\ 0 \end{array} \quad \begin{array}{c} | \\ |0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ |1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{array}$$

$$\begin{array}{c} | \\ M_U(C) = |0 \\ | \\ |0 \\ |0 \end{array} \quad \begin{array}{c} | \\ 0 \\ | \\ 0 \\ 0 \end{array} \quad \begin{array}{c} | \\ |0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ |1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ |0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}$$

$$\begin{array}{c} | \\ |0 \\ |0 \end{array} \quad \begin{array}{c} | \\ 0 \\ 1 \end{array} \quad \begin{array}{c} | \\ |0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ |0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}$$

$$\begin{array}{c} | \\ | \\ | \end{array} \quad \begin{array}{c} | \\ | \\ | \end{array} \quad \begin{array}{c} | \\ | \\ | \end{array}$$

- Tính khoảng cách sinh bởi C và $C \cup \{d\}$ trên U

$$\sum_{c,d}^{} \left(\sum_{c,d}^{} \right) = \frac{1}{L} \sum_{i=1}^{L-1}$$

) =⁶

$$\begin{array}{c} U \\ n^2 \end{array} \quad \begin{array}{ccccccc} ij & ij & ij & ij \\ i=1 & j=1 & i=1 & j=1 \end{array} \quad \begin{array}{c} 92 \\ \parallel \\ ij \end{array} \quad \begin{array}{c} ij \\ ij \end{array} \quad \begin{array}{c} 8I \\ ij \end{array}$$

- Khi tập đổi tượng thay đổi giá trị $\Delta U = \{u_7, u_8, u_9\}$ trên bảng dữ liệu mới trị được biểu diễn ở bảng 2.6(b) dưới đây.

Bảng 2.6(b) Biểu diễn thông tin về các ô tô sau khi đã thay đổi giá trị

$\hat{O}_{tô}$	<i>Đơn giá</i>	<i>Km đã đi</i>	<i>Kích thước</i>	<i>Tốc độ tối đa</i>	<i>Gia tốc</i>
	c_1	c_2	c_3	c_4	d
u_1	Cao	Cao	Đầy đủ	Thấp	Tốt
u_2	Thấp	*	Đầy đủ	Thấp	Tốt
u_3	*	*	Gọn nhẹ	Cao	Xấu
u_4	Cao	*	Đầy đủ	Cao	Tốt
u_5	*	*	Đầy đủ	Cao	Tuyệt hảo
u_6	Thấp	Cao	Đầy đủ	*	Tốt
u_7	Cao	Thấp	Đầy đủ	Cao	Rất tốt
u_8	Cao	Cao	Đầy đủ	Thấp	Tuyệt hảo
u_9	Cao	Cao	Đầy đủ	Cao	Tuyệt hảo

- Tính ma trận $M(C) = [c^i]$

$$M(C) = [c^i] = \begin{bmatrix} ij \end{bmatrix}_{n \times n} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ | & & & & & & & & \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad \square$$

$$= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ | & & & & & & & & \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \square$$

$$= \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \end{vmatrix}$$

$$U^{\ast} \quad \left[\begin{smallmatrix} -ij \\ n \times n \end{smallmatrix} \right]$$

$$\left| \begin{array}{cccccccccc} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{array} \right|$$

$$\lfloor$$

- Tính $\sum_{i=k}^2 \sum_{j=l}^{i,j} \frac{2^{k+s-l-n}}{n} (c' - c) \binom{i,j}{I-d} \binom{i,j}{I-d}$

$\binom{i,j}{I-d} = \frac{2^{k+s-l-n}}{n} (c' - c)$

$$\sum_{i=k}^2 \sum_{j=l}^{i,j} \frac{2^{k+s-l-n}}{n} \binom{i,j}{I-d} \binom{i,j}{I-d} = 2^2 \cdot 4 = 8$$

$i=k \quad j=l$

- Tính khoảng cách $D \cdot (C, C \cup \{d\})$

sau khi cập nhật tập đối tượng ΔU

$$\sum (c' \cdot d)^U = \sum (c' \cdot d) = \sum (c' \cdot d) = 14$$

Vậy $D \equiv D \cdot (C, C \cup \{d\}) = \frac{14}{i=l \ j=l} = i=j \cdot 9^2$

$$\left(C, C \cup \{d\} \right) + \frac{2}{c} \left(c - \right) \left(I - d \right)$$

$$U^2 \sum_{i=k}^2 \sum_{j=1}^n \begin{matrix} i,j \\ i,j \\ i,j \end{matrix}$$

Mệnh đề 2.2. Cho $IDS = (U, C \cup \{d\})$ bảng quyết định không đầy đủ với

$$U = \{u_1, u_2, \dots, u_n\}.$$

Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$

bị thay đổi giá trị thành $\Delta U' = \{u'_1, u'_2, \dots, u'_n\}$. Với $M_U(C) = [c_{ij}]_{n \times n}$ và

$M_U(\{d\}) = [d_{ij}]$ tương ứng là ma trận dung sai trên C, giả sử các phần tử $c'_1, \dots, c'_{k-1}, c'_k, \dots, c'_{k+s-1}$ bị thay đổi giá trị thành c với $i = k \dots (k+s-1)$. Giả

$D_U(C, C \cup \{d\})$ là khoảng cách sau khi cập nhật và là công thức

khoảng cách trước khi cập nhật tập đối tượng ΔU . Khi đó ta có:

$$D^-(C, C \cup \{d\}) = D_U(C, C \cup \{d\})$$

2) $\sum_{i=1}^k c_i = c$ với mọi $k \leq i \leq k+s-1$, $1 \leq j \leq n$, nghĩa là $c_j \in S$ (u), thì

$$D_{\mathcal{C}}(C, C \cup \{d\}) = D_U(C, C \cup \{d\})$$

Chứng minh. Dễ dàng suy ra từ công thức của mệnh đề 2.1.

2.3.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi tập đổi tượng thay đổi giá trị

Mệnh đề 2.3. Cho $IDS = (U, C \cup \{d\})$ với bảng quyết định không đầy đủ

và $R \subseteq C$ là tập rút gọn dựa trên khoảng cách. Giả sử tập đổi

$$U = \{u_1, u_2, \dots, u_n\}$$

tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị

thành $\Delta U' = \{u^{'}, u^{'}, \dots, u^{'}\}$ và U' là tập đổi

tượng sau khi thay đổi giá trị. Với tương ứng là ma trận dung sai trên C, giả sử

$$M_U(C) = [c_{ij}] \quad \text{và} \quad M_U(\{d\}) = [d_{ij}]$$

các phần tử c, \dots, c bị thay đổi giá trị thành $c^{'}, \dots, c^{'}$ với

$$\begin{matrix} & i,k \\ & i, k+1-s \\ & \vdots \\ & n \\ M_U(C) = [c_{ij}] & \end{matrix} \quad \text{và} \quad \begin{matrix} & i,k \\ & i, k+1-s \\ & \vdots \\ & n \\ M_U(\{d\}) = [d_{ij}] & \end{matrix}$$

$i = k..(k + s - 1)$. Khi đó ta có:

Nếu $d = 1$ hoặc $c^{'}, c$ với mọi $k \leq i \leq k + s - 1, 1 \leq j \leq n$ thì R là tập rút gọn

của $IDS' = (U', C \cup \{d\})$

Chứng minh.

với mọi $k \leq i \leq k + s - 1$, $1 \leq j \leq n$

Theo mệnh đề 2.1, Nếu $d = 1$

hoặc $c' = c$

$$\text{thì } D_{\cdot}^{\cdot}(C, C \cup \{d\}) = D_{\cdot}^{\cdot}(C, C \cup \{d\})$$

$\overset{ij}{\text{với}}$ $\overset{ij}{\text{là khoảng cách}} \\ \text{sau khi}$

$$D_U(C, C \cup \{d\})$$

cập nhật và

$$D_U(C, C \cup \{d\})$$

là công thức khoảng cách
trước khi cập nhật tập đối

tượng

$$\Delta U.$$

$$D_U(R, R \cup \{d\}) = D_U(C, C \cup \{d\}) =$$

Do R là tập rút gọn của IDS
nên

$$D_{\cdot}^{\cdot}(C, C \cup \{d\})$$

U

và $\forall r \in R, D_{\cdot}^{\cdot}(R - \{r\}, (R - \{r\}) \cup \{d\}) \neq D_{\cdot}^{\cdot}(C, C \cup \{d\})$. Theo định nghĩa tập rút gọn dựa
trên

khoảng cách, R là tập rút gọn của

$$IDS' = (U', C \cup \{d\}).$$

Như đã đề cập ở trên, thuật toán FSMV [69] và thuật toán Object-R [86]
là các thuật toán gia tăng tìm tập rút gọn theo tiếp cận lọc trong trường hợp
tập đối tượng thay đổi giá trị. Cả hai thuật toán này đều theo hướng tiếp cận
lọc truyền thống. Với hướng tiếp cận này, tập rút gọn thu được là tập thuộc
tính

bảo toàn khoảng cách ban đầu. Độ chính xác phân lớp được tính toán sau khi thu được tập rút gọn.

Trong mục này, luận án đề xuất thuật toán gia tăng tìm tập rút gọn theo tiếp cận lọc - đóng gói. Thuật toán đề xuất của luận án bao gồm hai giai đoạn: giai đoạn lọc và giai đoạn đóng gói.

Giai đoạn lọc: Giả sử tập R là tập rút gọn của tập thuộc tính ban đầu và bị thay đổi giá trị thành tập ΔU . Đối với mỗi tập đối tượng ΔU đối tượng

$a_i \in C - R$, tính độ quan trọng của a_i theo công thức gia tăng trong định nghĩa 2.2

và chọn thuộc tính a_i có độ quan trọng lớn nhất, được ký hiệu là a_{\max} , a_{\max} được thêm vào R và R được lưu trữ trong danh sách các ứng viên tập rút gọn. Quá trình này được lặp lại cho đến khi R thỏa mãn điều kiện của tập rút gọn. Tiếp theo, thuật toán kiểm tra các thuộc tính dư thừa trong tập rút gọn R thu được và thuật toán thực hiện giai đoạn đóng gói.

Giai đoạn đóng gói: Thực hiện bộ phân lớp trên mỗi phần tử (t_j) trong danh sách ứng viên và chọn phần tử có độ chính xác phân lớp tối đa làm đầu ra của thuật toán.

Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong bảng quyết định không đầy đủ sử dụng khoảng cách khi tập đối tượng thay đổi giá trị được mô tả như sau:

Thuật toán FWIA_U_Obj (Filter-Wrapper Incremental Algorithm for

Attribute Reduction in Incomplete Decision Tables when Update Objects).

Đầu vào: Cho bảng quyết định không đầy đủ

$IDS \sqsubseteq U, C$

v

$U \sqsubseteq u_1, u_2, \dots, u_n \sqsubseteq$

$\sqcup \{d\} \sqsubseteq$

ó

i

- Tập rút gọn $R \sqsubseteq C$.

- Ma trận dung sai $M_U(R)$, $M_U(C)$ và $M_U(\{d\})$

- Tập đối tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$

bị thay đổi giá trị thành $U' = \{u^{'}, u^{'}, \dots, u^{'}\}$ } U' là tập đối tượng sau khi đổi giá trị.

Đầu ra: Tìm tập rút gọn R_{best} trên $IDS' = (U', C \cup \{d\})$;

Bước 1: Khởi tạo và kiểm tra

1. $T := \emptyset$; // T chứa các ứng viên của tập rút gọn

2. Tính $M_U(R)$, $M_U(C)$ và $M_U(\{d\})$
ma trận dung sai M_U

3. If $d = 1$ or $c' = c$ for any U $k \leq i \leq k + s - 1, 1 \leq j \leq n$
then
Return R ;

Bước 2: Tìm tập rút gọn

4. Tính độ đo khoảng cách

$$D_U(R, R \cup \{d\}), D_U(C, C \cup \{d\})$$

5. Tính độ đo $D_U(R, R \cup \{d\})$ sử dụng
khoảng cách

$$D_U(R, R \cup \{d\})$$

$$\}), D_U$$

U U

công thức gia tăng trong mệnh đề 2.1.

//Loại bỏ các thuộc tính dư thừa trong R (nếu có)

6. For each $a \in R$

7. If

$$D_U(R - \{a\}, (R - \{a\}) \cup \{d\}) = D_U(C, C \cup \{d\})$$

then $R := R - \{a\}$;

//Giai đoạn lọc, tìm các ứng viên cho tập rút gọn

//Bổ sung các thuộc tính còn lại vào R

8. $j := 0;$

9. Repeat

10. $j := j + 1;$

```

1   For each  $r \in C$ 
1      $\exists R$ 
.
    Tính  $SIG$ 
1     Chọn  $r_m \in C \cap R$  sao cho  $SIG_R[r_m] \cap max_{r \in A \cap R}$ 
1        $\exists SIG_R[r_m]$ ;
1      $R := R \setminus r_m$ 
4   };
.
1    $T := T$ 
1    $\forall T$ :
1     Until  $L_v \subseteq R \subseteq dL_v \subseteq C, C \neq$ 
.
// Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất

```

1). Tính độ chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp kiểm tra chéo 10-fold;

- $R_{best} \subseteq T_k$; //với $T_k (1 \leq k \leq t)$ có độ chính xác phân lớp cao nhất.
- Return R_{best} .

Đánh giá độ phức tạp của thuật toán FWIA_U_Obj

Ký hiệu $C, U, \Delta U$

tương ứng là số thuộc tính
điều kiện, số đối tượng và

số đối tượng thay đổi giá trị. Ở câu lệnh 2, độ phức tạp tính ma trận dung sai

$M(R)$ khi biết

$M(R)$ là $O(\Delta U * (U + \Delta U))$. Độ phức tạp
của vòng lặp Repeat ở

câu lệnh số 4 là $O(\Delta U * (U + \Delta U))$.

Trong trường hợp tốt nhất, thuật toán kết thúc ở câu lệnh 3 (tập rút gọn

không thay đổi). Khi đó, độ phức tạp thuật toán FWIA_U_Obj là $O(\Delta U * (U$

$+ \Delta U))$. Ngược lại, độ phức tạp tính khoảng cách theo công thức gia

tăng trong câu lệnh 5 khi biết ma trận $O(\Delta U * (U + \Delta U))$. Xét vòng
dung sai là

lặp Repeat từ câu lệnh $SIG_R(r)$ ta phải tính $D \cdot (R, R \cup \{d\})$
8 đến 14, để tính

$$E_U(R \cup \{r\}, R \cup \{r\} \cup \{d\}) = D \cdot (R, R \cup \{d\})$$

) vì

đã được tính ở bước trước.
Độ phức

tập tính gia tăng $D \cdot (R \cup \{r\}, R \cup \{r\} \cup \{d\})$ là $O(\Delta U * (U + \Delta U))$. Do đó, độ phức

U
và độ phức tập của

tập của vòng lặp Repeat là $O((C - R$

$$)^{2 * |\Delta U| * (|U| + |\Delta U|)}$$

giai đoạn lọc trong trường hợp xấu nhất là

$$O((C - R)^{2 * |\Delta U| * (|U| + |\Delta U|)})$$

Độ phức tạp của giai đoạn đóng gói phụ thuộc vào độ phức tạp của bộ phân lớp được sử dụng. Giả sử độ phức tạp của bộ phân lớp là $O(f(n))$, khi đó

độ phức tạp của giai đoạn đóng gói là $O(|C| - |R|)^* f(n)$. Vì vậy, độ phức tạp của thuật toán FWIA_U_Obj là

$$O\left(\left(|C| - |R|\right)^2 * |\Delta U|^2 * (|U| + |\Delta U|)\right) + O\left(|C| - |R|\right)^* f(n)$$

Nếu thực hiện thuật toán không gia tăng lọc - đóng gói IDS_FW_DAR[3] trực tiếp trên bảng quyết định có số đối tượng $U \cup \Delta U$, độ phức tạp của thuật toán IDS_FW_DAR là

$$O\left(|C|^2 * (|U| + |\Delta U|)^2\right) * O\left(|C| * f(n)\right). Do$$

đó, thuật toán

gia tăng FWIA_U_Obj giảm thiểu đáng kể độ phức tạp thời gian thực hiện, đặc biệt trong trường hợp $|U|$ lớn hoặc $|R|$ lớn.

2.3.3. Thực nghiệm, đánh giá thuật toán FWIA_U_Obj

2.3.3.1. Mục tiêu thực nghiệm

Mục tiêu thực nghiệm là đánh giá tính hiệu quả của thuật toán dựa trên các tiêu chí: *số lượng thuộc tính trong tập rút gọn, độ chính xác phân lớp và thời gian thực hiện*. Thuật toán FWIA_U_Obj được so sánh với thuật toán FSMV

[69] và thuật toán Object-R [86]. FSMV là thuật toán gia tăng tìm tập rút gọn trong bảng quyết định không đầy đủ theo tiếp cận lọc trong trường hợp tập đối tượng thay đổi giá trị sử dụng miền dương. Trong khi đó, Object-R là thuật toán gia tăng tìm tập rút gọn trong bảng quyết định không đầy đủ theo tiếp cận lọc trong trường hợp tập đối tượng thay đổi giá trị sử dụng độ đo không nhất quán.

2.3.3.2. Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Tiến hành thực nghiệm trên 06 bộ dữ liệu được lấy trong kho dữ liệu UCI [73] như mô tả ở bảng 2.7. Mỗi tập dữ liệu được chia ngẫu nhiên thành hai phần xấp xỉ bằng nhau: Tập dữ liệu không thay đổi được ký hiệu là O_{ori} và tập dữ liệu bị thay đổi được ký hiệu là O_{chan} . Tiếp theo, tập dữ liệu bị thay đổi O_{chan} được chia thành năm phần bằng nhau được ký hiệu lần lượt là O_1, O_2, O_3, O_4, O_5 . Với tập dữ liệu O_{chan} , thực hiện cập nhật ngẫu nhiên giá trị thuộc tính của các đối tượng bị thay đổi, bảo đảm nguyên tắc các giá trị bị thay đổi thuộc miền giá trị của thuộc tính ban đầu.

Trong bảng 2.7, các cột $|O|$, $|O_{ori}|$, $|O_{chan}|$, $|A|$, $|k|$ được ký hiệu tương ứng là: Số đối tượng; Số đối tượng trong O_{ori} ; Số đối tượng trong O_{chan} ; Số thuộc tính điều kiện; Số lớp quyết định.

Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold, nghĩa là bộ dữ liệu được chia thành 10 phần xấp xỉ bằng nhau, lấy lần lượt 1 phần làm bộ dữ liệu kiểm tra, 9 phần còn lại làm dữ liệu huấn luyện. Quá trình được lặp lại 10 lần.

Môi trường thực nghiệm: Thực nghiệm được thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel® Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

Bảng 2.7. Các bộ dữ liệu sử dụng trong thực nghiệm khi tập đối tượng thay đổi giá trị

STT	Tập dữ liệu	$ O $	$ O_{ori} $	$ O_{chan} $	$ A $	$ k $
1	Audiolgy.data	226	116	110	69	24
2	Soybean-laarge.data	307	157	150	35	2
3	house-votes-84.data	435	220	215	16	2
4	Arrhythmia.data	452	222	230	279	16

5	Anneal.data	798	393	405	38	6
6	Ad.data	3279	1644	1635	1558	2

2.3.3.3. Kịch bản thực nghiệm

Trước hết, luận án thực hiện cài đặt và chạy 03 thuật toán FWIA_U_Obj, FSMV [69] và Object-R [86] khi lần lượt đưa vào các tập đổi tượng thay đổi giá trị O_1, O_2, O_3, O_4, O_5 . Sau đó, các giá trị số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện được ghi lại.

2.3.3.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 2.8 trình bày kết quả về số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp của các thuật toán FWIA_U_Obj, FSMV và Object-R. Trong đó cột $|R|$ và Acc lần lượt là số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp.

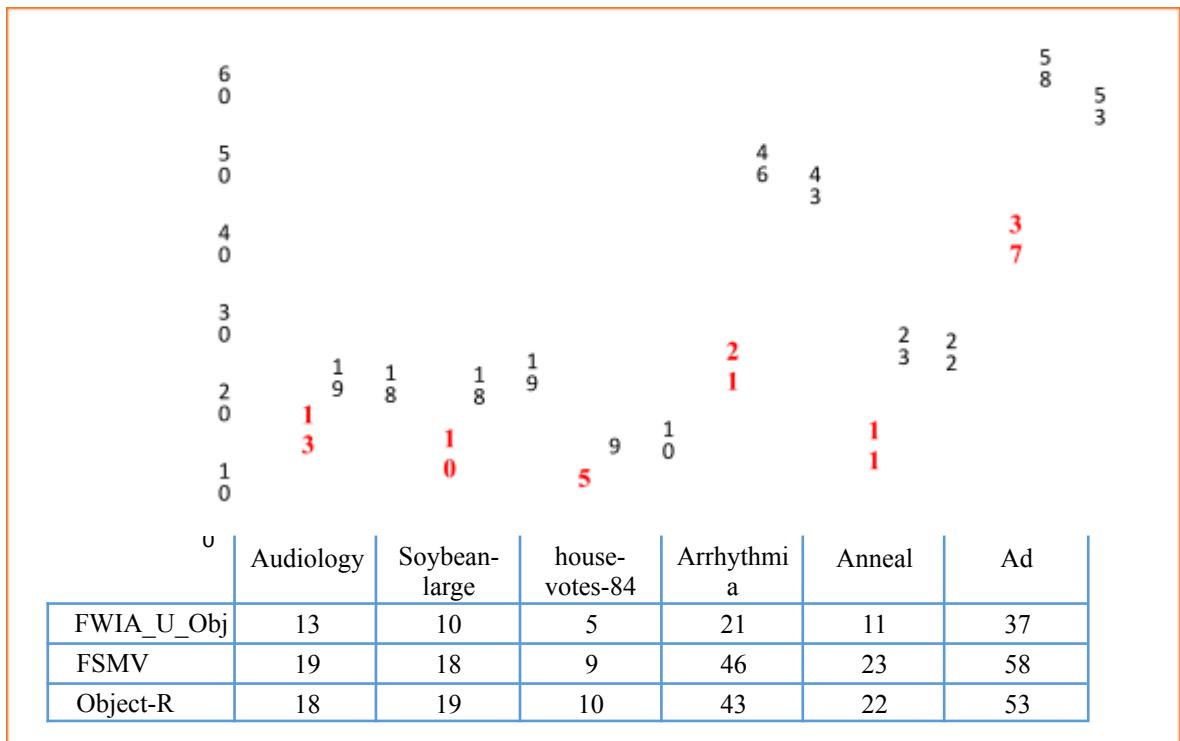
Dựa trên kết quả trong bảng 2.8 và nhìn trực quan vào hình 2.1(b) ta thấy rằng độ chính xác phân lớp của thuật toán FWIA_U_Obj cao hơn một chút so với FSMV và Object-R trên tất cả các tập dữ liệu và trên tất cả các bước lặp khi đưa lần lượt các tập đổi tượng thay đổi giá trị O_1, O_2, O_3, O_4, O_5 .

Hơn nữa, nhìn trực quan vào hình 2.1(a), số lượng thuộc tính trong tập rút gọn thu được bởi FWIA_U_Obj nhỏ hơn nhiều so với FSMV và Object-R, đặc biệt là trong tập dữ liệu có nhiều thuộc tính như Ad.data. Có thể thấy rằng thuật toán Object-R hiệu quả hơn một chút so với thuật toán FSMV về cả độ chính xác của phân lớp và số lượng thuộc tính trong tập rút gọn.

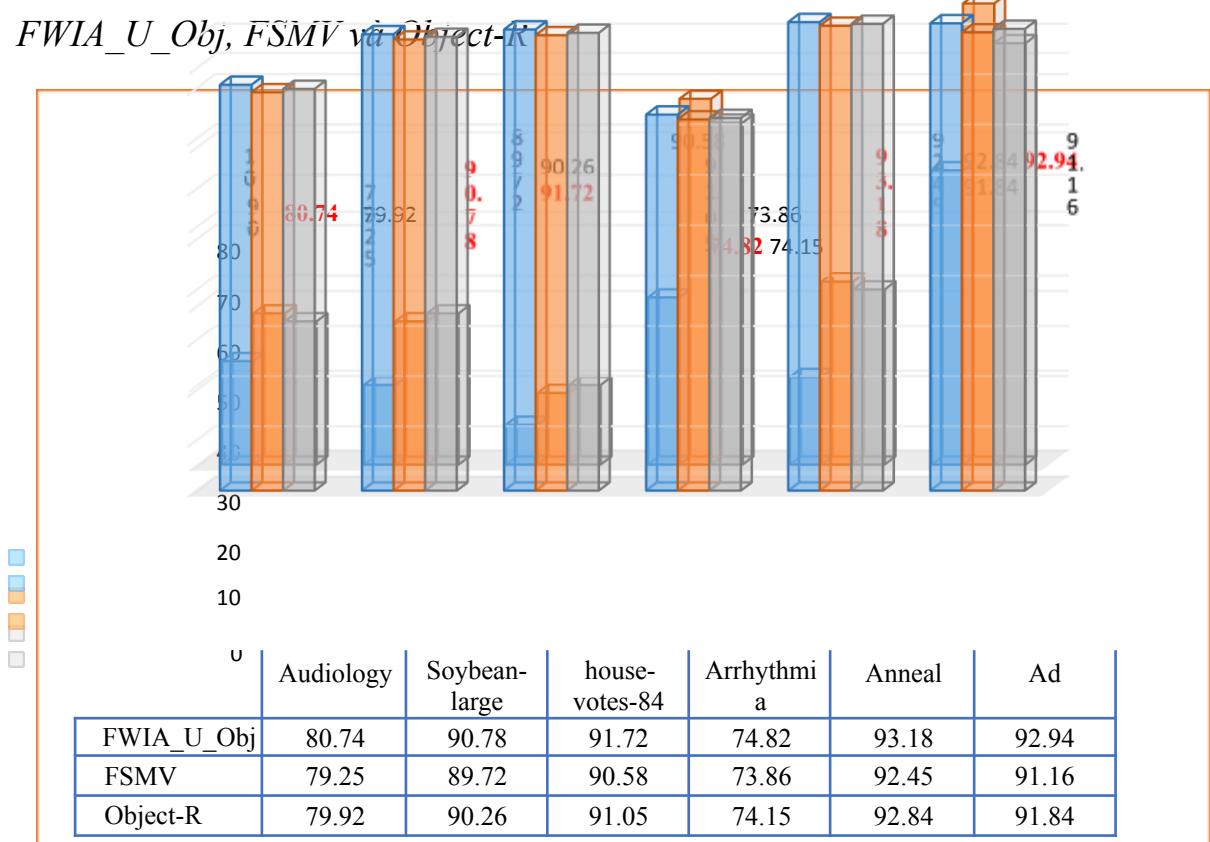
Do đó, mô hình phân lớp dựa trên tập rút gọn của thuật toán FWIA_U_Obj hiệu quả hơn mô hình phân lớp của thuật toán FSMV và thuật toán Object-R về chất lượng phân lớp và độ phức tạp của mô hình.

Bảng 2.8. Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của ba thuật toán FWIA_U_Obj, FSMV và Object-R

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		FSMV		Object-R	
			R	Acc	R	Acc	R	Acc
1	Audiology	O_1	11	78.12	18	78.06	17	77.92
		O_2	12	79.24	23	78.84	22	78.16
		O_3	9	75.46	12	72.46	13	73.45
		O_4	14	81.28	26	80.72	24	80.27
		O_5	13	80.74	19	79.25	18	79.92
2	Soybean-large	O_1	8	91.12	14	90.23	13	90.46
		O_2	9	92.54	16	91.17	16	91.17
		O_3	6	88.56	15	87.48	15	87.48
		O_4	11	89.23	22	89.24	21	90.15
		O_5	10	90.78	18	89.72	19	90.26
3	house-votes-84	O_1	5	92.36	10	92.05	9	91.84
		O_2	6	93.84	12	92.18	11	92.82
		O_3	8	94.15	12	93.46	12	93.46
		O_4	6	92.87	11	92.14	11	92.14
		O_5	5	91.72	9	90.58	10	91.05
4	Arrhythmia	O_1	22	72.18	41	71.24	38	71.69
		O_2	24	73.45	45	72.92	42	72.28
		O_3	18	71.26	51	71.02	46	70.89
		O_4	15	69.18	34	68.72	31	68.06
		O_5	21	74.82	46	73.86	43	74.15
5	Anneal	O_1	6	92.08	14	90.46	13	91.15
		O_2	8	93.16	18	92.95	18	92.54
		O_3	13	91.85	25	91.05	24	91.24
		O_4	9	89.28	17	88.48	17	88.95
		O_5	11	93.18	23	92.45	22	92.84
6	Ad	O_1	25	90.18	54	89.15	52	89.82
		O_2	32	91.23	61	90.68	55	90.23
		O_3	24	86.72	65	85.18	59	86.04
		O_4	36	92.55	54	91.45	51	91.11
		O_5	37	92.94	58	91.16	53	91.84



Hình 2.1(a): Số lượng thuộc tính tập rút gọn của ba thuật toán FWIA_U_Obj, FSMV và Object-R



Hình 2.1(b): Độ chính xác phân lớp của ba thuật toán FWIA_U_Obj, FSMV và Object-R

2.3.3.5. Đánh giá thời gian thực hiện

Thời gian thực hiện của thuật toán FWIA_U_Obj, FSMV và Object-R (tính theo giây), trong đó các cột *RT*, *Total RT* lần lượt là thời gian thực hiện, tổng thời gian thực hiện, được trình bày trong bảng 2.9 dưới đây

Bảng 2.9. Thời gian thực hiện của ba thuật toán FWIA_U_Obj, FSMV và Object-R (tính bằng giây)

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		FSMV		Object-R	
			RT	Total RT	RT	Total RT	RT	Total RT
1	Audiology	O_1	1.25	1.25	0.86	0.86	0.95	0.95
		O_2	1.38	2.63	0.92	1.78	1.02	1.97
		O_3	1.24	3.87	1.05	2.83	1.16	3.13
		O_4	1.64	5.51	1.16	3.99	1.25	4.38
		O_5	1.34	6.85	0.92	4.91	1.06	5.44
2	Soybean-large	O_1	0.86	0.86	0.54	0.54	0.59	0.59
		O_2	0.94	1.80	0.68	1.22	0.72	1.31
		O_3	1.06	2.86	0.75	1.97	0.82	2.13
		O_4	1.12	3.98	0.84	2.81	0.89	3.02
		O_5	0.85	4.83	0.68	3.49	0.75	3.77
3	house-votes-8 4	O_1	0.84	0.84	0.72	0.72	0.78	0.78
		O_2	0.63	1.47	0.52	1.24	0.59	1.37
		O_3	0.72	2.19	0.58	1.82	0.65	2.02
		O_4	0.68	2.87	0.49	2.31	0.52	2.54
		O_5	0.59	3.46	0.42	2.73	0.56	3.10
4	Arrhythmia	O_1	3.24	3.24	2.86	2.86	2.92	2.92
		O_2	3.65	6.89	2.95	5.81	3.05	5.97
		O_3	3.12	10.01	2.74	8.55	2.82	8.79
		O_4	2.96	12.97	2.25	10.80	2.34	11.13
		O_5	2.85	15.82	2.16	12.96	2.28	13.41
5	Anneal	O_1	0.98	0.98	0.65	0.65	0.72	0.72
		O_2	0.75	1.73	0.52	1.17	0.58	1.30
		O_3	0.86	2.59	0.68	1.85	0.76	2.06
		O_4	0.72	3.31	0.54	2.39	0.62	2.68
		O_5	0.78	4.09	0.57	2.96	0.65	3.33

6	Ad	O_1	7.35	7.35	5.46	5.46	5.82	5.82
		O_2	6.48	13.83	5.11	10.57	5.95	11.77
		O_3	7.84	21.67	6.08	16.65	6.24	18.01
		O_4	6.28	27.95	5.12	21.77	5.89	23.90
		O_5	5.72	33.22	4.86	26.63	5.17	29.07

Trên tất cả các tập dữ liệu trong bảng 2.9, thuật toán FWIA_U_Obj có thời gian thực hiện cao hơn thuật toán FSMV và thuật toán Object-R vì thuật toán FWIA_U_Obj cần nhiều thời gian hơn để thực hiện phân lớp trong giai đoạn đóng gói. Trong khi đó thời gian thực hiện của thuật toán Object-R cao hơn một chút thuật toán FSMV vì thời gian tính độ không nhất quán trong Object-R cao hơn thời gian tính miền dương trong FSMV.

2.3.4. Đánh giá thuật toán FWIA_U_Obj so với việc thực hiện gián tiếp hai thuật toán IDS_IFW_DO và IDS_IFW_AO

2.3.4.1. Mục tiêu thực nghiệm

Để tìm tập rút gọn trong trường hợp tập đối tượng O_i thay đổi giá trị, chúng ta có thể thực hiện phối hợp 2 thuật toán: thuật toán IDS_IFW_DO khi xóa tập đối tượng O_i cũ và thuật toán IDS_IFW_AO tìm tập rút gọn khi bổ sung tập đối tượng O_i mới. Kết quả thử nghiệm để đánh giá tính hiệu quả của thuật toán FWIA_U_Obj so với hướng tiếp cận trước đây là thực hiện đồng thời hai thuật toán: IDS_IFW_DO và IDS_IFW_AO. Việc đánh giá được thực hiện trên thời gian thực hiện và độ chính xác mô hình phân lớp sau rút gọn thuộc tính.

2.3.4.2. Số liệu và môi trường thực nghiệm

Số liệu và môi trường thực nghiệm giống như mô tả trong mục 2.3.3.2.

2.3.4.3. Kịch bản thực nghiệm

Trước hết, thực hiện thuật toán FWIA_U_Obj khi lần lượt các tập đối tượng O_1, O_2, O_3, O_4, O_5 thay đổi giá trị. Sau đó, với mỗi tập đối tượng O_i ($i=1..5$) thay đổi giá trị, thực hiện lần lượt hai thuật toán:

- 1) Thuật toán IDS_IFW_DO khi loại bỏ tập đối tượng cũ
- 2) Thuật toán IDS_IFW_AO khi bổ sung đối tượng mới (O_i).

So sánh hai kết quả hai cách tiếp cận trên thời gian thực hiện và độ chính xác phân lớp.

2.3.4.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Kết quả về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Obj và hai thuật toán IDS_IFW_DO và IDS_IFW_AO được trình bày trong bảng 2.10 dưới đây.

Bảng 2.10. Số lượng tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Obj so với 2 thuật toán IDS_IFW_DO và IDS_IFW_AO

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		IDS_IFW_DO và IDS_IFW_AO	
			R	Acc	R	Acc
1	Audiology	O_1	11	78.12	11	77.98
		O_2	12	79.24	13	79.16
		O_3	9	75.46	9	75.02
		O_4	14	81.28	13	80.98
		O_5	13	80.74	13	79.94
2	Soybean-large	O_1	8	91.12	8	91.56
		O_2	9	92.54	9	92.08
		O_3	6	88.56	5	88.24
		O_4	11	89.23	11	88.96
		O_5	10	90.78	11	89.86
3	house-votes-84	O_1	5	92.36	5	92.82
		O_2	6	93.84	6	93.18
		O_3	8	94.15	7	94.78
		O_4	6	92.87	6	93.05
		O_5	5	91.72	5	92.06
4	Arrhythmia	O_1	22	72.18	23	71.86
		O_2	24	73.45	23	73.08
		O_3	18	71.26	19	71.04
		O_4	15	69.18	16	68.84
		O_5	21	74.82	20	74.15
5	Anneal	O_1	6	92.08	6	91.26
		O_2	8	93.16	8	92.84
		O_3	13	91.85	12	91.17
		O_4	9	89.28	9	88.54
		O_5	11	93.18	11	92.75
6	Ad	O_1	25	90.18	26	89.76
		O_2	32	91.23	30	91.05
		O_3	24	86.72	25	85.98
		O_4	36	92.55	34	92.48
		O_5	37	92.94	36	92.06

Kết quả trong bảng 2.10 cho thấy, số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của hai hướng tiếp cận tính tập rút gọn nêu trên là xấp xỉ bằng nhau. Độ chính xác phân lớp với hướng tiếp cận trực tiếp cải thiện hơn một chút trên tất cả các tập dữ liệu.

2.3.4.5. Đánh giá thời gian thực hiện

Bảng 2.11. Thời gian thực hiện của thuật toán FWIA_U_Obj so với 2 thuật toán IDS_IFW_DO và IDS_IFW_AO (tính bằng giây)

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		IDS_IFW_DO và IDS_IFW_AO	
			Thời gian thực hiện	Tổng Thời gian thực hiện	Thời gian thực hiện	Tổng Thời gian thực hiện
1	Audiology	O_1	1.25	1.25	2.84	2.84
		O_2	1.38	2.63	2.96	5.80
		O_3	1.24	3.87	2.05	7.85
		O_4	1.64	5.51	2.92	10.77
		O_5	1.34	6.85	2.55	13.32
2	Soybean-large	O_1	0.86	0.86	1.48	1.48
		O_2	0.94	1.80	2.05	3.53
		O_3	1.06	2.86	2.18	5.71
		O_4	1.12	3.98	2.34	8.05
		O_5	0.85	4.83	1.84	9.89
3	house-votes-8	O_1	0.84	0.84	1.96	1.96
		O_2	0.63	1.47	1.72	3.68
		O_3	0.72	2.19	1.85	5.53
		O_4	0.68	2.87	1.16	6.69
		O_5	0.59	3.46	1.32	8.01
4	Arrhythmia	O_1	3.24	3.24	5.26	5.26
		O_2	3.65	6.89	5.82	11.08
		O_3	3.12	10.01	5.43	16.51
		O_4	2.96	12.97	4.94	21.45
		O_5	2.85	15.82	5.16	26.61
5	Anneal	O_1	0.98	0.98	1.94	1.94
		O_2	0.75	1.73	1.47	3.41
		O_3	0.86	2.59	1.54	4.95
		O_4	0.72	3.31	1.42	6.37
		O_5	0.78	4.09	1.56	7.93
6	Ad	O_1	7.35	7.35	13.58	13.58
		O_2	6.48	13.83	11.25	24.83
		O_3	7.84	21.67	13.25	28.08
		O_4	6.28	27.95	12.26	40.34
		O_5	5.72	33.22	9.84	50.18

Thời gian thực hiện của hai hướng tiếp cận tính toán được trình bày như trong bảng 2.11. Trên tất cả các tập dữ liệu, thời gian thực hiện thuật toán FWIA_U_Obj tính trực tiếp tập rút gọn nhỏ hơn nhiều so với hướng tiếp cận

tính toán gián tiếp sử dụng thuật toán loại bỏ tập đối tượng IDS_IFW_DO và thuật toán bổ sung tập đối tượng IDS_IFW_AO. Điều đó cho thấy thuật toán FWIA_U_Obj hiệu quả hơn so với cách tiếp cận cũ.

2.4. Kết luận chương 2

Như vậy chương 2 đã nghiên cứu về tập đối tượng thay đổi trong các trường hợp bổ sung, loại bỏ tập đối tượng và tập đối tượng thay đổi giá trị. Cụ thể như sau:

1) Thực nghiệm thuật toán gia tăng lọc - đóng gói tìm tập rút gọn IDS_IFW_AO trong trường hợp bổ sung tập đối tượng và thực nghiệm thuật toán gia tăng lọc - đóng gói IDS_IFW_DO trong trường hợp loại bỏ tập đối tượng.

2) Đề xuất xây dựng công thức gia tăng tính khoảng cách trong trường hợp tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc

- đóng gói FWIA_U_Obj tìm tập rút gọn trên bảng quyết định không đầy đủ trong trường hợp tập đối tượng thay đổi giá trị.

Kết quả thực nghiệm cho thấy, các thuật toán theo tiếp cận lọc - đóng gói giảm thiểu số lượng thuộc tính tập rút gọn và cải thiện độ chính xác của mô hình phân lớp so với các thuật toán gia tăng khác theo tiếp cận lọc đã công bố. Tuy nhiên về thời gian thực hiện cao hơn vì phải cần nhiều thời gian hơn để thực hiện phân lớp trong giai đoạn đóng gói.

Trong trường hợp tập đối tượng thay đổi giá trị, dựa trên kết quả thực nghiệm thu được, thuật toán FWIA_U_Obj hiệu quả hơn so với cách tiếp cận gián tiếp sử dụng thuật toán loại bỏ tập đối tượng IDS_IFW_DO và thuật toán bổ sung tập đối tượng IDS_IFW_AO.

Kết quả nghiên cứu của chương này được công bố trong các công trình [CT4, CT6, CT7], phần: “Danh mục công trình khoa học của luận án”.

CHƯƠNG 3. PHƯƠNG PHÁP RÚT GỌN THUỘC TÍNH TRONG BẢNG QUYẾT ĐỊNH KHÔNG ĐẦY ĐỦ KHI TẬP THUỘC TÍNH THAY ĐỔI

3.1. Mở đầu

Trong chương này, luận án nghiên cứu về tập thuộc tính thay đổi với các trường hợp bổ sung, loại bỏ tập thuộc tính và tập thuộc tính thay đổi giá trị. Trong các bài toán thực tế, các bảng quyết định không đầy đủ thường có số lượng các thuộc tính rất nhiều, luôn thay đổi số lượng thuộc tính và luôn thay đổi giá trị thuộc tính của các đối tượng. Ví dụ trong chuẩn đoán bệnh, các triệu chứng lâm sàng được xem như các thuộc tính ban đầu để bác sĩ chuẩn đoán bệnh. Sau đó, các chỉ số xét nghiệm được xem như các thuộc tính tiếp theo liên tục được bổ sung, cập nhật nhằm hỗ trợ bác sĩ trong việc nâng cao độ chính xác chuẩn đoán. Với dữ liệu tin sinh học, số lượng các thuộc tính ban đầu thu thập được rất lớn. Để xây dựng mô hình phân lớp hiệu quả, ta cần liên tục loại bỏ các thuộc tính dư thừa, không cần thiết. Với các bảng quyết định như vậy, việc áp dụng các thuật toán tìm tập rút gọn theo cách tiếp cận truyền thống gặp thách thức lớn. Đầu tiên, xuất phát từ nghiên cứu thuật toán gia tăng tìm tập rút gọn trên bảng quyết định không đầy đủ có dữ liệu cố định, sau đó nghiên cứu các thuật toán gia tăng tìm tập rút gọn trong các trường hợp bổ xung, loại bỏ tập thuộc tính. Việc nghiên cứu, thực nghiệm các thuật toán nêu trên nhằm hoàn thiện hơn nữa các thuật toán đã công bố, trên cơ sở đó làm tiền đề cho việc xây dựng, đề xuất thuật toán mới giải quyết trường hợp còn lại của bảng quyết định không đầy đủ thay đổi xuất hiện phổ biến trong các bài toán thực tiễn: trường hợp tập thuộc tính thay đổi giá trị.

Cụ thể chương này trình bày các nghiên cứu như sau:

- 1) Nghiên cứu, xây dựng công thức cập nhật khoảng cách trong trường hợp bổ sung, loại bỏ tập thuộc tính và xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập thuộc tính.

2) Để xuất xây dựng công thức cập nhật khoảng cách trong trường hợp tập thuộc tính thay đổi giá trị, trên cơ sở đó để xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập thuộc tính thay đổi giá trị. So sánh hướng tiếp cận rút gọn thuộc tính trực tiếp với hướng tiếp cận gián tiếp thực hiện đồng thời thuật toán loại bỏ tập thuộc tính sau đó thực hiện thuật toán bổ sung tập thuộc tính.

3.2. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung tập thuộc tính.

Trong các bài toán thực tế, các bảng quyết định không đầy đủ thường có số lượng các thuộc tính rất nhiều, luôn thay đổi số lượng thuộc tính, phổ biến là trường hợp bô xung tập thuộc tính. Để giải quyết bài toán đó, mục này luận án nghiên cứu, xây dựng công thức cập nhật khoảng cách trong trường hợp bổ sung tập thuộc tính, trên cơ sở đó xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp bổ sung tập thuộc tính. Sau đó tiến hành thực nghiệm so sánh với các công trình mới nhất để đánh giá thuật toán.

3.2.1. Công thức cập nhật khoảng cách khi bổ sung tập thuộc tính

Trong [1], tác giả xây dựng công thức tính độ đo khoảng cách khi bổ sung tập thuộc tính nhưng không sử dụng phương pháp gia tăng để xem xét phần thay đổi. Trong [2], các tác giả xây dựng công thức tính độ đo khoảng cách khi bổ sung tập thuộc tính bởi công thức sau đây:

$$D(C \cup \cup\{d\}) = \sum_{ij}^n b_{ij}$$

$$B, C \cup B = \sum^n$$

$$\cup\{d\}) =$$

$$D(C, C$$

(3.1)

$i=1 j=1$

Phần này, luận án nghiên cứu xây dựng công thức cập nhật khoảng cách trong trường hợp bổ sung tập thuộc tính có sử dụng phương pháp gia tăng và khác với công thức trong [2], được trình bày cụ thể bởi mệnh đề sau đây:

Mệnh đề 3.1. Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$

$U = \{u_1, u_2, \dots, u_n\}$. Giả sử tập thuộc tính điều kiện B được bổ sung vào C với

$B \cap C = \emptyset$. Đặt

$M(B) = [b_{ij}]_{n \times n}$ là ma trận dung sai trên B .

Khi đó ta có:

$$1) \text{ Nếu } c_{ij} \leq d_{ij} \quad \text{với mọi } 1 \leq i, j \leq n \text{ thì} \quad D(C \cup B, C \cup B \cup \{d\}) = 0$$

$$2) \text{ Nếu } b_{ij} \cdot d_{ij} = 1 \text{ với mọi } 1 \leq i, j \leq n \quad \text{thì} \quad D(C \cup B, C \cup B \cup \{d\}) = 0 \\ \text{và} \quad .c_{ij} = 1$$

$$3) \text{ Nếu } b_{ij} \geq c_{ij} \quad \text{với mọi } 1 \leq i, j \leq n \text{ thì} \quad D(C \cup B, C \cup B \cup \{d\}) = D \\ (C, C \cup \{d\})$$

$$4) \text{ Với mọi } 1 \leq i \leq n, \text{ nếu tồn tại } 1 \leq j \leq n \text{ sao cho } b_{ij} \cdot c_{ij} = 1 \text{ và} \\ \leq n \text{ sao cho } b_{ij} \cdot c_{ij} = 1 \text{ và}$$

$$D(C \cup B, C \cup \{d\}) = D(C, C \cup \{d\}) - \sum_{i=1}^n \sum_{j=1}^{b_{ij}}$$

(3.2)

$$(c - c.b)$$

$$n_2 \quad \quad \quad j=1 \quad \quad \quad ij \quad \quad \quad ij \quad ij$$

Chứng minh

1) $\sum_{i=1}^n c_{ij} \leq d_{ij}$ với mọi $1 \leq i, j \leq n$ thì $S_C(u_i) \subseteq S_{\{d\}}(u_i)$

và

$S_C(u_i) \cap S_B(u_i) \cap S_{\{d\}}(u_i) = S_C(u_i) \cap S_B(u_i)$. Từ đó ta có:

$$D(C) - S \cap \sum^n \cup$$

B,C

$$\cup B$$

$$\cup\{d\}$$

) =

1

$$(u) \Big)$$

$$n_2 \sum_{i=1}^{C \cup B - i} \{d\} - i$$

$$(u) \cap S - (u) - S + (u) \cap S - (u) \cap S + (u) = 0$$

$$n_2 \sum_{i=1}^{C - i} \sum_{j=1}^{B - i} \{d\} - i$$

2. $b_{ij} \cdot c_{ij} = 1$ và $d_{ij} = 1$ với mọi $1 \leq i, j \leq n$ thì với mọi $u_i \in U$ ta có

$$S_C(u_i) \cap S_B(u_i) \subseteq S_{\{d\}}(u_i) \text{ và } S_C(u_i) \cap S_B(u_i) \cap S_{\{d\}}(u_i) = S_C(u_i) \cap S_B(u_i).$$

Từ đó ta có

$$D(C \cup B, C \cup B \cup \{d\}) = 0.$$

3) Từ $b_{ij} \geq$ ta có

$$S_C(u_i) \subseteq S_B(u_i) \text{ và } S_C(u_i) \cap S_B(u_i) = S_C$$

$$\mathcal{C}_{ij} (u_i) \text{ với mọi } u_i \in U.$$

Từ đó ta có:

$$D(C \cup B, C \cup B \cup \{d\}) = 0.$$

$$\cup B, C$$

$$\cup B$$

$$\cup \{d\}$$

$$) = 1$$

$$\cdot \sum^n$$

$$\binom{S}{n}$$

$$n^2_{i=1} \qquad C-i \qquad \qquad B-i \qquad \qquad C-i \qquad \qquad B-i \qquad \qquad \{d\}-i$$

$$= \frac{1}{n} \cdot \sum_{i=1}^n \left(|S(u) \cap S(u)| - |S(u) \cap S(u)| \right) = D(C, C \cup \{d\})$$

- S

\dots

$$n^2 \quad i=1 \quad \dots \quad \{d\} \quad i$$

4) Với mọi $1 \leq i \leq n$ $b_{ij} \cdot c_{ij} = 1$ và $d_{ij} = 0$, khi đó
 , nếu tồn tại $1 \leq j \leq n$ sao cho

$$\begin{aligned} \text{với mọi } u_i \in U & \text{ ta có: } S_C(u_i) \cap S_B(u_i) - S_C(u_i) \\ & (u_i) \cap S_B(u_i) \cap S_{\{d\}} \\ & (u_i) = \end{aligned}$$

$$= \left(|S_C(u_i) - S_C(u_i) \cap S_{\{d\}}| \right) - \left(|S_C(u_i) - S_C(u_i) \cap S_B(u_i)| \right)$$

Từ đó ta có: $) = \frac{1}{n}$

$$D(C \cup B, C)$$

$\cup B$

$$\cup \{d\} \quad \left(\sum_{i=1}^n$$

$$\binom{(u)}{\binom{(u)}{S}}$$

$$n^2 \sum_{i=1}^{C-i} \sum_{j=1}^{B-i} \sum_{k=1}^{C-i} \sum_{l=1}^{B-i} \sum_{m=1}^{\#\{d\}-i}$$

$$=\left(\left(\begin{smallmatrix} S & (u)-S & (u)\cap S \\ & (u) & \end{smallmatrix}\right)-\left(\begin{smallmatrix} (u)-S & (u)\cap S \\ & (u) \end{smallmatrix}\right)\right)$$

$$\frac{1}{\cdot}$$

$$\left(\begin{smallmatrix} S & \end{smallmatrix}\right)$$

$$\sum_n$$

$$_n$$

$$n_2 \sum_{i=1}^{C-i} \sum_{j=1}^{\#\{d\}-i} \sum_{k=1}^{C-i} \sum_{l=1}^{B-i} \sum_{m=1}^{C-i} \sum_{n=1}^{B-i}$$

$$=\left(\begin{smallmatrix} (u)-S & (u)\cap S \\ & (u) \end{smallmatrix}\right)-\left(\left(\begin{smallmatrix} S & (u)-S & (u)\cap S \\ & (u) & \end{smallmatrix}\right)-\right)$$

$$\frac{1}{\cdot}$$

$$\left(\begin{smallmatrix} & \end{smallmatrix}\right)$$

$$\frac{1}{\cdot} \sum_n$$

$$\sum_n S$$

$$_n$$

(u)))

$$n_2 \quad i=1 \quad C \quad i \quad \{d\} \quad i \quad n_2 \quad i=1 \quad C \quad i \quad B \quad i$$

⋮

$$= D(C, C \cup \{d\}) - \frac{1}{\sum^n} \sum (c_i - c_j \cdot b_{ij})$$

$$n_2 \quad i=1 \quad j=1 \quad ij \quad ij \quad ij$$

Ví dụ 3.1. Xét bảng quyết định
không đầy đủ

với
 $IDS = (U, C \cup \{d\})$
 được biểu diễn thông tin ở bảng 3.1 dưới
 $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}, C = \{c_1, c_2, c_3, c_4\}$

đây. Tập thuộc tính bổ sung $B = \{c_5, c_6\}$, với $B \cap C = \emptyset$.

Bảng 3.1. Biểu diễn thông tin về các tivi

Tivi	Đơn giá	Màu sắc	Kích cỡ	Độ phân giải	Tiết kiệm điện năng	Kết nối Internet	Chuất lượng
	c_1	c_2	c_3	c_4	c_5	c_6	d
u_1	Cao	Đen	Lớn	Thấp	Không	Không	Tốt
u_2	Thấp	*	Lớn	Thấp	Có	Có	Tốt
u_3	*	*	Nhỏ	Cao	Không	Không	Xấu
u_4	Cao	*	Lớn	Cao	*	Không	Tốt
u_5	*	*	Lớn	Cao	Có	Có	Tuyệt hảo

u_6	Thấp	Đen	Lớn	*	Có	*	Tốt
-------	------	-----	-----	---	----	---	-----

- Tính các ma trận $M(C)$, $M(B)$, $M(C \cup B)$, $M(\{d\})$

$$\begin{array}{c} [1 \ 0 \ 0 \ 0 \ 0 \ 0] \\ | \\ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \end{array} \quad \begin{array}{c} [1 \ 0 \ 1 \ 1 \ 0 \ 0] \\ | \\ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \end{array}$$

$$M(C) = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix} \quad M(B) = \begin{vmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{vmatrix}$$

$$\begin{array}{c} [0 \ 0 \ 0 \ 1 \ 1 \ 1] \\ | \\ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \end{array} \quad \begin{array}{c} [0 \ 1 \ 0 \ 0 \ 1 \ 1] \\ | \\ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \end{array}$$

$$\begin{array}{c} [1 \ 0 \ 0 \ 0 \ 0 \ 0] \\ | \\ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \end{array} \quad \begin{array}{c} [1 \ 1 \ 0 \ 1 \ 0 \ 1] \\ | \\ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \end{array}$$

$$M(C \cup B) = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{vmatrix} \quad M(d) = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

- Tính khoảng cách theo công thức: $D(C, C \cup \{d\}) = \frac{1}{n^2} \sum \sum$

ta được: D

$$\left(C, \underset{i=1}{\overset{n}{\sum}} C \cup \{d\} \right)_2 = \left(C, \underset{j=1}{\overset{n}{\sum}} C \cup \{d\} \right)_2$$

$$= ^4 \text{và } D(C$$

$$\cup B, C \cup B$$

$$\cup \{d\} \right) =$$

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n (c - c.b)^{36} \\ & - \begin{array}{l} T \\ i \\ n \\ h \end{array} \end{aligned}$$

) ta được:

$$\left(c - c.b \right)^3$$

=

$$n^2 \sum_{i=1}^n \sum_{j=1}^n ij = \frac{2}{ij} = 36$$

- Nhận thấy rằng $b_{56}.c_{56}=1$ và $d_{56}=0$

$$\text{Vậy: } D(C \cup B, C \cup B \cup \{d\}) = \frac{2}{ij} = D(c - c.b)$$

$$(C, C \cup \{d\}) - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n$$

$i=1$

$j=1$

$ij \quad ij \quad ij$

Từ mệnh đề 3.1 và định nghĩa 2.1 về tập rút gọn dựa trên khoảng cách ta có mệnh đề 3.2 sau đây:

Mệnh đề 3.2. Cho $IDS = (U, C \cup \{d\})$ với
bảng quyết định $R \subseteq C$ là tập rút gọn dựa trên khoảng cách.
không đầy đủ

$U = \{u_1, u_2, \dots, u_n\}$ Giả sử tập thuộc

tính điều kiện B được bổ sung vào C $M(B) = [b_{ij}]_{n \times n}$ là ma trận
với $B \cap C = \emptyset$. Đặt

dung sai trên B . với mọi $1 \leq i \leq n, 1 \leq j \leq n$ thì R là tập rút

Khi đó, nếu $b_{ij} \geq$

c_{ij}

gọn của $IDS_I = (U, C \cup B \cup \{d\})$

Chứng minh. Theo mệnh đề 3.1

Nếu $b \geq c$ với $1 \leq i \leq n, 1 \leq j \leq n$ thì: $D(C \cup B, C \cup B \cup \{d\}) = D(C, C \cup \{d\})$.

$$ij \quad ij$$

Do R là tập rút gọn của IDS nên: $D(R, R \cup \{d\}) = D(C, C \cup \{d\}) = D(C \cup B, C \cup B \cup \{d\})$

và $\forall r \in R, D(R - \{r\}, (R - \{r\}) \cup \{d\}) \neq D(C, C \cup \{d\})$.

Theo định nghĩa 2.1, R là tập rút $IDS_1 = (U, C \cup B \cup \{d\})$ gọn của

3.2.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung tập thuộc tính.

Nhu đã đề cập ở trên, thuật toán UARA [70] và thuật toán IDRA [12] là các thuật toán gia tăng rút gọn thuộc tính theo tiếp cận lọc khi bổ sung tập thuộc tính. Hai thuật toán này và các thuật toán lọc trước đây chỉ bao gồm một pha là pha lọc để lấy tập rút gọn. Tập rút gọn thu được trong giai đoạn lọc là tập thuộc tính có điều kiện bảo toàn độ đo ban đầu. Độ chính xác phân lớp được tính toán sau khi thu được tập rút gọn.

Từ các mệnh đề trên, thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ sử dụng khoảng cách khi bổ sung tập thuộc tính được mô tả như sau:

Thuật toán FWIA_AA (*Filter-Wrapper Incremental Algorithm for Attribute Reduction in Incomplete Decision Tables when Add Attributes*).

Đầu vào

- Bảng quyết định không đầy đủ $IDS \sqsubseteq (U, C \cup \{d\})$ với $U \sqsubseteq [u_1, u_2, \dots, u_n]$, tập rút gọn $R \sqsubseteq C$, các ma trận dung sai $M(C) \sqsubseteq [c_{ij}]_{n \times n}$, $M_U([d]) \sqsubseteq [d_{ij}]_{n \times n}$, khoảng cách $D \sqsubseteq C, C \sqsubseteq d \sqsubseteq [d_{ij}]_{n \times n}$;
- Tập thuộc tính bổ sung B với $B \sqsubseteq C \sqsubseteq [d]$;

Đầu ra: Tập rút gọn R_I của $IDS_I \sqsubseteq (U, C \cup B \cup d)$

Bước 1: Khởi tạo và kiểm tra tập thuộc tính bổ sung

1. $T := \emptyset; // Chứa các ứng viên tập rút gọn$

2. Tính ma trận dung sai

$$M(B) = \left[b_{ij} \right]_{n \times n};$$

3. If $b \geq c$

với mọi $1 \leq i \leq n, 1 \leq j \leq n$

then Return R ;

ij

Bước 2: Thực hiện thuật toán tìm tập rút gọn

// Giai đoạn lọc, tìm các ứng viên cho tập rút gọn xuất phát từ tập R.

4. $j := 0;$

5. While

$$D(R, R \cup \{d\}) \neq D(C \cup B, C \cup B \cup \{d\})$$

6. Begin

7. tính
 $j := j + 1;$

$$SIG_R(a) = D(R, R \cup \{d\}) - D(R \cup \{a\}, R)$$

8. For each $a \in B$

$$\cup \{a\} \cup \{d\})$$

$D(R \cup \{a\}, R \cup \{a\} \cup \{d\})$ được tính bởi công thức gia tăng trong mệnh đề 3.1;

$$9. \quad \begin{matrix} & \in B \\ a_m & h \\ & o \\ & n \end{matrix}$$

sao cho

$$SIG_R(a_m) = \max\{SIG_R(a_i)$$

$$\}$$

$$10. \quad R := R \cup \{a_m\};$$

$$11. \quad T_j = R;$$

12. $T = T \cup T_j;$

13. End;

// Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất

14. For $i = 1$ to j

15. Tính độ chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp kiểm tra chéo 10-fold;

16. $R_{best} = T_k;$ // với $T_k (1 \leq k \leq t)$ có độ chính xác phân lớp cao nhất;

17. Return

$R_{best}.$

Đánh giá độ phức tạp của thuật toán FWIA_AA.

Ký hiệu C, U, B

tương ứng là số thuộc tính
điều kiện, số đối tượng và số

thuộc tính điều kiện bổ sung thêm. Ở câu lệnh 2, độ phức tạp tính ma trận dung

sai $M(B)$ là $O(BU^2)$. Trong trường hợp tốt
nhất, thuật toán kết thúc ở câu lệnh
 \parallel

3 (tập rút gọn không thay đổi). Khi đó, độ phức tạp thuật toán FWIA_AA là

$O(B\|U^2)$. Ngược lại xét vòng lặp (a) ta

While từ câu lệnh 4 đến 10, để tính SIG

phải tính

$D(R \cup \{a\}, R \cup \{a\} \cup \{d\})$. Độ phức
tập tăng

$D(R \cup \{a\}, R \cup \{a\} \cup \{d\})$ là

$O(U^2)$. Do đó, độ phức tạp của vòng lặp

While là

và độ phức tạp của giai đoạn
lọc là

$O(B^2U^2)$. Giả sử độ phức

$O(B^2U^2)$

tập

của bộ phân lớp là $O(f(n))$, khi đó độ phức tạp của giai đoạn đóng gói là

$O(|B| * f(n))$. Vì vậy, độ phức tạp của thuật toán FWIA_AA là $O(|B|^2|U|^2)$

$+O(|B|*f(n))$. Nếu thực hiện thuật toán không gia tăng lọc - đóng gói trực tiếp trên bảng quyết định có số thuộc tính $C \cup B$, độ phức tạp là

$$O\left(\left(C + B\right)^2 * U^2\right) + O\left((|C| + |B|)*f(n)\right).$$

Do đó, thuật toán gia tăng FWIA_AA giảm thiểu đáng kể độ phức tạp thời gian thực hiện, đặc biệt trong trường hợp B nhỏ.

3.2.3. Thực nghiệm, đánh giá thuật toán FWIA_AA

3.2.3.1. Mục tiêu thực nghiệm

Mục tiêu thực nghiệm là đánh giá tính hiệu quả của thuật toán theo tiêu chí đánh giá là số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện. Thuật toán FWIA_AA được so sánh với thuật toán UARA[70] và IDRA[12]. Thuật toán UARA là thuật toán tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp bổ sung tập thuộc tính theo tiếp cận lọc sử dụng miền dương. Trong khi đó, IDRA là thuật toán tìm tập rút gọn của bảng quyết định

không đầy đủ trong trường hợp bổ sung tập thuộc tính theo tiếp cận lọc sử dụng quan hệ phân biệt.

3.2.3.2. Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Thực nghiệm được thực hiện trên 06 tập dữ liệu mẫu từ kho dữ liệu UCI [73], được mô tả như trong bảng 3.2, các thuộc tính điều kiện được tách ngẫu nhiên thành hai phần xấp xỉ bằng nhau: các thuộc tính ban đầu được ký hiệu là C_0 và các thuộc tính còn lại. Phần thuộc tính còn lại được chia ngẫu nhiên thành 5 phần xấp xỉ bằng nhau dưới dạng tập thuộc tính gia tăng và được ký hiệu là C_1, C_2, C_3, C_4, C_5 .

Bảng 3.2. Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_AA

STT	Tập dữ liệu	Số đối tượng	Số thuộc tính điều kiện	Thuộc tính ban đầu	Thuộc tính còn lại	Số lớp quyết định
1	Audiology	226	69	34	35	24
2	Soybean-large	307	35	20	15	2
3	Cong.Voting Records	435	16	6	10	2
4	Arrhythmia	452	279	139	140	16
5	Anneal	798	38	18	20	6
6	Internet Advertisements (Advers)	3279	1558	778	780	2

Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold.

Môi trường thực nghiệm: Thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel®, Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

3.2.3.3. Kích bản thực nghiệm

Đầu tiên, các thuật toán FWIA_AA, UARA và IDRA thực hiện trên tập thuộc tính ban đầu C_0 để tìm tập rút gọn. Tiếp theo, thực hiện ba thuật toán khi bổ sung lần lượt từ C_1 đến C_5 .

3.2.3.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 3.3 trình bày kết quả so sánh của FWIA_AA, UARA và IDRA về số thuộc tính tập rút gọn ($|R|$) và độ chính xác phân lớp (Acc).

Bảng 3.3. Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của 3 thuật toán FWIA_AA, UARA và IDRA

STT	Tập dữ liệu	Tập thuộc tính	NA	TA	FWIA_AA		UARA		IDRA	
					$ R $	Acc	$ R $	Acc	$ R $	Acc
1	Audiology	C_0	34	34	4	64.26	8	62.18	7	61.45
		C_1	7	41	5	68.19	9	65.17	9	65.92
		C_2	7	48	5	68.19	10	69.26	9	70.18
		C_3	7	55	6	72.36	12	72.38	11	72.42
		C_4	7	62	7	78.26	14	74.18	12	75.48
		C_5	7	69	7	78.26	15	77.02	14	77.86
2	Soybean – large	C_0	20	20	4	82.34	8	81.16	8	81.16
		C_1	3	23	4	82.34	8	81.16	8	81.16
		C_2	3	26	5	86.92	9	82.08	9	82.24
		C_3	3	29	5	86.92	10	85.14	11	86.18
		C_4	3	32	7	90.27	11	90.26	12	90.96
		C_5	3	35	8	92.85	12	91.18	13	92.06
3	Cong. Voting Records	C_0	6	6	4	81.36	5	81.04	6	82.68
		C_1	2	8	5	86.24	7	85.52	7	85.14
		C_2	2	10	6	89.18	8	89.18	7	88.36
		C_3	2	12	6	89.18	9	89.18	8	88.94
		C_4	2	14	7	91.15	11	90.29	10	89.85
		C_5	2	16	8	94.06	12	93.68	11	92.06
4	Arrhythmia	C_0	139	139	5	62.14	9	62.86	8	61.08
		C_1	28	167	6	69.27	14	68.15	15	69.16
		C_2	28	195	7	70.48	16	69.84	17	70.34
		C_3	28	223	7	70.48	17	69.84	18	71.23
		C_4	28	251	9	71.37	24	70.92	21	72.44
		C_5	28	279	10	76.24	25	74.68	22	75.84
5	Anneal	C_0	18	18	3	68.24	5	68.24	5	68.16
		C_1	4	22	4	72.46	7	71.62	7	71.54
		C_2	4	26	4	72.46	7	71.62	7	71.54
		C_3	4	30	5	79.88	8	76.85	8	77.42
		C_4	4	34	6	86.13	9	85.19	9	90.48
		C_5	4	38	7	91.28	10	90.84	9	90.48
		C_0	778	778	9	71.18	15	70.68	12	68.48

6	Advers.	C_1	156	934	12	76.64	22	72.85	19	71.65
		C_2	156	1090	15	79.14	29	78.94	25	79.04
		C_3	156	1246	19	86.18	35	83.17	31	84.26
		C_4	156	1402	20	89.24	38	86.26	33	88.18
		C_5	156	1558	21	92.85	44	91.46	36	92.04

Trong bảng này, các cột *NA*, *TA* lần lượt là số thuộc tính trong mỗi tập con và tổng các thuộc tính được xem xét tương ứng.

Kết quả trong bảng 3.3 cho thấy, với mỗi lần lặp lại bổ sung tập thuộc tính gia tăng, độ chính xác phân lớp của FWIA_AA cao hơn một chút so với UARA và IDRA trong tất cả các tập dữ liệu. Hơn nữa, kích thước của tập rút gọn do FWIA_AA thu được nhỏ hơn nhiều so với của UARA và IDRA trong tất cả các tập dữ liệu, đặc biệt là trên các tập dữ liệu có số lượng lớn các thuộc tính như Arrhythmia, Advers. Số lượng thuộc tính trong tập rút gọn thu được của FWIA_AA nhỏ hơn nhiều so với của UARA, IDRA.

Với kết quả đó, mô hình phân lớp trên tập rút gọn của FWIA_AA hiệu quả hơn so với UARA, IDRA. Điều này cho thấy rằng FWIA_AA giảm đáng kể kích thước của tập rút gọn trong khi vẫn bảo toàn độ chính xác của phân lớp so với các thuật toán gia tăng tiếp cận lọc.

3.2.3.5. Đánh giá thời gian thực hiện

Kết quả về thời gian thực hiện của ba thuật toán được thể hiện trong bảng 3.4, trong đó các cột *RT*, *Total RT* lần lượt là thời gian thực hiện, tổng thời gian thực hiện.

Từ bảng 3.4 cho thấy, thời gian thực hiện của FWIA_AA cao hơn thời gian thực hiện của UARA và IDRA trên tất cả các tập dữ liệu. Điều này chủ yếu là do tốn thời gian để thực hiện trình phân lớp trong giai đoạn đóng gói của FWIA_AA. Đây là hạn chế chung của các thuật toán sử dụng phương pháp lọc
- đóng gói.

Thời gian thực hiện của thuật toán IDRA nhỏ hơn của thuật toán UARA vì độ phức tạp về thời gian của việc tính toán quan hệ không phân biệt nhỏ hơn so với thời gian tính toán miền dương.

Bảng 3.4. Thời gian thực hiện ba thuật toán FWIA_AA, UARA, IDRA
(tính bằng giây)

STT	Tập dữ liệu	Tập thuộ c tính	NA	TA	FWIA_AA		UARA		IDRA	
					RT	Total RT	RT	Tota l RT	RT	Total RT
1	Audiology	C ₀	34	34	5.36	5.36	4.28	4.28	3.96	3.96
		C ₁	7	41	0.48	5.84	0.39	4.67	0.32	4.28
		C ₂	7	48	0.45	6.29	0.41	5.08	0.39	4.67
		C ₃	7	55	0.52	6.81	0.38	5.46	0.31	4.98
		C ₄	7	62	0.44	7.25	0.39	5.85	0.38	5.36
		C ₅	7	69	0.59	7.84	0.34	6.19	0.43	5.79
2	Soybean-large	C ₀	20	20	2.84	2.84	2.18	2.18	2.03	2.03
		C ₁	3	23	0.14	2.98	0.36	2.54	0.32	2.35
		C ₂	3	26	0.21	3.19	0.22	2.76	0.24	2.59
		C ₃	3	29	0.16	3.35	0.15	2.91	0.18	2.78
		C ₄	3	32	0.33	3.68	0.21	3.12	0.15	2.93
		C ₅	3	35	0.28	3.96	0.16	3.28	0.12	3.05
3	Cong. Voting Records	C ₀	6	6	4.12	4.12	3.08	3.08	2.86	2.86
		C ₁	2	8	0.54	4.66	0.54	3.62	0.58	3.44
		C ₂	2	10	0.32	4.98	0.43	4.05	0.36	3.80
		C ₃	2	12	0.63	5.61	0.54	4.59	0.48	4.28
		C ₄	2	14	0.51	6.12	0.53	5.12	0.42	4.90
		C ₅	2	16	0.72	6.84	0.56	5.68	0.36	5.26
4	Arrhythmia	C ₀	139	139	24.68	24.68	20.78	20.78	18.64	18.64
		C ₁	28	167	3.04	27.72	2.06	22.84	2.33	20.97
		C ₂	28	195	3.24	30.96	2.33	25.17	2.28	22.25
		C ₃	28	223	3.69	34.65	2.89	28.06	2.34	24.59
		C ₄	28	251	2.07	36.72	2.06	30.12	2.15	26.74
		C ₅	28	279	2.12	38.84	1.16	31.28	1.90	28.64
5	Anneal	C ₀	18	18	6.84	6.84	5.19	5.19	4.86	4.86
		C ₁	4	22	0.48	7.32	0.55	5.74	0.51	5.37
		C ₂	4	26	0.43	7.75	0.55	6.29	0.48	5.85
		C ₃	4	30	0.44	8.19	0.53	6.82	0.46	6.31
		C ₄	4	34	0.45	8.64	0.36	7.18	0.32	6.33
		C ₅	4	38	0.42	9.06	0.24	7.42	0.26	6.59
6	Advers	C ₀	778	778	77.24	77.24	68.35	68.35	54.46	54.46
		C ₁	156	934	6.51	83.75	4.54	72.89	4.28	58.74
		C ₂	156	1090	6.09	89.84	5.35	78.24	5.17	63.91
		C ₃	156	1246	6.13	95.97	4.94	83.18	4.34	68.25
		C ₄	156	1402	5.26	101.23	4.58	87.76	4.06	72.31
		C ₅	156	1558	5.55	106.78	4.52	92.28	4.19	76.50

3.3. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi loại bỏ tập thuộc tính.

Trong các bài toán thực tế, không chỉ xem xét tập thuộc tính thay đổi như bổ sung mà còn xảy ra các trường hợp loại bỏ thuộc tính. Mục này, nghiên cứu, đề xuất công thức cập nhật khoảng cách trong trường hợp loại bỏ tập thuộc tính, trên cơ sở đó để xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp loại bỏ tập thuộc tính. Sau đó tiến hành thực nghiệm so sánh với công trình mới nhất để đánh giá hiệu quả thuật toán.

3.3.1. Công thức gia tăng cập nhật khoảng cách khi loại bỏ tập thuộc tính.

Trong [1], tác giả xây dựng công thức tính độ đo khoảng cách khi loại bỏ tập thuộc tính nhưng không sử dụng phương pháp gia tăng để xem xét phần thay đổi. Trong các công trình [CT3, CT5], chúng tôi đã đề xuất công thức cập nhật khoảng cách trong trường hợp loại bỏ tập thuộc tính có sử dụng phương pháp gia tăng và được trình bày cụ thể bởi mệnh đề sau đây:

Mệnh đề 3.3. Cho $IDS = (U, C \cup \{d\})$ bảng quyết định
với
không đầy đủ

$U = \{u_1, u_2, \dots, u_n\}$. Giả sử tập thuộc tính điều kiện B được loại bỏ khỏi C với $B \subset C$

và $A = C - B$ là tập thuộc tính còn lại. $M(A) = [a_{ij}]_{n \times n}$ tương

Đặt $M(B) = [b_{ij}]_{n \times n}$ và ứng là ma trận

dung sai trên B và A . Khi đó ta có:

$$\binom{3}{3}$$

$$). \left(1-d\right)$$

$$^{a.c}$$

$$\sum(^a$$

$$n^2{}^{i=I,j=I}$$

$$S_A\left(u_i\right)\cap S$$

$${\mathbb C}$$

$$h$$

$$\acute{u}r$$

$$n$$

$$g$$

$$m$$

$$i$$

$$n$$

$$h$$

$$\left(u_i\right) \Big)$$

$$\begin{matrix} \mathrm{T} \\ \mathrm{a} \\ \mathrm{c} \\ \mathrm{o} \\ \vdots \end{matrix}$$

$$D\left(\mathcal{A},\mathcal{A}\cup\{d\}\right)=$$

$$\frac{1}{n}.\sum^n\Big(\;S_A\big(u_i\big)-$$

$$= \frac{1}{n} \cdot \sum_i^n \left(S_C(u_i) - S_C(u_i^*) \right)$$

$$) \cap S \quad \quad \quad (u_i) \Big)^+^{-\frac{1}{L}} \cdot \sum^n \Big({}^S_A(u_i) - S_C(u_i$$

$$) - \left(S_A(u_i) - S_C(u_i) \right) \cap S$$

$$n^2 \quad i=1 \qquad \qquad \qquad \{d\} \qquad \qquad n^2 \quad i=1 \qquad \qquad \qquad \{d\}$$

$$= D(C, C \cup \{d\}) + \frac{1}{n} \sum^n \sum^{\text{II}} (a - a.c) - (a - a.c).d$$

$$n^2 \sum_{i=1, j=1}^{\text{II}} ij - ij - ij - ij - ij - ij$$

$$= D(C, C \cup \{d\}) + \sum^a (a.c) \cdot (1-d)$$

$$\frac{1}{n} \sum^n$$

$$n^2 \sum_{i=1, j=1}^{\text{II}} ij - ij - ij - ij$$

Ví dụ 3.2. Xét bảng quyết định $IDS = (U, C \cup \{d\})$ với

được biểu diễn thông tin ở bảng 3.1.
 $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$, $C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$

Tập thuộc tính loại bỏ $B = \{c_5, c_6\}$, $A = C - B = \{c_1, c_2, c_3, c_4\}$.

- Tính các ma trận $M(C)$, $M(A)$, $M(\{d\})$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 \end{vmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 & 1 \end{vmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M(C) = \begin{vmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 1 \end{vmatrix}$$

$$\begin{vmatrix} \square & & & \square & & \end{vmatrix}$$

$$\begin{vmatrix} 0 & 1 & 0 & 0 & 1 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

$$M(A) = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix}$$

$$\square \quad \square$$

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 1 & \square & 0 & \square & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} \square & 0 & 0 & 1 & 0 & 0 & 0 \end{vmatrix}$$

$$M(d) = \begin{vmatrix} \square & \end{vmatrix}$$

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

$$\begin{vmatrix} & & & & & \end{vmatrix}$$

$$\begin{vmatrix} 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

- Tính khoảng cách theo công thức:

$$D(C, C \cup \{d\}) = \frac{1}{n} \sum \sum (c - c \cdot d)$$

được:

$$D(A, A \cup \{d\}) = 4$$

và $D(C, C$

36

$$\begin{matrix} \square \\ \square \end{matrix} \cup \{d\} = 2$$

Ta

36

II

$$n^2 \sum_{i=1}^n \sum_{j=1}^n ij - ij - ij$$

- Tính $\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n (a - a.c) \cdot (1-d)$ ta được: $\frac{1}{n} (a - a.c) \cdot (1-d) = \frac{2}{3}$

$$n^2 \sum_{i=1}^n \sum_{j=1}^n ij - ij - ij$$

$$n^2 \sum_{i=1}^n \sum_{j=1}^n ij - ij - ij$$

36

$$\text{Vậy: } D(A, A \cup \{d\}) = \frac{4}{3} = D(C, C \cup (a - a.c) \cdot (1-d))$$

$$\cup \{d\}) + \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n$$

$$ij - ij - ij$$

3.3.2. Thuật toán *gia tăng lọc - đóng gói* tìm tập rút gọn của bảng quyết định không đầy đủ khi loại bỏ tập thuộc tính.

Từ mệnh đề 3.3, thuật toán lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ sử dụng khoảng cách khi loại bỏ tập thuộc tính B được mô tả chi tiết như hình 3.1 dưới đây. Thuật toán bao gồm hai giai đoạn:

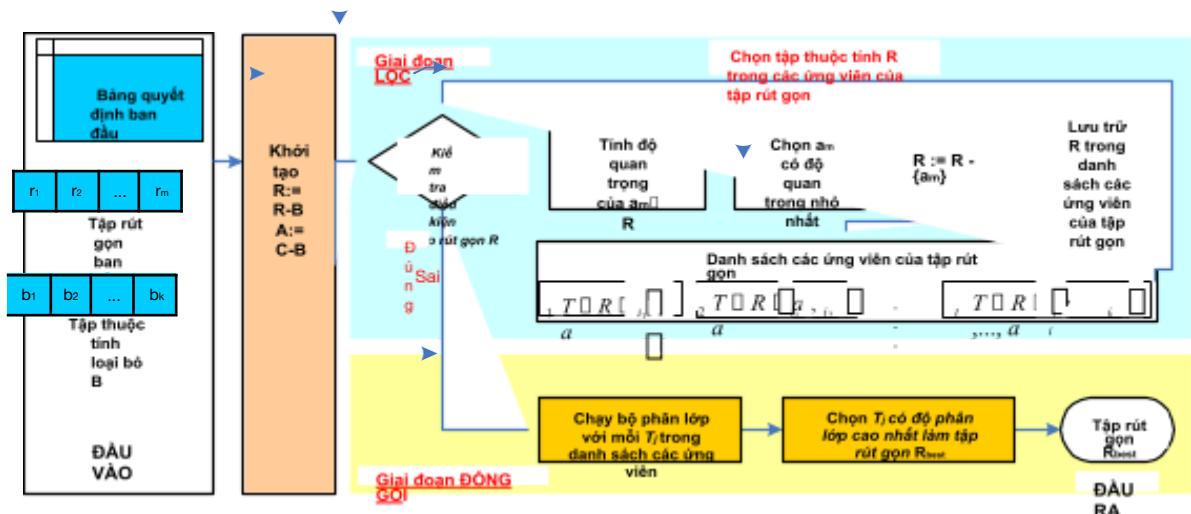
Giai đoạn lọc: Giả sử tập R là tập rút gọn của tập thuộc tính ban đầu và tập B là tập thuộc tính loại bỏ. Đối với $a_i \in R$, tính độ quan trọng của a_i mỗi

theo công thức gia tăng trong định nghĩa 2.2 và chọn a_i với độ quan trọng nhỏ nhất (được ký a_{min}). Sau đó a_{min} được loại bỏ khỏi $R := R - \{a_{min}\}$ và R hiệu là

được lưu trữ trong danh sách các ứng viên cho tập rút gọn. Quá trình này được lặp lại cho đến khi R thỏa mãn điều kiện của tập rút gọn, sau đó thuật toán thực hiện giai đoạn thứ hai là giai đoạn đóng gói.

Giai đoạn đóng gói: Thực hiện trong danh sách bộ phân lớp trên mỗi phần tử T_j

ứng viên và chọn phần tử có độ chính xác phân lớp tối đa làm đầu ra của thuật toán.



Hình 3.1: Sơ đồ khái niệm của thuật toán *gia tăng lọc - đóng gói* tìm tập rút gọn

gọn trong trường hợp loại bỏ tập thuộc tính

Thuật toán gia tăng lọc - đóng gói FWIA_DA cập nhật tập rút gọn trong bảng quyết định không đầy đủ sử dụng khoảng cách khi loại bỏ tập thuộc tính B được mô tả như sau:

Thuật toán FWIA_DA (*Filter-Wrapper Incremental Algorithm for attribute reduction when Delete Attributes*).

Đầu vào:

1) Bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$,

tập rút gọn khoảng cách $R \subseteq C$, các ma trận dung sai $M(C) = [c_{ij}]_{n \times n}$, $M_U(\{d\}) = [d_{ij}]_{n \times n}$,

$D(C, C \cup \{d\})$

2) Tập thuộc tính B loại bỏ khỏi C với $B \subset C$;

Đầu ra: Tập rút gọn R_1 của cua $IDS_1 = (U, (C - B) \cup \{d\})$;

- 1) Trường hợp 1: If $B \subseteq C - R$ then Return(R);
- 2) Trường hợp 2: If $R \subseteq B$ then thực hiện thuật toán không gia tăng lọc - đóng gói tìm tập rút gọn sử dụng khoảng cách IDS_FW_DAR[3].
- 3) Trường hợp 3: If $R \cap B = \emptyset$ then thực hiện các bước của thuật toán tìm tập rút gọn.

Bước 1: Khởi tạo

1. Đặt $T := \emptyset$; $A := C - B$; // Chứa các ứng viên tập rút gọn

2. Tí n M(B) = $[b_{ij}]$ $n \times n$, $M(A) = [a_{ij}]$
 n g
 h sa
 m i
 a
 tr
 ận
 d

3. Đặt $R := R - B$ // Xét các thuộc tính trong tập rút gọn

Bước 2: Thực hiện thuật toán tìm tập rút gọn

// Giai đoạn lọc, tìm các ứng viên cho tập rút gọn

4. $j := 0;$

5. While $D(R, R \cup \{d\}) \neq D(A, A \cup \{d\})$ do

6. Begin

7. $j := j + 1;$

8. For each $a \in R$ tính $SIG_R(a) = D(R - \{a\}, \{R - \{a\}\} \cup \{d\}) - D(R, R \cup \{d\})$

9. $a_m \in R$ sao cho $SIG_R(a_m) = \min \{SIG_R(a)$

)};

⋮

10. $R := R - \{a_m\};$

```

1       $T_j \sqsubseteq R$ 
1      ;
       $T \sqsubseteq T$ 
· End;

// Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất;
· For  $i = 1$  to  $j$ 
·     Tính độ chính xác phân lớp trên  $T_i$  bằng một bộ phân lớp sử
         dụng phương pháp kiểm tra chéo 10-fold;
·      $R_{best} \sqsubseteq T_k$ ; //với  $T_k (1 \leq k \leq t)$  có độ chính xác phân lớp lớn nhất;
· Return  $R_{best}$ .

```

Đánh giá độ phức tạp của thuật toán FWIA_DA.

Ký hiệu C, U, B

tương ứng là số thuộc tính
điều kiện, số đối tượng và số

thuộc tính điều kiện xóa khỏi C. *Trường hợp tốt nhất*, thuật toán rơi vào *trường hợp 1*, nghĩa là tập rút gọn không thay đổi. *Trường hợp xấu nhất*, thuật toán rơi vào *trường hợp 2*, thực hiện lại thuật toán IDS_FW_DAR[3] tìm tập rút gọn

trên bảng quyết định sau khi xóa tập thuộc tính B , giả sử độ phức tạp của bộ phân lớp là

$$O(f(n)), \text{ khi đó độ phức tạp là: } O(C - B^2 * U^2) + O(|C - B| * f(n)).$$

Tiếp theo, ta xét độ phức tạp trong *trường hợp 3*. Xét vòng lặp While từ câu lệnh 4 đến 10, để tính ta phải tính

$$D(R - \{a\}, \{R - \{a\}\} \cup \{d\}). \text{ Độ phức tạp}$$

$SIG_R(a)$

tính $D(R - \{a\}, \{R - \{a\}\} \cup \{d\})$ là $O(U^2)$. Do đó, độ phức tạp của vòng lặp While

và độ phức tạp của giai đoạn lọc là

là $O(|R - B|^2 * U^2)$ $O(|R - B|^2 * U^2)$. Khi đó

độ phức tạp của giai đoạn đóng gói là $O(|R - B|) * O(f(n))$. Vì vậy, độ phức tạp của thuật toán FWIA_DA là:

$$O(|R - B|^2 * U^2) + O(|R - B|) * O(f(n)).$$

Nếu thực hiện

thuật toán không gia tăng lọc - đóng gói trực tiếp trên bảng quyết định có số thuộc tính $C - B$ độ phức tạp là

$$O(|C - B|^2 * U^2) + O(|C - B|) * O(f(n)).$$

Đó, với trường

hợp 3 thì thuật toán FWIA_DA hiệu quả. Nếu $|R|$ càng nhỏ thì thuật toán FWIA_DA càng hiệu quả. Nếu thuật toán rơi vào *trường hợp 2* (tính lại tập rút gọn) thì độ phức tạp thuật toán FWIA_DA tương đương thuật toán IDS_FW_DAR[3].

3.3.3. Thực nghiệm, đánh giá thuật toán FWIA_DA

3.3.3.1. Mục tiêu thực nghiệm

Mục tiêu thực nghiệm là đánh giá tính hiệu quả của thuật toán theo tiêu chí đánh giá là số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện. Thuật toán FWIA_DA được so sánh với thuật toán UARD[70]. Thuật toán UARD là thuật toán tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp loại bỏ tập thuộc tính theo tiếp cận lọc sử dụng miền dương.

3.3.3.2. Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Thực nghiệm được thực hiện trên 06 tập dữ liệu mẫu từ kho dữ liệu UCI [73], được mô tả như trong bảng 3.5.

Trong bảng 3.5, các thuộc tính điều kiện được tách ngẫu nhiên thành hai phần xấp xỉ bằng nhau: các thuộc tính ban đầu được ký hiệu là C_0 và các thuộc tính còn lại. Phần thuộc tính còn lại được chia ngẫu nhiên thành 5 phần xấp xỉ bằng nhau, được ký hiệu là C_1, C_2, C_3, C_4, C_5 .

Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold.

Môi trường thực nghiệm: Thực nghiệm được thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel®, Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

Bảng 3.5. Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_DA

STT	Tập dữ liệu	Số đối tượng	Số thuộc tính điều kiện	Thuộc tính ban đầu	Thuộc tính còn lại	Số lớp quyết định
1	Audiology	226	69	34	35	24
2	Soybean-large	307	35	20	15	2
3	Cong. Voting Records	435	16	6	10	2

4	Arrhythmia	452	279	139	140	16
5	Anneal	798	38	18	20	6
6	Internet Advertisement s (Advers)	3279	1558	778	780	2

3.3.3.3. Kích bản thực nghiệm

Đầu tiên, sử dụng thuật toán IDT_FW_DAR[3] để tìm tập rút gọn trên tập thuộc tính ban đầu. Các tập rút gọn tìm được này là đầu vào cho các thuật toán gia tăng lọc - đóng gói FWIA_DA, UARD. Tiếp theo, thực hiện các thuật toán FWIA_DA, UARD khi loại bỏ lần lượt các tập thuộc tính từ C_1 đến C_5 .

3.3.3.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

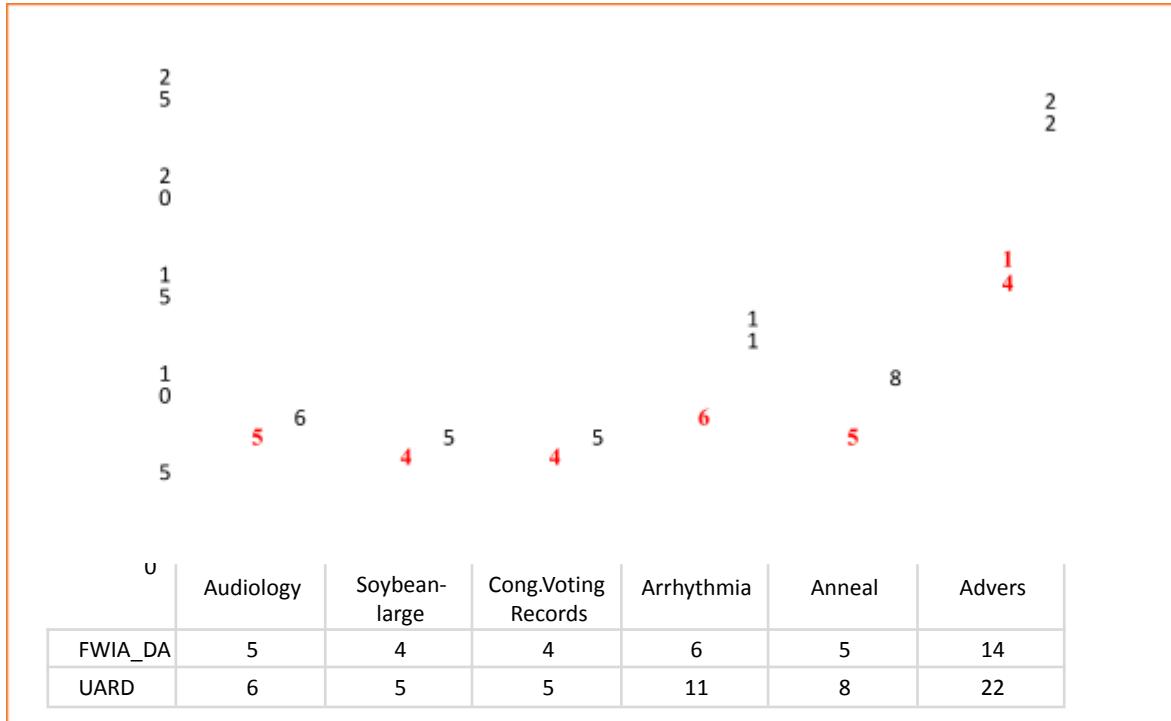
Bảng 3.6 trình bày kết quả so sánh của FWIA_DA và UARD về số thuộc tính tập rút gọn (R) và độ chính xác phân lớp (Acc). Trong bảng này, các cột NA,

TA lần lượt là số thuộc tính trong mỗi tập con và tổng các thuộc tính được xem xét tương ứng. Kết quả trong bảng 3.6 cho thấy, với mỗi lần lặp loại bỏ tập thuộc tính, độ chính xác phân lớp của FWIA_DA cao hơn một chút so với UARD trong tất cả các tập dữ liệu. Hơn nữa, kích thước của tập rút gọn do FWIA_DA thu được nhỏ hơn nhiều so với của UARD trong tất cả các tập dữ liệu, đặc biệt là trên các tập dữ liệu có số lượng lớn các thuộc tính như Arrhythmia, Advers.

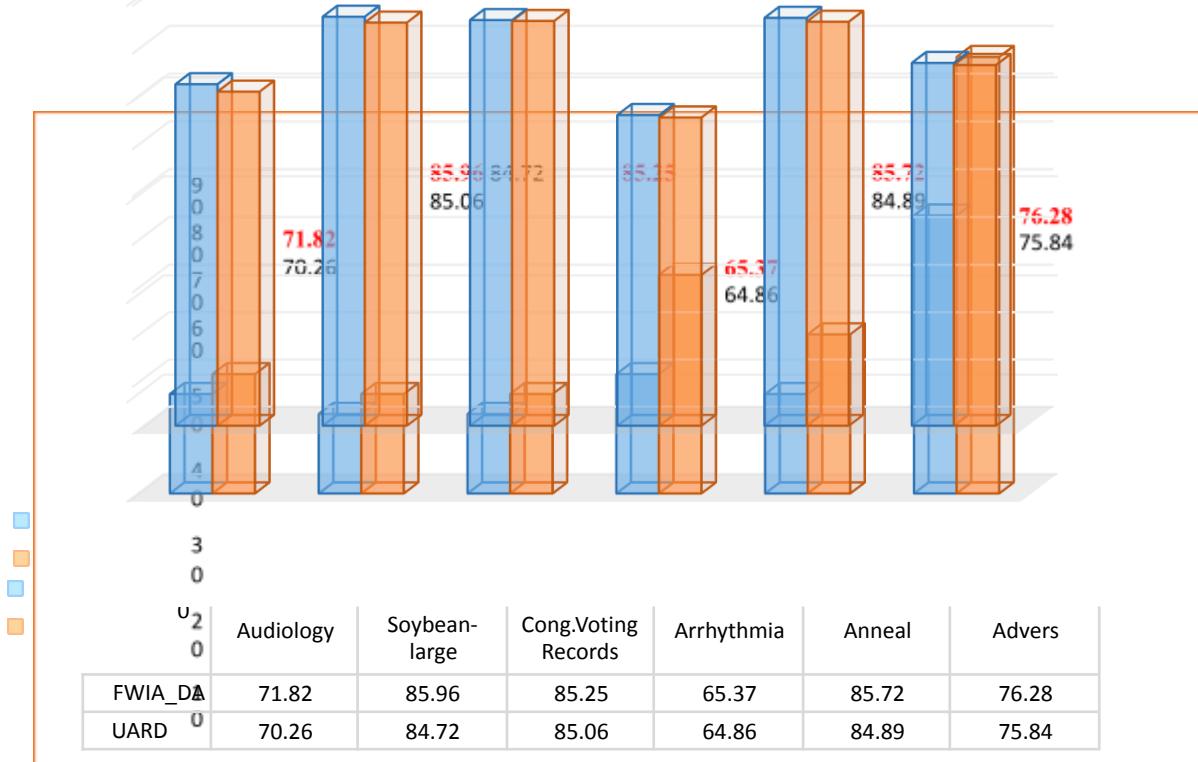
Hình 3.2(a) cho thấy số lượng thuộc tính trong tập rút gọn thu được của FWIA_DA nhỏ hơn nhiều so với của UARD. Với kết quả đó, mô hình phân lớp trên tập rút gọn của FWIA_DA hiệu quả hơn so với UARD. *Hình 3.2(b)* cho thấy độ chính xác phân lớp của các thuật toán là xấp xỉ bằng nhau. Độ chính xác phân lớp của FWIA_DA cao hơn một chút so với UARD trong tất cả các tập dữ liệu. Điều này cho thấy rằng FWIA_DA giảm đáng kể kích thước của tập rút gọn trong khi vẫn bảo toàn độ chính xác của phân lớp so với các thuật toán gia tăng tiếp cận lọc.

Bảng 3.6. Số thuộc tính tập rút gọn và độ chính xác phân lớp của hai thuật toán FWIA_DA và UARD

STT	Tập dữ liệu	Tập thuộc tính	NA	TA	FWIA_DA		UARD	
					R	Acc	R	Acc
1	Audiology	C_1	7	62	7	78.14	7	78.14
		C_2	7	55	6	72.69	7	71.94
		C_3	7	48	6	72.69	7	71.94
		C_4	7	41	5	71.82	6	70.26
		C_5	7	34	5	71.82	6	70.26
2	Soybean –large	C_1	3	32	8	92.12	10	91.06
		C_2	3	29	7	91.86	8	90.28
		C_3	3	26	6	90.18	6	90.18
		C_4	3	23	6	90.18	6	90.18
		C_5	3	20	4	85.96	5	84.72
3	Cong. Voting Records	C_1	2	14	8	94.25	8	94.25
		C_2	2	12	8	94.25	8	94.25
		C_3	2	10	7	92.84	7	92.84
		C_4	2	8	6	92.18	7	92.84
		C_5	2	6	4	85.25	5	85.06
4	Arrhythmia	C_1	28	251	10	76.68	24	72.46
		C_2	28	223	8	75.87	19	75.06
		C_3	28	195	7	70.68	16	69.75
		C_4	28	167	7	70.68	15	69.23
		C_5	28	139	6	65.37	11	64.86
5	Anneal	C_1	4	34	7	91.12	10	90.27
		C_2	4	30	7	91.12	10	90.27
		C_3	4	26	6	90.86	9	89.02
		C_4	4	22	6	90.86	9	89.02
		C_5	4	18	5	85.72	8	84.89
6	Advers.	C_1	156	1402	21	92.94	42	92.28
		C_2	156	1246	19	92.08	40	91.85
		C_3	156	1090	17	85.24	33	84.79
		C_4	156	934	16	84.76	28	83.85
		C_5	156	778	14	76.28	22	75.84



Hình 3.2(a): Số thuộc tính tập rút gọn của hai thuật toán FWIA_DA và UARD



Hình 3.2(b): Độ chính xác phân lớp của hai thuật toán FWIA_DA và UARD

3.3.3.5. Đánh giá thời gian thực hiện

Kết quả về thời gian thực hiện của 02 thuật toán được thể hiện trong bảng 3.7, trong đó các cột *RT*, *Total RT* lần lượt là thời gian thực hiện, tổng thời gian thực hiện. *Bảng 3.7. Thời gian thực hiện hai thuật toán FWIA_DA và UARD (tính bằng giây)*

STT	Tập dữ liệu	Tập thuộc tính	NA	TA	FWIA_AA		UARA	
					RT	Total RT	RT	Total RT
1	Audiology	<i>C₁</i>	7	62	0.36	0.36	0.28	0.28
		<i>C₂</i>	7	55	0.54	0.90	0.42	0.70
		<i>C₃</i>	7	48	0.48	1.38	0.36	1.06
		<i>C₄</i>	7	41	0.51	1.89	0.42	1.48
		<i>C₅</i>	7	34	0.42	2.31	0.34	1.82
2	Soybean-large	<i>C₁</i>	3	32	0.37	0.37	0.28	0.28
		<i>C₂</i>	3	29	0.22	0.59	0.18	0.46
		<i>C₃</i>	3	26	0.32	0.91	0.21	0.67
		<i>C₄</i>	3	23	0.28	1.19	0.19	0.86
		<i>C₅</i>	3	20	0.25	1.44	0.18	1.04
3	Cong. Voting Records	<i>C₁</i>	2	14	0.68	0.68	0.54	0.54
		<i>C₂</i>	2	12	0.72	1.40	0.57	1.11
		<i>C₃</i>	2	10	0.59	1.99	0.42	1.53
		<i>C₄</i>	2	8	0.71	2.70	0.59	2.12
		<i>C₅</i>	2	6	0.84	3.54	0.65	2.77
4	Arrhythmia	<i>C₁</i>	28	251	4.36	4.36	3.82	3.82
		<i>C₂</i>	28	223	3.95	8.31	3.01	6.83
		<i>C₃</i>	28	195	4.27	12.58	3.84	10.67
		<i>C₄</i>	28	167	4.84	17.42	4.15	14.82
		<i>C₅</i>	28	139	3.18	20.60	2.56	17.38
5	Anneal	<i>C₁</i>	4	34	0.75	0.75	0.62	0.62
		<i>C₂</i>	4	30	0.68	1.43	0.51	1.13
		<i>C₃</i>	4	26	0.72	2.15	0.54	1.67
		<i>C₄</i>	4	22	0.58	2.73	0.42	2.09
		<i>C₅</i>	4	18	0.65	3.38	0.48	2.57
6	Advers	<i>C₁</i>	156	1402	8.26	8.26	6.05	6.05
		<i>C₂</i>	156	1246	7.65	15.91	5.84	11.89
		<i>C₃</i>	156	1090	8.12	24.03	7.29	19.18
		<i>C₄</i>	156	934	9.34	33.37	7.65	26.83

		C_5	156	778	8.85	42.22	6.88	33.71
--	--	-------	-----	-----	------	-------	------	--------------

Từ bảng 3.7 cho thấy, thời gian thực hiện của FWIA_DA cao hơn thời gian thực hiện của UARD trên tất cả các tập dữ liệu. Điều này chủ yếu là do tốn thời gian để thực hiện trình phân lớp trong giai đoạn đóng gói của FWIA_DA.

3.4. Phương pháp gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ khi tập thuộc tính thay đổi giá trị

Như đã trình bày ở trên, trong các bài toán thực tế giá trị các đối tượng thường xuyên thay đổi, cập nhật. Do đó, việc xây dựng các công thức gia tăng tìm khoảng cách nhằm giảm thiểu thời gian thực hiện của các thuật toán tìm tập rút gọn là giải pháp hiệu quả. Trong phần 2.3, luận án đã xây dựng công thức gia tăng tính khoảng cách trong trường hợp tập đối tượng thay đổi giá trị, từ đó nâng cao hiệu quả về thời gian thực hiện cho các thuật toán tìm tập rút gọn. Tuy nhiên, với tình huống các giá trị bị thay đổi nằm trên nhiều đối tượng khác nhau (ví dụ tất cả các đối tượng) và tập trung vào một số thuộc tính cụ thể, khi đó, việc xét tất cả các đối tượng bị thay đổi giá trị là giải pháp không hiệu quả vì số đối tượng bị thay đổi rất lớn. Vì vậy, với tình huống này, luận án nghiên cứu phương pháp gia tăng tìm độ đo khoảng cách dựa trên tập thuộc tính bị thay đổi giá trị. Với giải pháp này, chúng ta chỉ cần tính khoảng cách trên các tập thuộc tính bị thay đổi giá trị và không xét số lượng lớn các đối tượng. Trong phần này, luận án đề xuất công thức gia tăng tìm khoảng cách khi tập thuộc tính thay đổi giá trị, trên cơ sở đó xây dựng thuật toán gia tăng hiệu quả tìm tập rút gọn.

3.4.1. Công thức gia tăng tính khoảng cách khi tập thuộc tính thay đổi giá trị

Cho bảng quyết định $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$ và

$M(C) = [c_{ij}]_{nxn}, M(\{d\}) = [d_{ij}]_{nxn}$ tương ứng là ma trận dung sai trên C và d .

Theo [19], khoảng cách giữa hai tập thuộc tính C và $C \cup \{d\}$ được xác định như sau:

$$D(C, C \cup \{d\}) = \frac{1}{2} \|_n c - c.d \|$$

$$\sum_{n=1}^{\infty} \sum_{i=1}^{ij} ij - ij$$

Từ đó, luận án xây dựng công thức gia tăng tính khoảng cách trong trường hợp tập thuộc tính thay đổi giá trị bởi mệnh đề 3.4 dưới đây.

Mệnh đề 3.4. Cho $IDS = (U, C \cup \{d\})$
bảng quyết định
không đầy đủ

với

$U = \{u_1, u_2, \dots, u_n\}$. Giả sử tập s thuộc tính $\Delta C = \{c_k, c_{k+1}, \dots, c_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$

bị thay đổi giá trị. Giả sử

$$M^{old}(\Delta C) = \left[\begin{smallmatrix} c^{old} \\ \vdots \end{smallmatrix} \right]_{n \times n}, \quad M^{new}(\Delta C) = \left[\begin{smallmatrix} c^{new} \\ \vdots \end{smallmatrix} \right]_{n \times n}$$

tương ứng

là ma trận dung sai của tập thuộc tính ΔC trước và sau khi thay đổi giá trị và
 $M(A) = \left[\begin{smallmatrix} a_{ij} \\ \vdots \end{smallmatrix} \right]_{n \times n}$, $M(\{d\}) = \left[\begin{smallmatrix} d_{ij} \\ \vdots \end{smallmatrix} \right]_{n \times n}$

tương ứng là ma trận dung sai trên là ma trận

dung sai của tập thuộc tính còn lại không thay đổi giá trị $A = C - \Delta C$ và $\{d\}$.

Giả sử $D(C, C \cup \{d\})$, $D(C, C \cup \{d\})$ tương ứng là khoảng cách trước khi và sau

khi tập thuộc tính ΔC

thay đổi giá trị. Khi đó, công thức tính gia tăng khoảng

cách như sau:

$$D(C, C \cup \{d\}) = D(C, C \cup \{d\}) + \frac{1}{n(n-a)} \cdot (c^{new} -$$

$$\frac{c^{old}}{n-a}$$

$$\cup \{d\}) + \frac{1}{n(n-a)} \cdot (c^{new} -$$

$$c^{old}) \cdot (1 - d)$$

(3.4)

$$n \sum_{i=1}^n \sum_{j=1}^n$$

Chứng minh

Theo công thức tính khoảng cách, sau khi cập nhật giá trị tập thuộc tính ΔC ta có:

$$\begin{aligned} & D(C, C \cup \{d\}) = D(A, A \cup \Delta C) \\ & = D(A, A \cup \Delta C) + D(\Delta C, A) \\ & = D(A, A \cup \Delta C) + \sum_{i=1}^n \sum_{j=1}^n |ij - ij| \end{aligned}$$

$$\sum_{i=1}^n \sum_{j=1}^n |ij - ij| \quad (*)$$

Mặt khác, trước khi cập nhật giá trị tập thuộc tính ΔC ta có:

$$D(C, C \cup \{d\}) = D(A, A \cup \Delta C) + \sum_{i=1}^n \sum_{j=1}^n |ij - ij| \quad (**)$$

$$A \cup \Delta C, A \cup \Delta C$$

$$\cup \{d\}) + \sum_{i=1}^n \sum_{j=1}^n |ij - ij|$$

$$i=1 \sum_{j=1}^n$$

Từ (*) và (**) ta có:

$$D(C, C \cup \{d\}) - D(A, A \cup \Delta C) =$$

$$\begin{aligned} & \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |ij - ij| \\ & = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |ij - ij| \end{aligned}$$

$$i=1 \sum_{j=1}^n$$

$$i=1 \sum_{j=1}^n$$

$$= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |ij - ij|$$

$$\sum \left(a_{\cdot c^{new}} - a_{\cdot c^{new}.d} \quad a_{\cdot c^{old}} + a_{\cdot c^{old}.d} \right)$$

$$= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \left(a_{ij} \left(c^{new} - c^{old} \right) - a_{ij} \left(c^{new} - c^{old} \right)^2 \right)$$

$$\text{Từ đó ta có: } D(C, C \cup \{d\}) = D(C, C \cup \{d\}) + a_{ij} \cdot (c^{new} - c^{old}) \cdot (1 - d_{ij})$$

$$\frac{1}{n} \sum_n \sum$$

1

Ví dụ 3.2 Xét bảng quyết định không đầy đủ

$$IDS = (U, C \cup \{d\})$$

với

$$U = \{u_1, u_2, u_3, u_4, u_5, u_6\},$$

$$C = \{c_1, c_2, c_3, c_4, c_5, c_6\}$$

được biểu diễn thông tin ở bảng 3.1

Tập thuộc tính thay đổi $\Delta C = \{c_4, c_5, c_6\}$, giá trị

$A = C - \Delta C = \{c_1, c_2, c_3\}$. Các giá trị

của thuộc tính thay đổi thành các giá trị mới như sau:

$$c_4(u_1) = \text{"Cao"}, c_4(u_2) = \text{"Cao"},$$

$$c_5(u_2) = \text{"Không"}, c_5(u_5) = \text{"Không"}, c_5(u_6) = \text{"Không"},$$

$$c_6(u_1) = \text{"Có"}, c_6(u_4) = \text{"Có"}.$$

Khi đó, thông tin được biểu diễn ở bảng 3.8 dưới đây.

Bảng 3.8. Biểu diễn thông tin về các tivi khi thay đổi giá trị

Tivi	Đơn giá	Màu sắc	Kích cỡ	Độ phân giải	Tiết kiệm điện năng	Kết nối Internet	Chuất lượng
	c_1	c_2	c_3	c_4	c_5	c_6	d
u_1	Cao	Đen	Lớn	Cao	Không	Có	Tốt
u_2	Thấp	*	Lớn	Cao	Không	Có	Tốt
u_3	*	*	Nhỏ	Cao	Không	Không	Xấu
u_4	Cao	*	Lớn	Cao	*	Có	Tốt
u_5	*	*	Lớn	Cao	Không	Có	Tuyệt hảo
u_6	Thấp	Đen	Lớn	*	Không	*	Tốt

- Tính ma trận

$$M(C), M(\{d\}), M(A) \text{ và}$$

$$M^{old}(\Delta C) = [c^{old}]$$

$$\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ \square & | & 0 & \square & 1 & 0 & 0 & 0 & 1 \end{array} \quad \begin{array}{cccccc} 1 & 1 & 0 & 1 & 0 & 1 \\ | & & & & & \\ 1 & 1 & 0 & 1 & 0 & 1 \end{array}$$

$$M(\mathbb{C}) = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \square & 0 & 0 & 1 & 0 & 0 \end{vmatrix}^{M(d)} = \begin{vmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix} \\ \begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ | & & & & | & \\ 0 & 1 & 0 & 0 & 1 & 1 \end{vmatrix} \quad \begin{vmatrix} & & & & | & \\ 1 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

$$M^{old}(\Delta C) = [c^{old}]$$

$$= \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 0 & 0 & 1 \\ 0 & 0 \\ 0 \end{vmatrix}$$

\mathcal{M}

$$[ij]_{n \times n} = \begin{vmatrix} 0 & 0 & 1 & 1 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 1 \\ 0 \end{vmatrix}$$

$$= \begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 1 \end{vmatrix}$$

$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 \\ 1 \end{vmatrix}$$

$$= \begin{vmatrix} 0 & 1 & 0 & 1 & 1 & 1 \end{vmatrix}$$

- Tính $D(C, C \cup \{d\}) = \frac{1}{n} \sum_c (c \cdot d)$
 khoảng cách theo công thức:

$$D(C, C \cup \{d\}) = 2$$

$$n^2 \underset{i=1}{\underset{j=1}{\dots}} ij \quad ij \quad ij$$

36 \square \square

Ta có:

- Khi tập thuộc $\Delta C = \{c_4, c_5, c_6\}$ thì tính lại ma trận
tính thay đổi giá trị

trên bảng dữ liệu mới, ta được:

$$M'(C) = [c'] \quad \text{và } M^{new}(\Delta C) = [c^{new}]$$

$$[ij]_{n \times n}$$

$$[ij]_{n \times n}$$

$$[1 \ 0 \ 0 \ 1 \ 1 \ 0]$$

$$M'(C) = [c'] \quad | \quad 0 \ 1 \ 0 \ 0 \ 1 \ 1 |$$

\square

$$= |0 \ 0 \ 1 \ 0 \ 0 \ 0|$$

$$[ij]_{n \times n} | 1 \ 0 \ 0 \ 1 \ 1 \ 0 |$$

$$| 1 \ 1 \ 0 \ 1 \ 1 \ 1 |$$

$$| |$$

$$| 0 \ 1 \ 0 \ 0 \ 1 \ 1 |$$

$$| 1 \ 1 \ 0 \ 1 \ 1 \ 1 |$$

$$| |$$

\square

$$M^{new}(\Delta C) = [c^{new}]$$

$$= |0 \ 0 \ 1 \ 0 \ 0 \ 1|$$

$$\left[\begin{smallmatrix} ij & & & & & \\ & I_{n \times n} & | & & & \\ & & |1 & 1 & 0 & 1 & 1 & 1 \\ & & |1 & & & & & \\ & & & | & & & & \\ & & & & |1 & & & \\ & & & & & | & & \\ & & & & & & | & \\ & & & & & & & | \\ & & & & & & & 1 \end{smallmatrix} \right]$$

- Tính

$$\frac{1}{n} \sum_n a . (c^{new} - c^{old}) . (1 - d) \quad \text{trong đó } n=6, \text{ ta được:}$$

$$\frac{1}{n} \sum_{i=1}^n j=1 a . (c^{new} - c^{old}) = 6$$

$$. (1 - d)$$

$$n \sum_{i=1}^n j=1 6$$

- Tính khoảng cách $D(C, C \cup \{d\})$

sau khi cập nhật tập đối tượng ΔC

$$D(C, C \cup \{d\}) = 8$$

Vậy

6

$$D(C, C \cup \{d\}) = 8 = D(C, C \cup \{d\}) + \frac{1}{n} \sum_n a . (c^{new} - c^{old}) . (1 - d)$$

$$_{i=1}^{ij}\quad _{j=1}^{ij}\qquad \qquad \qquad _{i=1}^{ij}\qquad \qquad \qquad _{j=1}^{ij}$$

3.4.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ khi tập thuộc tính thay đổi giá trị

Trong mục này, luận án đề xuất một thuật toán gia tăng lọc - đóng gói cập nhật tập rút gọn của bảng quyết định không đầy đủ khi tập thuộc tính thay đổi giá trị sử dụng độ đo khoảng cách. Thuật toán đề xuất bao gồm hai giai đoạn: giai đoạn lọc và giai đoạn đóng gói.

Giai đoạn lọc: Giả sử tập R là tập rút gọn của tập thuộc tính ban đầu và là tập thuộc tính thay đổi giá trị $a_i \in A$, tính độ quan trọng tập ΔC đối với mỗi

của a_i theo công thức gia tăng trong định nghĩa 2.2 và chọn a_i với độ quan trọng lớn nhất (được a_{max}). Sau đó a_{max} được thêm vào R : $R := R \cup a_{max}$ và R ký hiệu là

được lưu trữ trong danh sách các ứng viên cho tập rút gọn. Quá trình này được lặp lại cho đến khi R thỏa mãn điều kiện của tập rút gọn, sau đó thuật toán thực hiện giai đoạn thứ hai là giai đoạn đóng gói.

Giai đoạn đóng gói: Thực hiện bộ phân lớp trên mỗi phần tử (T_j) trong danh sách ứng viên và chọn phần tử có độ chính xác phân lớp tối đa làm đầu ra của thuật toán.

Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ sử dụng khoảng cách khi tập thuộc tính thay đổi giá trị được mô

Thuật toán FWIA_U_Attr (*Filter-Wrapper Incremental Algorithm for Attribute Reduction in Incomplete Decision Tables when Update Attributes*).

Đầu vào

- 1) Bảng quyết định không đầy đủ $IDS = \langle U, C \rangle \sqcup d \rangle \rangle$ với $U = \{u_1, u_2, \dots, u_n\}$, tập rút gọn $R \subseteq C$, các ma trận dung sai $M(C) = \langle c_{ij} \rangle_{i=1, j=1}^n$, $M(d) = \langle d_{ij} \rangle_{i=1, j=1}^n$, khoảng cách $D \langle C, C \rangle \sqcup d \rangle \rangle$;

tả như sau:

2) Tập thuộc tính ΔC bị thay đổi giá trị, với $\Delta C \subseteq C$;

Đầu ra: R' của $IDS' = (U, C \cup \{d\})$ sau khi ΔC bị thay đổi giá trị.

Bước 1: Khởi tạo

1. $T := \emptyset$; // Chứa các ứng viên tập rút gọn

2. Đặt $A := C - \Delta C$;

$$3. \text{ Tính } M(A) = [a_{ij}]_{n \times n}, M^{new}(\Delta C) = [c^{new}]_{n \times n}$$

ma

trận

dung

sai

$$M^{old}$$

(ΔC)

$$= \lfloor \lceil C^{old}$$

11

•
•

4. Tính
khoản
g cách
trong
mênh
 $\overset{ij}{\underset{n \times n}{\text{để}}}$
 $D(R, R \cup \{d\}),$
 $D(C, C \cup \{d\})$
3.4;

bởi công thức gia tăng

// Loại bỏ các thuộc tính dư thừa trong R ;

5. For each $a \in R$

6. If $D(R - \{a\}, (R - \{a\}) \cup \{d\}) = D(C, C \cup \{d\})$ then
 $R := R - \{a\}$;

Bước 2: Thực hiện thuật toán tìm tập rút gọn

// Giai đoạn lọc, tìm các ứng viên cho tập rút gọn xuất phát từ tập R .

7. $j:=0$;

8. While $D(R, R \cup \{d\}) \neq D(C, C \cup \{d\})$ Do

9. Begin 10.

$j:=j+1$; $SIG_R(a) = D(R, R \cup \{d\}) - D(R \cup \{a\}, R \cup \{a\} \cup \{d\})$

11. For each $a \in C - R$ tính

với $D(R \cup \{a\}, R \cup \{a\} \cup \{d\})$ được tính bởi công thức gia tăng trong mệnh đề 3.4;

12. $a_m \in C - R$ sao cho $SIG_R(a_m) = \max_{a \in C - R} \{SIG_R(a)\}$;

$a \in C - R$

13.

$$R := R \cup \{a_m\};$$

14.

$$T_j = R;$$

```

1            $T \sqsubseteq T$ 
-
· End;

// Giai đoạn đóng gói, tìm tập rút gọn có độ chính xác phân lớp cao nhất

· For  $i = 1$  to  $j$ 
·     Tính độ chính xác phân lớp trên  $T_i$  bằng một bộ phân lớp sử
        dụng phương pháp kiểm tra chéo 10-fold;
·      $R_{best} \sqsubseteq T_k$ ; // với  $T_k$  ( $1 \leq k \leq t$ ) có độ chính xác phân lớp cao nhất;
·     Return  $R_{best}$ .

```

Đánh giá độ phức tạp của thuật toán FWIA_U_Attr

Ký hiệu $C, U, \Delta C$ tương ứng là số thuộc tính điều kiện, số đối tượng và

số thuộc tính điều kiện bị thay đổi giá trị. Ở câu lệnh 3, độ phức tạp tính ma trận dung sai $M^{new}(\Delta C) ||$ là $O(\Delta C U^2)$. Xét vòng lặp While từ câu lệnh 7 đến 13,

để tính SIG_R (a) ta phải tính $D'(R$

$\cup \{a\}, R \cup \{a\} \cup \{d\}\)$.

Độ phức tạp tính gia tăng

theo mệnh đề 3.5 là

$D'(R \cup \{a\}, R \cup \{a\} \cup \{d\})$ là $O(U^2)$. Do đó, độ phức tạp của

vòng lặp While là $O(C - R^2 U^2)$ và độ phức tạp của giai đoạn lọc là $O(C - R^2 U^2)$

Giả sử độ phức tạp của bộ phân lớp là $O(f(n))$, khi đó độ phức tạp của giai đoạn đóng gói là FWIA_U_Attr là $O(|C - R|) * O(f(n))$. Vì vậy, độ phức tạp của thuật toán

$O(C - R^2 U^2) + O(|C - R|) * O(f(n))$. Nếu thực hiện thuật toán không gia tăng lọc - đóng gói trực tiếp trên bảng quyết định có số thuộc tính C

độ phức tạp là $O(C^2 * U^2) + O(|C|) * O(f(n))$. Do đó, thuật toán gia tăng

FWIA_U_Attr giảm thiểu đáng kể độ phức tạp thời gian thực hiện, đặc biệt trong trường hợp R lớn.

3.4.3. Thực nghiệm, đánh giá thuật toán FWIA_U_Attr

3.3.3.1. Mục tiêu thực nghiệm

Mục tiêu thực nghiệm là đánh giá tính hiệu quả của thuật toán theo tiêu chí: *số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện.*

Thuật toán FWIA_U_Attr so với thuật toán Attribute-R[86] trong trường hợp tập thuộc tính thay đổi giá trị. Attribute-R là thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập thuộc tính thay đổi giá trị theo tiếp cận lọc truyền thống sử dụng độ đo không nhất quán.

3.3.3.2. Số liệu và môi trường thực nghiệm

Số liệu thực nghiệm: Thực nghiệm được thực hiện trên 06 tập dữ liệu mẫu từ kho dữ liệu UCI [73] được mô tả như trong bảng 3.9.

- Các thuộc tính điều kiện được tách ngẫu nhiên thành hai phần (tệp dữ liệu) xấp xỉ bằng nhau: (i) Tập các thuộc tính ban đầu được ký hiệu là **C₀** và (ii) Tập các thuộc tính bị thay đổi giá trị **C_{chan}**.

- Tiếp theo tập **C_{chan}** được chia ngẫu nhiên thành 5 tập dữ liệu xấp xỉ bằng nhau dưới dạng các tập thuộc tính bị thay đổi giá trị và được ký hiệu lần lượt là: **C₁, C₂, C₃, C₄, C₅**.

Bảng 3.9. Các bộ dữ liệu thực nghiệm cho thuật toán FWIA_U_Attr

STT	Tập dữ liệu	Số đổi tượn g	Số thuộc tính điều kiện	Số thuộc tính ban đầu	Số thuộc tính bị thay đổi giá trị	Số lớp quyết định
1	Audiology	226	69	34	35	24
2	Soybean-large	307	35	20	15	2
3	Cong.Voting Records	435	16	6	10	2
4	Arrhythmia	452	279	139	140	16
5	Anneal	798	38	18	20	6
6	Internet Advertisement s (Advers)	3279	1558	778	780	2

Với tập thuộc tính bị thay đổi giá trị , luận án thực hiện cập nhật ngẫu nhiên giá trị thuộc tính của các đối tượng, bảo đảm nguyên tắc các giá trị bị thay đổi thuộc miền giá trị của thuộc tính ban đầu.

Bộ phân lớp C4.5 được sử dụng để tính toán độ chính xác phân lớp của các thuật toán bằng cách sử dụng phương pháp kiểm tra chéo 10-fold.

Môi trường thực nghiệm: Thực nghiệm được thực hiện trên máy tính cá nhân PC: Bộ xử lý Intel®, Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab.

3.3.3.3. Kích bản thực nghiệm

Trước hết, thực hiện cài đặt và chạy 02 thuật toán FWIA_U_Attr và Attribute-R[86] khi lần lượt các tập thuộc tính thay đổi giá trị C_1, C_2, C_3, C_4, C_5 . Sau đó, các giá trị số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện được ghi lại.

3.3.3.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 3.10 trình bày kết quả so sánh của hai thuật toán FWIA_U_Attr và Attribute-R về số thuộc tính tập rút gọn (R) và độ chính xác phân lớp (Acc).

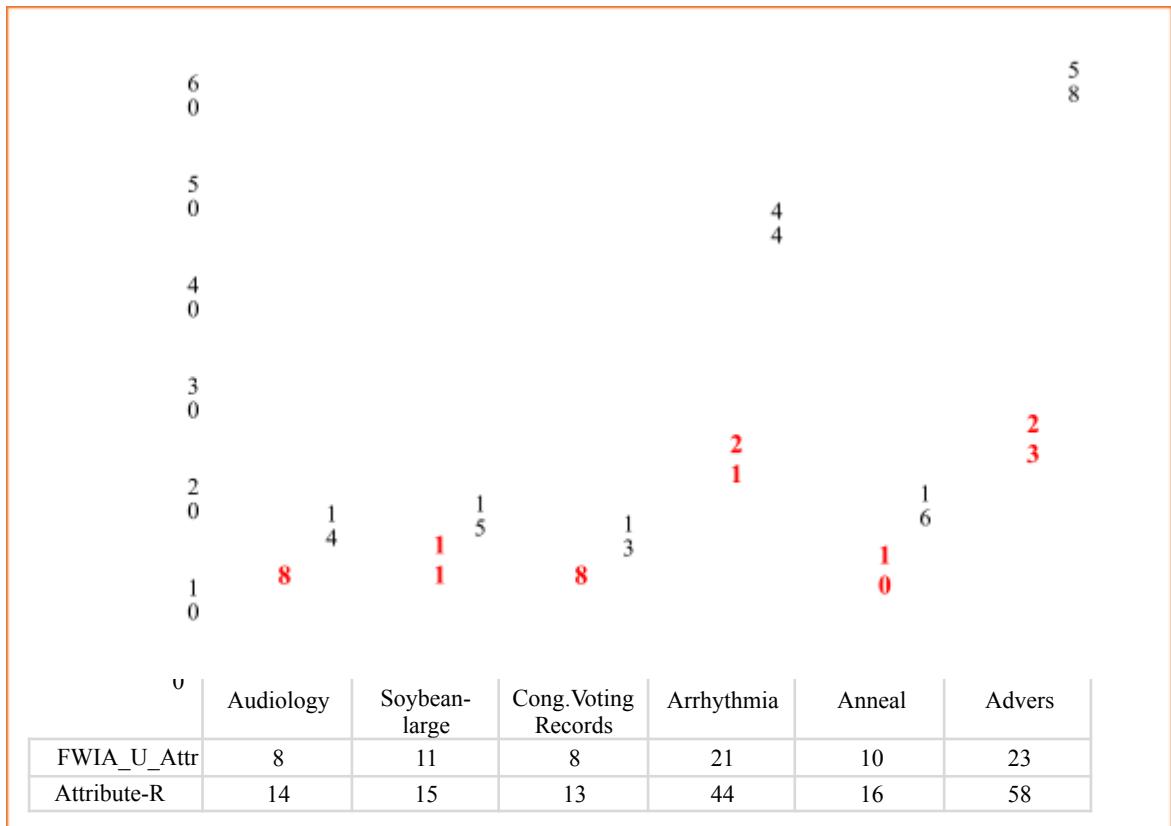
Kết quả trong bảng 3.10 và nhìn trực quan vào hình 3.3(b) cho thấy, với mỗi bước lặp bổ sung tập thuộc tính thay đổi giá trị, độ chính xác phân lớp của FWIA_U_Attr cao hơn một chút so với Attribute-R trên tất cả các tập dữ liệu.

Hơn nữa, kết quả trong bảng 3.10 và nhìn trực quan vào hình 3.3(a) cho thấy số thuộc tính của tập rút gọn thu được bởi thuật toán FWIA_U_Attr nhỏ hơn nhiều so với thuật toán Attribute-R trên tất cả các tập dữ liệu, đặc biệt là trên các tập dữ liệu có số lượng lớn các thuộc tính như Arrhythmia, Advers.

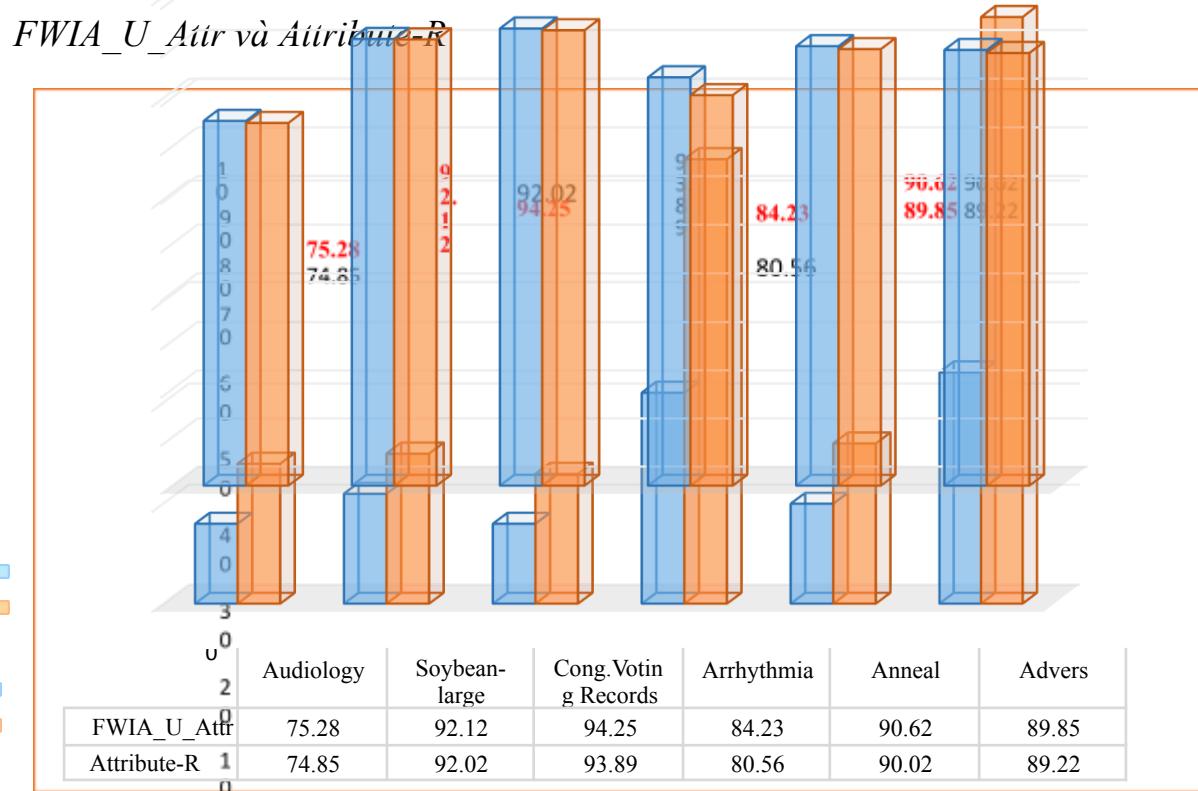
Do đó, mô hình phân lớp dựa trên tập rút gọn của thuật toán FWIA_U_Attr hiệu quả hơn so với thuật toán Attribute-R.

Bảng 3.10. Số thuộc tính tập rút gọn và độ chính xác phân lớp của hai thuật toán FWIA_U_Attr và Attribute-R

STT	Tập dữ liệu	Tập thuộc tính thay đổi giá trị	Số thuộc tính thay đổi giá trị	FWIA_U_Attr		Attribute-R	
				R	Acc	R	Acc
1	Audiology	C_1	7	6	76.84	13	75.26
		C_2	7	8	78.29	15	78.04
		C_3	7	9	77.25	18	76.15
		C_4	7	7	74.62	12	74.29
		C_5	7	8	75.28	14	74.85
2	Soybean –large	C_1	3	9	92.46	16	92.05
		C_2	3	10	94.75	18	93.46
		C_3	3	8	85.18	14	84.95
		C_4	3	12	92.78	21	92.25
		C_5	3	11	92.12	15	92.02
3	Cong. Voting Records	C_1	2	8	92.45	8	92.45
		C_2	2	6	89.17	9	88.95
		C_3	2	6	89.17	9	88.95
		C_4	2	7	93.46	12	93.15
		C_5	2	8	94.25	13	93.89
4	Arrhythmia	C_1	28	16	82.68	42	82.12
		C_2	28	22	86.42	58	85.68
		C_3	28	19	81.25	34	80.76
		C_4	28	25	86.78	47	86.42
		C_5	28	21	84.23	44	80.56
5	Anneal	C_1	4	8	86.49	15	86.14
		C_2	4	12	88.94	19	87.23
		C_3	4	11	88.25	16	86.94
		C_4	4	14	91.15	22	90.28
		C_5	4	10	90.62	16	90.02
6	Advers.	C_1	156	21	88.26	46	87.45
		C_2	156	18	86.15	38	85.12
		C_3	156	17	87.93	42	85.24
		C_4	156	25	90.18	54	89.45
		C_5	156	23	89.85	58	89.22



Hình 3.3(a): Số lượng thuộc tính tập rút gọn của hai thuật toán



Hình 3.3(b): Độ chính xác phân lớp của hai thuật toán FWIA_U_Attr và Attribute-R

3.3.3.5. Đánh giá thời gian thực hiện

Kết quả về thời gian thực hiện của hai thuật toán FWIA_U_Attr và Attribute-R được thể hiện trong bảng 3.11, trong đó các cột *RT*, *Total RT* lần lượt là thời gian thực hiện, tổng thời gian thực hiện.

Bảng 3.11. Thời gian thực hiện hai thuật toán FWIA_U_Attr và Attribute-R (tính bằng giây)

STT	Tập dữ liệu	Tập thuộc tính thay đổi giá trị	Số thuộc tính thay đổi giá trị	FWIA_U_Attr		Attribute-R	
				RT	Total RT	RT	Total RT
1	Audiology	C_1	7	0.64	0.64	0.52	0.52
		C_2	7	0.62	1.26	0.46	0.98
		C_3	7	0.58	1.84	0.42	1.40
		C_4	7	0.69	2.53	0.52	1.92
		C_5	7	0.54	3.07	0.41	2.33
2	Soybean-large	C_1	3	0.45	0.45	0.36	0.36
		C_2	3	0.42	0.87	0.32	0.68
		C_3	3	0.38	1.25	0.29	0.97
		C_4	3	0.46	1.71	0.34	1.31
		C_5	3	0.35	2.06	0.31	1.62
3	Cong. Voting Records	C_1	2	0.52	0.52	0.41	0.41
		C_2	2	0.54	1.06	0.44	0.85
		C_3	2	0.48	1.54	0.38	1.23
		C_4	2	0.58	2.12	0.47	1.70
		C_5	2	0.51	2.63	0.40	2.10
4	Arrhythmia	C_1	28	4.52	4.52	3.98	3.98
		C_2	28	4.96	9.48	3.72	7.70
		C_3	28	5.15	14.63	4.28	11.98
		C_4	28	4.38	19.01	3.82	15.80
		C_5	28	4.92	23.93	3.98	19.78
5	Anneal	C_1	4	2.34	2.34	1.98	1.98
		C_2	4	2.06	4.40	1.75	3.73
		C_3	4	1.95	6.35	1.58	5.31
		C_4	4	2.19	8.54	1.74	7.05
		C_5	4	1.88	10.42	1.52	8.57
		C_1	156	9.45	9.45	7.92	7.92
		C_2	156	8.96	18.41	7.85	15.77

6	Advers	C_3	156	8.15	26.56	6.96	22.73
		C_4	156	9.32	35.88	7.83	30.56
		C_5	156	8.04	43.92	6.84	37.40

Kết quả thực hiện ở bảng 3.11 cho thấy, thời gian thực hiện của thuật toán FWIA_U_Attr cao hơn thời gian thực hiện của thuật toán Attribute-R trên tất cả các tập dữ liệu. Nguyên nhân là, mặc dù độ đo khoảng cách tính toán đơn giản hơn độ đo không nhất quán, tuy nhiên thuật toán FWIA_U_Attr mất thêm chi phí thời gian đáng kể để thực hiện bộ phân lớp trong giai đoạn đóng gói. Đây là hạn chế chung của các thuật toán sử dụng phương pháp lọc - đóng gói.

3.4.4. Thực nghiệm, đánh giá thuật toán FWIA_U_Attr so với việc thực hiện gián tiếp hai thuật toán FWIA_DA và FWIA_AA

3.3.4.1. Mục tiêu thực nghiệm

Để tìm tập rút gọn trong trường hợp tập thuộc tính C_i thay đổi giá trị, chúng ta có thể thực hiện phối hợp 2 thuật toán: thuật toán FWIA_DA khi loại bỏ tập thuộc tính C_i cũ và thuật toán FWIA_AA tìm tập rút gọn khi bổ sung tập thuộc tính C_i mới. Kết quả thử nghiệm để đánh giá tính hiệu quả của thuật toán FWIA_U_Attr so với hướng tiếp cận trước đây là thực hiện đồng thời hai thuật toán: FWIA_DA và FWIA_AA. Việc đánh giá được thực hiện trên thời gian thực hiện và độ chính xác mô hình phân lớp sau rút gọn thuộc tính.

3.3.4.2. Số liệu và môi trường thực nghiệm

Số liệu và môi trường thực nghiệm giống như mô tả trong mục 3.3.3.2.

3.3.4.3. Kịch bản thực nghiệm

Trước hết, thực hiện thuật toán FWIA_U_Attr khi lần lượt các tập thuộc tính C_1, C_2, C_3, C_4, C_5 thay đổi giá trị. Với mỗi tập thuộc tính C_i ($i=1..5$) thay đổi giá trị, thực hiện lần lượt hai thuật toán:

- 1) Thuật toán FWIA_DA khi loại bỏ tập thuộc tính cũ.
- 2) Thuật toán FWIA_AA khi bổ sung tập thuộc tính mới (C_i).

So sánh hai kết quả hai cách tiếp cận trên thời gian thực hiện và độ chính xác phân lớp, số lượng tập rút gọn.

3.3.4.4. Đánh giá về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp

Bảng 3.12 trình bày kết quả về số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Attr so với cách tiếp cận gián tiếp khi lần lượt thực hiện hai thuật toán FWIA_DA và FWIA_AA.

Bảng 3.12. Số lượng tập rút gọn và độ chính xác phân lớp của thuật toán FWIA_U_Attr và 2 thuật toán FWIA_DA và FWIA_AA.

STT	Tập dữ liệu	Tập thuộc tính thay đổi giá trị	Số thuộc tính thay đổi giá trị	FWIA_U_Attr		FWIA_DA và FWIA_AA	
				R	Acc	R	Acc
1	Audiology	C_1	7	6	76.84	6	75.92
		C_2	7	8	78.29	8	77.84
		C_3	7	9	77.25	8	77.06
		C_4	7	7	74.62	7	74.18
		C_5	7	8	75.28	8	74.92
2	Soybean –large	C_1	3	9	92.46	9	92.15
		C_2	3	10	94.75	9	94.16
		C_3	3	8	85.18	8	84.87
		C_4	3	12	92.78	13	92.26
		C_5	3	11	92.12	11	91.85
3	Cong. Voting Records	C_1	2	8	92.45	8	92.68
		C_2	2	6	89.17	6	89.75
		C_3	2	6	89.17	6	89.62
		C_4	2	7	93.46	7	93.67
		C_5	2	8	94.25	8	94.86
4	Arrhythmia	C_1	28	16	82.68	17	82.05
		C_2	28	22	86.42	23	85.98
		C_3	28	19	81.25	19	80.98
		C_4	28	25	86.78	26	86.15
		C_5	28	21	84.23	22	83.98
5	Anneal	C_1	4	8	86.49	8	86.02
		C_2	4	12	88.94	13	88.26
		C_3	4	11	88.25	11	87.85
		C_4	4	14	91.15	13	90.84
		C_5	4	10	90.62	10	90.18
		C_1	156	21	88.26	20	87.96

6	Advers.	C_2	156	18	86.15	19	85.85
		C_3	156	17	87.93	18	87.36
		C_4	156	25	90.18	24	89.85
		C_5	156	23	89.85	24	89.16

Kết quả trong bảng 3.12 cho thấy, số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của hai hướng tiếp cận tính tập rút gọn nêu trên là xấp xỉ bằng nhau. Độ chính xác phân lớp với hướng tiếp cận trực tiếp cải thiện hơn một chút trên tất cả các tập dữ liệu.

3.3.4.5. Đánh giá thời gian thực hiện

Bảng 3.13. Thời gian thực hiện của thuật toán FWIA_U_Attr và 2 thuật toán FWIA_DA và FWIA_AA (tính bằng giây)

STT	Tập dữ liệu	Tập thuộc tính thay đổi giá trị	Số thuộc tính thay đổi giá trị	FWIA_U_Attr		FWIA_DA và FWIA_AA	
				RT	Total RT	RT	Total RT
1	Audiology	C_1	7	0.64	0.64	1.15	1.15
		C_2	7	0.62	1.26	1.06	2.21
		C_3	7	0.58	1.84	0.84	3.05
		C_4	7	0.69	2.53	1.18	4.23
		C_5	7	0.54	3.07	0.96	5.19
2	Soybean- large	C_1	3	0.45	0.45	0.82	0.82
		C_2	3	0.42	0.87	0.78	1.60
		C_3	3	0.38	1.25	0.54	2.14
		C_4	3	0.46	1.71	0.98	3.12
		C_5	3	0.35	2.06	0.69	3.81
3	Cong. Voting Records	C_1	2	0.52	0.52	1.12	1.12
		C_2	2	0.54	1.06	1.95	3.07
		C_3	2	0.48	1.54	0.86	3.93
		C_4	2	0.58	2.12	1.08	5.01
		C_5	2	0.51	2.63	0.94	5.95
4	Arrhythmia	C_1	28	4.52	4.52	7.85	7.85
		C_2	28	4.96	9.48	8.48	16.33
		C_3	28	5.15	14.63	9.63	25.96
		C_4	28	4.38	19.01	7.67	33.63
		C_5	28	4.92	23.93	8.16	41.79
5	Anneal	C_1	4	2.34	2.34	4.28	4.28
		C_2	4	2.06	4.40	3.85	8.13
		C_3	4	1.95	6.35	3.78	11.91
		C_4	4	2.19	8.54	4.25	16.16
		C_5	4	1.88	10.42	3.06	19.22

6	Advers	C_1	156	9.45	9.45	15.76	15.76
		C_2	156	8.96	18.41	14.84	30.60
		C_3	156	8.15	26.56	14.96	45.56
		C_4	156	9.32	35.88	16.25	61.81
		C_5	156	8.04	43.92	15.08	76.89

Thời gian thực hiện của hai hướng tiếp cận tính toán được trình bày như trong bảng 3.13. Trên tất cả các tập dữ liệu, thời gian thực hiện thuật toán FWIA_U_Attr tính trực tiếp tập rút gọn nhỏ hơn nhiều so với hướng tiếp cận tính toán gián tiếp sử dụng thuật toán loại bỏ tập thuộc tính FWIA_DA và thuật toán bỏ sung tập thuộc tính FWIA_AA. Điều đó cho thấy tính hiệu quả của thuật toán FWIA_U_Attr so với cách tiếp cận cũ.

3.5. Kết luận chương 3

Như vậy chương 3 đã nghiên cứu về các trường hợp tập thuộc tính thay đổi trong các trường hợp bỏ sung, loại bỏ tập thuộc tính và tập thuộc tính thay đổi giá trị. Cụ thể như sau:

- 1) Xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trên bảng quyết định không đầy đủ trong trường hợp bỏ sung, loại bỏ tập thuộc tính.
- 2) Đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trên bảng quyết định không đầy đủ trong trường hợp tập thuộc tính thay đổi giá trị.

Kết quả thực nghiệm cho thấy, các thuật toán theo tiếp cận lọc - đóng gói giảm thiểu số lượng thuộc tính tập rút gọn và cải thiện độ chính xác của mô hình phân lớp so với các thuật toán gia tăng khác theo tiếp cận lọc đã công bố. Tuy nhiên về thời gian thực hiện cao hơn vì phải cần nhiều thời gian hơn để thực hiện phân lớp trong giai đoạn đóng gói.

Trong trường hợp tập thuộc tính thay đổi giá trị, dựa trên kết quả thực nghiệm thu được, thuật toán FWIA_U_Attr hiệu quả hơn so với cách tiếp cận gián tiếp sử dụng thuật toán loại bỏ tập thuộc tính FWIA_DA và thuật toán bỏ sung tập thuộc tính FWIA_AA.

Kết quả nghiên cứu của chương này được công bố trong công trình [CT3, CT5, CT7], phần: “Danh mục công trình khoa học của luận án”.

KẾT LUẬN

Những kết quả chính của luận án

Trong xu thế phát triển của dữ liệu lớn (Big data), các bảng quyết định thường không đầy đủ, không chắc chắn, ngày càng có kích thước lớn và luôn thay đổi, cập nhật. Việc xây dựng các thuật toán gia tăng hiệu quả theo hướng tiếp cận lọc

- đóng gói nhằm giảm thiểu số thuộc tính tập rút gọn, từ đó nâng cao hiệu quả các mô hình phân lớp, học máy là vấn đề nghiên cứu có ý nghĩa khoa học và thực tiễn. Cho đến nay, đã có một số kết quả nghiên cứu liên quan đến các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong các trường hợp bổ sung, loại bỏ tập đối tượng, tập thuộc tính. Tuy nhiên, các nghiên cứu này chưa hoàn chỉnh về phần thực nghiệm cũng như về công thức tính toán công thức gia tăng. Do đó, mục tiêu của luận án là hoàn thiện các nghiên cứu trên, đồng thời giải quyết những tình huống còn lại trong bảng quyết định không đầy đủ là trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị. Từ những kết quả thu được, luận án tiến hành so sánh tiếp cận rút gọn trực tiếp với rút gọn thuộc tính gián tiếp, thực hiện đồng thời loại bỏ và bổ sung tập đối tượng, tập thuộc tính. Kết quả đó cho thấy tính hiệu quả của thuật toán đề xuất.

Luận án có ba đóng góp chính như sau:

1) Xây dựng công thức gia tăng cập nhật khoảng cách trong trường hợp bổ sung, loại bỏ tập thuộc tính, trên cơ sở đó xây dựng thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trên bảng quyết định không đầy đủ trong trường hợp bổ sung tập thuộc tính: thuật toán FWIA_AA và trong trường hợp loại bỏ tập thuộc tính: thuật toán FWIA_DA.

2) Đề xuất công thức tính khoảng cách khi tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập đối tượng thay đổi giá trị:

thuật toán FWIA_U_Obj.

3) Đề xuất công thức tính khoảng cách khi tập thuộc tính thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp tập thuộc tính thay đổi giá trị: thuật toán FWIA_U_Attr.

Kết quả thực nghiệm trên các tập dữ liệu mẫu từ kho dữ liệu UCI [73] cho thấy, các thuật toán lọc - đóng gói đề xuất giảm thiểu số lượng thuộc tính tập rút gọn, từ đó nâng cao hiệu quả của mô hình phân lớp sau khi rút gọn thuộc tính. Tuy nhiên, thời gian thực hiện của các thuật toán lọc - đóng gói đề xuất cao hơn các thuật toán lọc đã công bố do phải tính toán các bộ phân lớp.

Hướng phát triển của luận án

1) Cải tiến các thuật toán gia tăng lọc- đóng gói đã công bố nhằm giảm thiểu thời gian thực hiện bằng giải pháp không thực hiện lặp lại các bộ phân lớp.

2) Tiếp tục nghiên cứu, đề xuất các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn theo các mô hình tập thô mở rộng khác trong trường hợp bổ sung, loại bỏ tập đối tượng, tập thuộc tính và tập đối tượng, tập thuộc tính thay đổi giá trị nhằm phù hợp với các lớp bài toán khác nhau trong thực tế.

3) Tiếp tục nghiên cứu thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong trường hợp tập đối tượng, tập thuộc tính khi cùng thay đổi giá trị.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA LUẬN ÁN

[CT1]. Nguyen Anh Tuan, Nguyen Long Giang (2019), “About a Distance Measure and Application for Finding Reduct in Incomplete Decision Tables”, *International Journal of Engineering and Advanced Technology* (IJEAT), ISSN: 2249-8958, Volume 9, Issue 1, pp. 6294-6298.

[CT2]. Nguyễn Anh Tuấn (2020), “Nghiên cứu cải tiến một số độ đo trong lý thuyết tập thô cho bảng quyết định không đầy đủ”, *Chuyên san khoa học tự nhiên - kỹ thuật - công nghệ*, ISSN: 1859-2171, Tập 225(06), trang 200 - 204. [CT3]. Nguyễn Anh Tuấn, Nguyễn Văn Thiện, Nguyễn Long Giang (2020),

“Về các thuật toán gia tăng filter-wrapper tìm tập rút gọn của bảng quyết định không đầy đủ khi bổ sung, loại bỏ tập thuộc tính”, *Kỷ yếu Hội thảo Quốc gia lần thứ XXIII - Một số vấn đề chọn lọc của CNTT và TT*, trang 477-482.

[CT4]. Nguyễn Anh Tuấn, Nguyễn Văn Thiện, Nguyễn Long Giang (2020), “Phương pháp filter-wrapper rút gọn thuộc tính trong bảng quyết định không đầy đủ khi bổ sung, loại bỏ tập đối tượng”, *Kỷ yếu Hội thảo Quốc gia lần thứ XXIII - Một số vấn đề chọn lọc của CNTT và TT*, trang 394-399.

[CT5]. Giang Nguyen, Le Hoang Son, Nguyen Anh Tuan, Tran Thi Ngan, Nguyen Nhu Son, Nguyen Truong Thang (2021), “Filter-Wrapper Incremental Algorithms for Finding Reduct in Incomplete Decision Systems when Adding and Deleting an Attribute Set”, *International Journal of Data Warehousing and Mining (SCIE)*, Volume 17, Issue 2, Article 3, pp. 39-62.

[CT6]. Nguyen Truong Thang, Nguyen Long Giang, Hoang Viet Long, Nguyen Anh Tuan, Tran Manh Tuan, Ngo Duy Tan (2021), “Efficient Algorithms for Dynamic Incomplete Decision Systems”, *International Journal of Data Warehousing and Mining (SCIE)*, Volume 17, Issue 3, Article 2, pp. 47-67.

[CT7]. Nguyễn Anh Tuấn, Nguyễn Long Giang, Vũ Đức Thi (2021), “Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn trong bảng quyết định

không đầy đủ khi tập đổi tượng và tập thuộc tính thay đổi giá trị”, *Chuyên san khoa học tự nhiên - kỹ thuật - công nghệ*, ISSN: 1859-2171, Tập 226(11), trang 234 - 242.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

1. Nguyễn Bá Quang, “Phát triển một số phương pháp rút gọn thuộc tính trong bảng quyết định không đầy đủ theo tiếp cận filter-wrapper”, *Luận án tiến sĩ toán học*, Viện Khoa học và Công nghệ Quân sự, Hà Nội, 2021.
2. Nguyễn Bá Quang, Nguyễn Long Giang, “Về một thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp bổ sung tập thuộc tính”, *Tạp chí Nghiên cứu KH&CN quân sự*, số 63, 10-2019, tr. 171-183.
3. Nguyễn Bá Quang, Nguyễn Long Giang, Trần Thanh Đại, Nguyễn Ngọc Cường (2019), “Phương pháp filter-wrapper rút gọn thuộc tính trong bảng quyết định không đầy đủ sử dụng khoảng cách”, *Kỷ yếu Hội thảo Quốc gia lần thứ XXII - Một số vấn đề chọn lọc của CNTT và TT*, Thái Bình, 246-252.
4. Nguyễn Long Giang (2012), “Nghiên cứu một số phương pháp khai phá dữ liệu theo tiếp cận lý thuyết tập thô”, *Luận án Tiến sĩ Toán học*, Viện Công nghệ thông tin.
5. Nguyễn Long Giang, Nguyễn Thanh Tùng (2012), “Một phương pháp mới rút gọn thuộc tính trong bảng quyết định sử dụng metric”, *Kỷ yếu Hội thảo Một số vấn đề chọn lọc về CNTT và TT*, Cần Thơ, 10/2011, Tr. 249-266.
6. Nguyễn Long Giang, Vũ Đức Thi (2011), “Thuật toán tìm tất cả các rút gọn trong bảng quyết định”, *Tạp chí Tin học và Điều khiển học*, T.27, S.3, tr. 199-205.
7. Phạm Minh Ngọc Hà, Nguyễn Long Giang, Nguyễn Văn Thiện, Nguyễn Bá Quang (2019), “Về một thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ”, *Chuyên san các công trình nghiên cứu phát triển Công nghệ Thông tin và Truyền thông*, T2019, S.1, tr. 11-18.

Tài liệu tiếng Anh

8. Abbas A., Noor R., Irfan M., & Kostaq H. (2019) “Soft ordered approximations and incomplete information system”, *Journal of Intelligent & Fuzzy Systems*, 36(6), pp. 5653-5667.
9. Atay C., Garani G. (2019), “Maintaining Dimension's History in Data Warehouses Effectively”, *International Journal of Data Warehousing and Mining (IJDWM)*, 15(3), pp. 46-62.

10. Cai M., Lang, G., Hamido F., Li Z., Yang T. (2019), “Incremental approaches to updating reducts under dynamic covering granularity”, *Knowledge-Based Systems* 172, pp. 130-140.

11. Cai M., Li Q., Ma J. (2017), “Knowledge reduction of dynamic covering decision information systems caused by variations of attribute values”, *International Journal of Machine Learning and Cybernetics* 8(4), pp. 1131-1144.
12. Chen D., Dong, L., Mi J. (2020), “Incremental mechanism of attribute reduction based on discernible relations for dynamically increasing attribute”, *Soft Computing* 24, pp. 321-332.
13. Chen D., Yang Y., Dong Z. (2016), “An incremental algorithm for attribute reduction with variable precision rough sets”, *Appl. Soft Comput.*, vol. 45, pp. 129-149.
14. Dai H., Yan J., Li Z., Liao B. (2018), “Dominance-based fuzzy rough set approach for incomplete interval-valued data”, *Journal of Intelligent & Fuzzy Systems*, 34, pp. 423-436.
15. Das A., Sengupta, S., Bhattacharyya S. (2018), “A group incremental feature selection for classification using rough set theory based genetic algorithm”, *Applied Soft Computing*, 65, pp. 400-411.
16. Demetrovics, J., Thi V.D., Giang, N.L. (2014), “Metric Based Attribute Reduction in Dynamic Decision systems”, *Annales Univ. Sci. Budapest., Sect. Comp.*, Vol. 42, pp. 157-172.
17. Dinh V.V., Giang N.L., Thi V.D. (2013), “Generalized Discernibility Function based Attribute Reduction in Incomplete Decision Systems”, *Serdica Journal of Computing* 7, *Institute of Mathematics and Informatics, Bulgarian Academy of Sciences*, No 4, 2013, pp. 375-388.
18. Feng W., Zhang M. (2019), “Reduction algorithm based on finding the maximum mutual information in incomplete information systems”, *In Journal of Physics: Conference Series* (Vol. 1237, No. 2, p. 022020). IOP Publishing.
19. Giang N.L., Son N.H. (2013), “Metric based attribute reduction in incomplete decision tables”, *in International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing. Springer*, pp. 99 -110
20. Giang, N. L., Ngan, T. T., Tuan, T. M., Phuong, H. T., Abdel-Basset, M., de Macêdo, A. R. L., & Albuquerque, V. (2020), “Novel Incremental Algorithms for Attribute Reduction from Dynamic Decision systems using Hybrid Filter-Wrapper with Fuzzy Partition Distance”, *IEEE Transactions on Fuzzy Systems*, 28 (5), pp. 858-873.

21. Guan, Lihe. (2019), “A heuristic algorithm of attribute reduction in incomplete ordered decision systems”, *Journal of Intelligent & Fuzzy Systems*, 36(4), pp. 3891-3901.
22. Guo X., Xiang Y., Shu L. (2019), “An Information Quantity-Based Uncertainty Measure to Incomplete Numerical Systems”, *International Conference on Fuzzy Information & Engineering*, pp. 23-29.

23. Hao G., Longshu L., Chuanjian Y., Jian D. (2019), “Incremental reduction algorithm with acceleration strategy based on conflict region”, *Artificial Intelligence Review*, 51(4), pp. 507-536.
24. Hu C., Liu S., Liu G. (2017), “Matrix-based approaches for dynamic updating approximations in multigranulation rough sets”, *Knowl Based Syst* 122, pp. 51-63.
25. Hu J., Wang K., Yu H. (2017), “Attribute Reduction on Distributed Incomplete Decision Information System”, *In International Joint Conference on Rough Sets, Springer*, Cham, pp. 289-305.
26. Hua B., Lin W., Ga Y. (2019), “Attribute reduction based on improved information entropy”, *Journal of Intelligent & Fuzzy Systems*, 36(1), pp. 709-718.
27. Huang Y., Li T., Luo C., Fujita H., Horng S. (2017), “Dynamic variable precision rough set approach for probabilistic set-valued information systems”, *Knowledge-Based Systems* 122, pp. 131-147.
28. Huang Y., Li T., Luo C., Fujita H., Horng S. (2017), “Matrix-based dynamic updating rough fuzzy approximations for data mining”, *Knowl. Based Syst.* 119, pp. 273-283.
29. Huong N.T.L., Giang N.L. (2016), “Incremental algorithms based on metric for finding reduct in dynamic decision systems”, *Journal on Research and Development on Information & Communications Technology*, Vol.E-3, No.9, pp. 26-39.
30. Huong N.T.L., Giang N.L. (2016), “Incremental algorithms based on metric for finding reduct in dynamic decision tables”, *Journal on Research and Development on Information & Communications Technology*, Vol.E-3, No.9 (13), pp. 26-39.
31. Huyen T., Thinh C., Yamada K., Do N.V. (2018), “Incremental Updating Methods with Three-way Decision Models in Incomplete Information Systems”, *IEEE Joint 10th International Conference on Soft Computing and Intelligent Systems*, pp. 27-32.
32. Janos D., Huong N.T.L., Thi V.D., Giang N. L. (2016), “Metric based attribute reduction method in dynamic decision tables”, *Cybernetics and Information Technologies*, 16(2), pp. 3-15.
33. Jensen, R., Shen, Q. (2008), “Computational Intelligence and Feature Selection, Rough and Fuzzy Approaches, Aberystwyth University”, *IEEE Computational Intelligence Society, Sponsor*.

34. Jing Y., Li T., Fujita H., Wang B., Cheng N. (2018), “An incremental attribute reduction method for dynamic data mining”, *Information Sciences* 465, pp. 202-218.
35. Jing Y., Li T., Huang J., Chen H., Horng S. (2017), “A Group Incremental Reduction Algorithm with Varying Data Values”, *International Journal of Intelligent Systems* 32(9), pp. 900-925.

36. Jing Y., Li T., Huang J., et al. (2016), “An incremental attribute reduction approach based on knowledge granularity under the attribute generalization”, *Int. J. Approx. Reason.* 76, pp. 80-95.
37. Jing Y., Li T., Luo C., Horng S., Wang G., Yu Z. (2016), “An incremental approach for attribute reduction based on knowledge granularity”, *Knowledge-Based Systems*, Vol.104, pp. 24-38.
38. Kryszkiewicz M. (1998), “Rough set approach to incomplete information systems”, *Information Science*, Vol. 112, pp. 39-49.
39. Lang G., Cai M., Fujita H., Xiao Q. (2018), “Related families-based attribute reduction of dynamic covering decision information systems”, *Knowledge-Based Systems*, 162, pp. 161-173.
40. Lang G., Li Q., Cai M., Yang T., Xiao Q. (2017), “Incremental approaches to knowledge reduction based on characteristic matrices”, *International Journal of Machine Learning and Cybernetics*, 8(1), pp. 203-222.
41. Lang G., Miao D., Cai M., Zhang Z. (2017), “Incremental approaches for updating reducts in dynamic covering information systems”, *Knowledge-Based Systems*, 134, pp. 85-104.
42. Lang G., Miao D., Yang T., Cai M. (2016), “Knowledge reduction of dynamic covering decision information systems when varying covering cardinalities”, *Information Sciences* 346-47, pp. 236-260.
43. Li S., Li T. (2015), “Incremental update of approximations in dominance-based rough sets approach under the variation of attribute values”, *Inf. Sci.* 294, pp.348-361.
44. Liang J., Wang F., Dang C., Qian Y. (2014), “A group incremental approach to feature selection applying rough set technique”, *IEEE Transactions on Knowledge and Data Engineering*, 26(2), pp. 294-308.
45. Liu D., Li T., Zhang J. (2014), “A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems”, *International Journal of Approximate Reasoning*, 55(8), pp. 1764-1786.

46. Liu G., Wang C. (2020), “A Novel Multi-Scale Feature Fusion Method for Region Proposal Network in Fast Object Detection”, *International Journal of Data Warehousing and Mining (IJDWM)*, 16(3), pp. 132-145.
47. Liu W. (2016), “An incremental approach to obtaining attribute reduction for dynamic decision systems”, *Open Math* 2016, 14, pp. 875-888.
48. Liu Y., Zhao S., Chen H., Li C., Lu Y. (2017), “Fuzzy rough incremental attribute reduction applying dependency measures”, *In Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*, pp. 484-492.
49. Liu Y., Zheng L., Xiu Y., Yin H., Zhao S., Wang X., Chen H., Li, C. (2020), “Discernibility matrix based incremental feature selection on

- fused decision tables”, *International Journal of Approximate Reasoning* 118, pp. 1-26.
- 50. Long N., Gianola D., Weigel K. (2011), “Dimension reduction and variable selection for genomic selection: application to predicting milk yield in Holsteins”, *Journal of Animal Breeding and Genetics*. 128 (4), pp. 247-257.
 - 51. Luo C., Li T., Chen H., Fujita H., Yi Z. (2017), “Efficient updating of probabilistic approximations with incremental objects”, *Knowledge-Based Systems* 109, pp. 71-83.
 - 52. Luo C., Li T., Huang Y., Fujita H. (2019), “Updating three-way decisions in incomplete multi-scale information systems”, *Information Sciences* 476, pp. 274-289.
 - 53. Luo C., Li T., Yao Y. (2017), “Dynamic probabilistic rough sets with incomplete data”, *Information Sciences* 417, pp. 39-54.
 - 54. Luo S. (2018), “Attribute reductions in an inconsistent decision information system”, *Journal of Intelligent & Fuzzy Systems*, 35(3), pp. 3543-3552.
 - 55. Ma F., Chen J., Han W. (2016), “A Positive Region Based Incremental Attribute Reduction Algorithm for Incomplete System”, *International Conference on Electronic Information Technology and Intellectualization (ICEITI 2016)*, pp. 153-158.
 - 56. Ma F., Ding M., Zhang T., Cao J. (2019), “Compressed binary discernibility matrix based incremental attribute reduction algorithm for group dynamic data”, *Neurocomputing*, Vol. 344, No. 7, pp. 20-27.
 - 57. Ma F., Zhang T. (2017), “Generalized binary discernibility matrix for attribute reduction in incomplete information systems”, *The Journal of China Universities of Posts and Telecommunications*, Volume 24, Issue 4, pp. 57-75.
 - 58. Meng Z., Shi Z. (2009), “A fast approach to attribute reduction in incomplete decision systems with tolerance relation-based rough sets”, *Information Sciences*, Vol. 179, pp. 2774-2793.
 - 59. Nandhini N., Thangadurai K. (2019), “An incremental rough set approach for faster attribute reduction”, *International Journal of Information Technology*. <https://doi.org/10.1007/s41870-019-00326-6>.
 - 60. Ni P., Zhao S., Wang X., Chen H., Li C., Tsang E. (2020), “Incremental feature selection based on fuzzy rough sets”, *Information Sciences*,

- Volume 536, pp. 185-204. <https://doi.org/10.1016/j.ins.2020.04.038>
- 61. Pawlak Z. (1982), “Rough sets”, *International Journal of Computer and Information Sciences*, 11(5), pp. 341-356.
 - 62. Pawlak Z. (1991), “Rough sets: Theoretical aspects of reasoning about data”, *The Netherlands: Kluwer Academic Publishers*.

63. Qian Y.H., Liang J.Y., Li D.Y., Zhang H.Y. and Dang C.Y. (2008), “Measures of Evaluating The Decision Performance of a Decision Table in Rough Set Theory”, *Information Sciences*, Vol.178, pp.181-202.
64. Qian W., Shu W. (2015), “Mutual information criterion for feature selection from incomplete data”, *Neurocomputing*, Volume 168, pp. 210-220.
65. Raza M., Qamar U. (2016), “An incremental dependency calculation technique for feature selection using rough sets”, *Information Sciences* 343–344, pp. 41- 65.
66. Shu W., Qian W. (2015), “An incremental approach to attribute reduction from dynamic incomplete decision systems in rough set theory”. *Data & Knowledge Engineering*, 100, pp. 116-132.
67. Shu W., Qian W., Xie Y. (2019), “Incremental approaches for feature selection from dynamic data with the variation of multiple objects”, *Knowledge-Based Systems*, 163, pp. 320-331.
68. Shu W., Qian W., Xie Y. (2020), “Incremental feature selection for dynamic hybrid data using neighborhood rough set”, *Knowledge-Based Systems*. Volume 194, 105516
69. Shu W., Shen H. (2014), “Incremental feature selection based on rough set in dynamic incomplete data”, *Pattern Recognition* 47, pp.3890-3906.
70. Shu W., Shen H. (2014), “Updating attribute reduction in incomplete decision systems with the variation of attribute set”, *International Journal of Approximate Reasoning*, 55(3), pp. 867-884.
71. Song Y., Li Y., Qu J. (2018), “A New Approach for Supervised Dimensionality Reduction”, *International Journal of Data Warehousing and Mining (IJDWM)* 14(4), pp. 20-37.
72. Tao Y., Chongzhao H. (2017), “Entropy based attribute reduction approach for incomplete decision table”, In 2017 20th International Conference on Information Fusion (Fusion), IEEE, pp. 1-8.
73. The UCI machine learning repository,
[<https://archive.ics.uci.edu/ml/datasets.php>](https://archive.ics.uci.edu/ml/datasets.php)
74. Thien N.V., Giang N.L., Son N.N. (2018), “Fuzzy Partition Distance based Attribute Reduction in Decision Tables”, *IJCRS 2018: International Joint Conference on Rough Sets 2018*, LNCS, Vol. 11103, Springer Link, pp. 614-627.

75. Tiwar K., Shreevastava S., Shukla K., Subbiah K. (2018), “New approaches to intuitionistic fuzzy-rough attribute reduction”, *Journal of Intelligent & Fuzzy Systems*, 34(5), pp. 3385-3394.
76. Visalakshi S., Radha V. (2017), “A hybrid filter and wrapper feature selection approach for detecting contamination in drinking water management system”, *Journal of Engineering Science and Technology*, Vol. 12, No. 7, pp. 1819 - 1832.

77. Wang F., Liang J., Dang C. (2013), “Attribute reduction for dynamic data sets”, *Applied Soft Computing*, 13(1), pp. 676-689. <https://doi.org/10.1016/j.asoc.2012.07.018>
78. Wang F., Liang J., Qian Y. (2013), “Attribute reduction: A dimension incremental strategy”, *Knowledge-Based Systems*, Volume 39, pp. 95-108.
79. Wang L., Yang X., Chen Y., Liu L., An S., Zhuo P. (2018), “Dynamic composite decision-theoretic rough set under the change of attributes”, *Int. J. Comput. Intell. Syst*, Vol. 11, pp. 355-370.
80. Wang S. (2020), “Research on Data Mining and Investment Recommendation of Individual Users Based on Financial Time Series Analysis”, *International Journal of Data Warehousing and Mining* (IJDWM), 16(2), pp. 64-80.
81. Wei-Yin Loh., (2011), “Classification and regression trees”, *John Wiley & Sons, Inc. WIREs Data Mining Knowl Discov* Volume 1, pp. 14-23. DOI: 10.1002/widm.8
82. Wei W., Song P., Liang J., Wu X. (2018), “Accelerating incremental attribute reduction algorithm by compacting a decision table”, *International Journal of Machine Learning and Cybernetics*, Springer.
83. Wei W., Song P., Liang J., Wu X. (2019), “Accelerating incremental attribute reduction algorithm by compacting a decision system”, *International Journal of Machine Learning and Cybernetics* 10, pp. 2355-2373.
84. Wei W., Wu X., Liang, J., Cui J., Sun Y. (2018), “Discernibility matrix based incremental attribute reduction for dynamic data”, *Knowledge-Based Systems*, Vol. 140, pp. 142-157.
85. Xia W., Lu J., Ming J. (2020), “Attributes correlation coefficients and their application to attributes reduction”, *Journal of Intelligent & Fuzzy Systems*, 38(3), pp. 2443-2455.
86. Xie X., Qin X. (2018), “A novel incremental attribute reduction approach for dynamic incomplete decision systems”, *International Journal of Approximate Reasoning*, 93, pp. 443-462.
87. Yang X., Li T., Fujita H., Liu D., Yao Y. (2017), “A unified model of sequential three-way decisions and multilevel incremental processing”, *Knowledge-Based Systems* 134, pp. 172-188.

88. Yang X., Li T., Liu D., Chen H., Luo C. (2017), “A unified framework of dynamic three-way probabilistic rough sets”, *Information Sciences* 420, pp. 126-147.
89. Yang Y., Chen D., Wang H. (2017), “Active Sample Selection Based Incremental Algorithm for Attribute Reduction With Rough Sets”, *IEEE Transactions on Fuzzy Systems*, 25(4), pp. 825-838.

90. Yang Y., Chen D., Wang H., Tsang E., Zhang D. (2017), “Fuzzy rough set based incremental attribute reduction from dynamic data with sample arriving”, *Fuzzy Sets and Systems*, 312, pp. 66-86.
91. Yang Y., Chen D., Wang H., Wang X. (2017), “Incremental perspective for feature selection based on fuzzy rough sets”, *IEEE Transactions on Fuzzy Systems*, 26(3), pp. 1257-1273.
92. Yang, C., Ge H., Li L., Ding J. (2019), “A unified incremental reduction with the variations of the object for decision tables”, *Soft Computing* 23, pp. 6407-6427.
93. You Z., Hu Y., Tsai C., Kuo Y. (2020), “Integrating Feature and Instance Selection Techniques in Opinion Mining”, *International Journal of Data Warehousing and Mining (IJDWM)*, 16(3), pp. 168-182.
94. Yu J., Sang L., Dong H. (2018), “Based on attribute order for dynamic attribute reduction in the incomplete information system”, *In 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, IEEE, pp. 2475-2478. <https://doi.org/10.1007/s13042-020-01089-4>.
95. Zeng A., Li T., Hu J., Chen H., Luo C. (2016), “Dynamical updating fuzzy rough approximations for hybrid data under the variation of attribute values”, *Information Sciences*, pp. 1-26. <https://doi.org/10.1016/j.ins.2016.07.056>
96. Zeng A., Li T., Liu D., Zhang J., Chen H. (2015), “A fuzzy rough set approach for incremental feature selection on hybrid information systems”, *Fuzzy Sets and Systems*, Volume 258, pp. 39-60. <https://doi.org/10.1016/j.fss.2014.08.014>
97. Zhang C., & Dai J. (2019), “An incremental attribute reduction approach based on knowledge granularity for incomplete decision systems”, *Granular Computing*, pp. 1-15.
98. Zhang C., Dai J., Chen J. (2020), “Knowledge granularity based incremental attribute reduction for incomplete decision systems”, *International Journal of Machine Learning and Cybernetics*. <https://doi.org/10.1007/s13042-020-01089-4>.
99. Zhang D., Li R., Tang X., Zhao Y. (2008), “An incremental reduct algorithm based on generalized decision for incomplete decision tables”, *In 2008 3rd International Conference on Intelligent System and Knowledge Engineering*, IEEE, Vol. 1, pp. 340-344.

100. Zhang X., Mei C., Chen D., Li J. (2016), “Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy”, *Pattern Recognition* 56, pp. 1-15.
101. Zhang X., Mei C., Chen D., Yang Y., Li J. (2020), “Active Incremental Feature Selection Using a Fuzzy-Rough-Set-Based Information Entropy”, *IEEE Transactions on Fuzzy Systems*, Volume 28, Issue 5, pp. 901-915.

THÔNG TIN HỎI ĐÁP:

Bạn còn nhiều thắc mắc hoặc muốn tìm kiếm thêm nhiều tài liệu luận văn mới mẻ khác của Trung tâm [Best4Team](#),

Liên hệ [dịch vụ viết thuê luận văn](#)

Hoặc qua SDT Zalo: 091.552.1220 hoặc email: best4team.com@gmail.com để hỗ trợ ngay nhé!

