

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC LẠC HỒNG



ĐỒ SĨ TRƯỜNG

**PHƯƠNG PHÁP LỰA CHỌN THUỘC TÍNH
VÀ KỸ THUẬT GOM CỤM DỮ LIỆU PHÂN LOẠI
SỬ DỤNG TẬP THÔ**

LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

Đồng Nai – năm 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC LẠC HỒNG



ĐỒ SĨ TRƯỜNG

**PHƯƠNG PHÁP LỰA CHỌN THUỘC TÍNH
VÀ KỸ THUẬT GOM CỤM DỮ LIỆU PHÂN LOẠI
SỬ DỤNG TẬP THÔ**

LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

Chuyên ngành: Khoa học máy tính

Mã số ngành: 9480101

**NGƯỜI HƯỚNG DẪN KHOA HỌC
PGS.TS NGUYỄN THANH TÙNG**

Đồng Nai, năm 2023

LỜI CẢM ƠN

Xin trân trọng cảm ơn PGS.TS. Nguyễn Thanh Tùng đã tận tình hướng dẫn nghiên cứu sinh hoàn thành luận án tiến sĩ.

Xin trân trọng cảm ơn quý thầy/cô khoa sau đại học, trường đại học Lạc Hồng đã tạo điều kiện thuận lợi và hỗ trợ nghiên cứu sinh hoàn thành luận án.

Xin trân trọng cảm ơn trường đại học Lạc Hồng đã tạo điều kiện thuận lợi trong công tác và hỗ trợ nghiên cứu sinh tham gia học tập.

Xin chân thành cảm ơn quý bạn bè, đồng nghiệp đã tạo điều kiện mọi mặt giúp nghiên cứu sinh hoàn thành luận án.

Đồng Nai, ngày tháng năm 2023

Nghiên cứu sinh

Đỗ Sĩ Trường

LỜI CAM ĐOAN

Tôi xin cam đoan luận án này là công trình nghiên cứu của riêng tôi dưới sự hướng dẫn của PGS.TS. Nguyễn Thanh Tùng. Các số liệu và tài liệu trong nghiên cứu là trung thực và chưa được công bố trong bất kỳ công trình nghiên cứu nào. Tất cả các tham khảo và kế thừa đều được trích dẫn và tham chiếu đầy đủ.

Đồng Nai, ngày tháng năm 2023

Nghiên cứu sinh

Đỗ Sĩ Trường

MỤC LỤC

CHƯƠNG 1.	MỞ ĐẦU.....	1
CHƯƠNG 2.	KHÁI QUÁT VỀ LÝ THUYẾT TẬP THÔ VÀ ỨNG DỤNG TRONG KHAI PHÁ DỮ LIỆU	9
2.1	Mở đầu.....	9
2.2	Các khái niệm cơ bản của lý thuyết tập thô.....	9
2.2.1	Hệ thông tin	9
2.2.2	Quan hệ không phân biệt được và các xấp xỉ của một tập hợp	10
2.2.3	Bảng quyết định.....	11
2.2.4	Các khái niệm lý thuyết thông tin liên quan.....	13
2.3	Một số thuật toán hiệu quả của lý thuyết tập thô.....	16
2.4	Ứng dụng của lý thuyết tập thô trong khám phá tri thức từ cơ sở dữ liệu.....	19
2.5	Kết luận chương 2.....	21
CHƯƠNG 3.	LỰA CHỌN THUỘC TÍNH SỬ DỤNG LÝ THUYẾT TẬP THÔ.....	23
3.1	Mở đầu.....	23
3.2	Khái quát về bài toán lựa chọn thuộc tính	24
3.3	Các phương pháp lựa chọn thuộc tính sử dụng lý thuyết tập thô	27
3.3.1	Phương pháp lựa chọn thuộc tính sử dụng ma trận phân biệt	28
3.3.2	Phương pháp rút gọn thuộc tính dựa vào độ phụ thuộc.....	32
3.3.3	Phương pháp rút gọn thuộc tính sử dụng sử dụng độ phụ thuộc tương đối.....	34
3.3.4	Phương pháp rút gọn thuộc tính sử dụng Entropy thông tin	37
3.3.5	Phương pháp lựa chọn thuộc tính dựa trên gom cụm.....	39
3.4	Đề xuất thuật toán rút gọn thuộc tính dựa vào gom cụm ACBRC	42
3.4.1	Ý tưởng và những định nghĩa cơ bản	42
3.4.2	Giới thiệu thuật toán k-medoids	43
3.4.3	Thuật toán rút gọn thuộc tính dựa vào gom cụm ACBRC	45
3.4.4	Kết quả thực nghiệm thuật toán ACBRC	48
3.5	Kết luận chương 3.....	52
CHƯƠNG 4.	GOM CỤM DỮ LIỆU SỬ DỤNG LÝ THUYẾT TẬP THÔ	54
4.1	Mở đầu.....	54
4.2	Khái quát bài toán gom cụm dữ liệu.....	55

4.2.1	Các bước giải bài toán gom cụm dữ liệu	55
4.2.2	Các loại phương pháp gom cụm dữ liệu.....	56
4.2.3	Các tiêu chí đánh giá một thuật toán gom cụm hiệu.....	58
4.3	Gom cụm dữ liệu phân loại sử dụng Lý thuyết tập thô	59
4.3.1	Thuật toán lựa chọn thuộc tính gom cụm TR	61
4.3.2	Thuật toán lựa chọn thuộc tính gom cụm MDA.....	63
4.3.3	Thuật toán MMR (Min-Min-Roughness)	64
4.3.4	Thuật toán MGR (Mean Gain Ratio).....	67
4.4	Đề xuất thuật toán MMNVI gom cụm dữ liệu phân loại.....	69
4.4.1	Ý tưởng và những định nghĩa cơ bản	69
4.4.2	Thuật toán MMNVI.....	70
4.4.3	Độ phức tạp của thuật toán MMNVI.....	75
4.4.4	Nhận xét thuật toán MMNVI.....	76
4.4.5	Kết quả thực nghiệm thuật toán MMNVI.....	76
4.4.5.1	Bộ dữ liệu đánh giá.....	77
4.4.5.2	Phương pháp đánh giá hiệu suất	77
4.4.5.3	Kết quả gom cụm.....	79
4.4.5.4	So sánh MMNVI với thuật toán MMR và MGR.....	82
4.5	Kết luận chương 4.....	85
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		87
5.1	Những kết quả và đóng góp chính của luận án.....	87
5.2	Hướng phát triển của luận án.....	88

BẢNG THUẬT NGỮ ANH - VIỆT

Tiếng Anh	Viết tắt	Tiếng việt
Adjusted Rand Index	ARI	Chỉ số ngẫu nhiên hiệu chỉnh
Attribute clustering		Gom cụm thuộc tính
Attribute reduction		Rút gọn thuộc tính
Attribute Clustering Based Reduct Computing	ACBRC	Tính toán tập rút gọn dựa trên gom cụm thuộc tính
Categorical Data		Dữ liệu phân loại/phạm trù
Clustering data		Gom cụm dữ liệu
Data mining	KPDL	Khai phá dữ liệu
Database	CDSL	Cơ sở dữ liệu
Decision table	DT	Bảng quyết định
Feature selection		Lựa chọn thuộc tính/đặc trưng
Information system	IS	Hệ thống tin
Knowledge Discovery in Databases	KDD	Khám phá tri thức từ Cơ sở dữ liệu
Normalized Mutual Information	NMI	Thông tin tương hỗ chuẩn hóa
Machine learning	ML	Học máy

Minimum Mean Normalized Variation of Information	MMNVI	
Mean Gain Ratio	MGR	
Min-Min-Roughness	MMR	
Normalized Variation of Information	NVI	Biến thể thông tin chuẩn hóa
Overall Purity	OP	Độ thuần khiết tổng thể
Rough Sets Theory	LTTT	Lý thuyết tập thô

BẢNG CÁC KÝ HIỆU

Ký hiệu, từ viết tắt	Diễn giải
$IS = (U, A)$	Hệ thông tin
$ U $	Số đối tượng
$ d $	Thuộc tính điều kiện trong bảng quyết định
$ A $	Số thuộc tính trong hệ thông tin
$u(a)$	Giá trị của đối tượng u tại thuộc tính a
$IND(B)$	Quan hệ B – không phân biệt
$[u]_B$	Lớp tương đương chứa u của quan hệ $IND(B)$
U/B	Phân hoạch của U sinh bởi tập thuộc tính B .
$\underline{B}X$	B – xấp xỉ dưới của X
$\overline{B}X$	B – xấp xỉ trên của X
$\alpha_B(X)$	Độ chính xác của xấp xỉ X thông qua B
$R_B(X)$	Độ thô (roughness) của X đối với B
$POS_B(D)$	B – miền dương của D
$Core(C)$	Tập lõi
$\gamma_B(d)$	Độ phụ thuộc của d vào B
$H(a)$	Shannon Entropy của tập thuộc tính a
$H(a, b)$	Entropy đồng thời của a và b
$H(a b)$	Entropy có điều kiện của a khi đã biết b
$I(a; b)$	Thông tin tương hỗ giữa hai thuộc tính a và b
$NVI(a, b)$	Biến thể thông tin chuẩn hóa giữa a và b
$Rough_{a_j}(a_i)$	Độ thô trung bình của thuộc tính a_i đối với thuộc tính a_j
$R_{a_j}(X_k)$	Độ thô lớp tương đương X_k đối với a_j
$TR(a_i)$	Tổng độ thô TR của a_i với mọi thuộc tính $a_j \in A$
$MR(a_i)$	Độ thô cực tiểu
$GR_b(a)$	Tỷ lệ lợi thông tin của a_i đối với a_j
$MGR(a_i)$	Tỷ lệ lợi thông tin trung bình của a_i đối với mọi a_j
$MNVI(a_i)$	Biến thể thông tin chuẩn hóa trung bình giữa a_i với mỗi $a_j \in A$
$Entropy(X)$	Entropy của tập dữ liệu $X \subseteq U$
$argmin$	Xác định phần tử có giá trị nhỏ nhất trên một miền giá trị

DANH MỤC BẢNG BIỂU

Bảng 3.1 Bảng quyết định ví dụ 3.1.	30
Bảng 3.2 Ma trận phân biệt của Bảng quyết định 3.1.	31
Bảng 3.3 Bảng quyết định	34
Bảng 3.4 Bảng mô tả các tập dữ liệu thực nghiệm	49
Bảng 3.5 Những thuộc tính được chọn bởi ba giải thuật rút gọn thuộc tính	50
Bảng 3.6 Bảng so sánh thời gian thực hiện của các thuật toán (theo giây)	50
Bảng 3.7 Độ chính xác phân lớp khi chưa rút gọn thuộc tính	51
Bảng 3.8 Độ chính xác phân lớp với các thuộc tính được chọn bởi ACBRC	51
Bảng 3.9 Độ chính xác phân lớp bằng C5.0 sau khi sử dụng các phương pháp rút gọn thuộc tính khác nhau	52
Bảng 3.10 Độ chính xác phân lớp Bayes sử dụng các thuật toán rút gọn thuộc tính	52
Bảng 4.1 Hệ thông tin về chất lượng đầu vào của sinh viên	74
Bảng 4.2 Độ chắc chắn trung bình của các thuộc tính	75
Bảng 4.3 Tám bộ dữ liệu chuẩn UCI.....	77
Bảng 4.4 Bảng dự phòng	78
Bảng 4.5 Kết quả gom cụm MMNVI trên tập dữ liệu Soybean Small.....	80
Bảng 4.6 Kết quả gom cụm MMNVI trên tập dữ liệu Breast Cancer Wisconsin.	80
Bảng 4.7 Kết quả gom cụm MMNVI trên tập dữ liệu Car Evaluation.....	80
Bảng 4.8 Kết quả gom cụm MMNVI trên tập dữ liệu Vote.	81
Bảng 4.9 Kết quả gom cụm MMNVI trên tập dữ liệu Chess.	81
Bảng 4.10 Kết quả gom cụm MMNVI trên tập dữ liệu Mushroom.	81
Bảng 4.11 Kết quả gom cụm MMNVI trên tập dữ liệu Balance Scale	81
Bảng 4.12 Kết quả gom cụm MMNVI trên tập dữ liệu Zoo	81
Bảng 4.13 Độ thuần khiết tổng thể của 3 thuật toán trên 8 bộ dữ liệu.	82
Bảng 4.14 Chỉ số ngẫu nhiên hiệu chỉnh (ARI) của ba thuật toán trên 8 tập dữ liệu.	83
Bảng 4.15 Thông tin tương hỗ chuẩn hóa (NMI) của ba thuật toán trên 8 tập dữ liệu.	84

DANH MỤC HÌNH VẼ

Hình 3.1 Hình minh họa thuật toán ACBRC.....	47
Hình 4.1 Hình minh họa so sánh độ thuần khiết tổng thể của ba thuật toán trên tám tập dữ liệu thực nghiệm	83
Hình 4.2 Hình minh họa so sánh chỉ số ngẫu nhiên hiệu chỉnh trung bình của ba thuật toán trên tám tập dữ liệu thực nghiệm	84
Hình 4.3 Hình minh họa so sánh thông tin tương hỗ chuẩn hóa của ba thuật toán đối với các tập dữ liệu có sự phân bố lớp cân bằng	85

DANH MỤC THUẬT TOÁN

Thuật toán 2.1 Thuật toán xác định lớp tương đương	17
Thuật toán 2.2 Thuật toán xác định xấp xỉ dưới	17
Thuật toán 2.3 Thuật toán xác định xấp xỉ trên	18
Thuật toán 2.4 Thuật toán xác định miền dương	19
Thuật toán 3.1 Thuật toán QuickReduct	33
Thuật toán 3.2 Thuật toán RelativeReduct	36
Thuật toán 3.3 Thuật toán CEBARKNC	38
Thuật toán 3.4 Thuật toán gom cụm thuộc tính MNF	41
Thuật toán 4.1 Thuật toán TR (Total Roughness)	62
Thuật toán 4.2 Thuật toán MDA (Maximumdegree of Dependency of Attributes)	63
Thuật toán 4.3 Thuật toán MMR (Min–Min–Mean-Roughness)	65
Thuật toán 4.4 Thuật toán MGR (Mean Gain Ratio)	67
Thuật toán 4.5 Thuật MMNVI	71

CHƯƠNG 1. MỞ ĐẦU

Ngày nay, cùng với sự phát triển của khoa học công nghệ, mạng máy tính và truyền thông đã có những bước phát triển mạnh mẽ và được ứng dụng rộng rãi trong tất cả các lĩnh vực đời sống. Cùng với đó, nhu cầu và khả năng thu thập, lưu trữ dữ liệu của con người không ngừng tăng lên theo cấp số nhân. Với lượng dữ liệu khổng lồ hiện nay, yêu cầu đặt ra đối với các công cụ xử lý, phân tích thông tin ngày càng cao. Đặc biệt hơn, con người luôn mong muốn thu nhận một cách tự động những tri thức tiềm ẩn, mang tính dự đoán từ nguồn dữ liệu quý giá này. Trong những năm qua, khám phá tri thức (khai phá dữ liệu), học máy, trích xuất quy tắc từ dữ liệu v.v. đã thu hút nhiều sự chú ý của các nhà khoa học trong lĩnh vực trí tuệ nhân tạo. Trên cơ sở đó, nhiều phương pháp khám phá tri thức từ cơ sở dữ liệu (CSDL) đã ra đời.

Khám phá tri thức từ CSDL (Knowledge Discovery in Databases – KDD) là một lĩnh vực khoa học nhằm nghiên cứu để tạo ra những công cụ khai phá những thông tin, tri thức hữu ích, tiềm ẩn mang tính dự đoán trong các CSDL lớn [1, 2].

Một quá trình chuẩn khám phá tri thức từ CSDL bao gồm 5 công đoạn [1]:

Công đoạn 1 - Lựa chọn dữ liệu: Là quá trình lựa chọn một tập dữ liệu, hoặc kết hợp một số tập dữ liệu sẵn với nhau để tạo ra một tập dữ liệu đích phù hợp với mục tiêu khai phá.

Công đoạn 2 - Tiền xử lý dữ liệu: Giai đoạn này bao gồm việc loại bỏ hoặc làm giảm giá trị bị nhiễu; xử lý giá trị bị thiếu và rời rạc hóa thuộc tính nếu cần. Công đoạn này nhằm cải thiện chất lượng tổng thể của bất kỳ thông tin nào có thể được phát hiện từ CSDL.

Công đoạn 3 - Rút gọn dữ liệu: Hầu hết các tập dữ liệu có thể chứa một lượng dư thừa nhất định. Lượng dữ liệu dư thừa này không những không hỗ trợ quá trình khám phá tri thức mà trên thực tế còn có thể làm sai lệch kết quả khai phá. Mục đích của công đoạn này là tìm ra các thuộc tính (đặc trưng) hữu ích để đại diện cho dữ liệu và loại bỏ các thuộc tính không liên quan. Từ đó, tiết kiệm được thời gian xử lý trong công đoạn khai phá dữ liệu tiếp theo.

Công đoạn 4 - Khai phá dữ liệu: Áp dụng các kỹ thuật khai phá dữ liệu (trích xuất thông tin hữu ích tiềm ẩn từ cơ sở dữ liệu) được lựa chọn phù hợp với mục tiêu của nhiệm vụ khám phá tri thức. Việc lựa chọn kỹ thuật sử dụng có thể phụ thuộc vào nhiều yếu tố, bao gồm nguồn của tập dữ liệu và các giá trị mà nó chứa.

Công đoạn 5 - Đánh giá và diễn giải tri thức. Một khi tri thức đã được khám phá, nó sẽ được đánh giá về giá trị, tính hữu ích, tính mới và tính đơn giản. Điều này có thể yêu cầu lặp lại một số bước trên của quá trình khám phá tri thức. Những mẫu thông tin và mối quan hệ trong dữ liệu đã được phát hiện sẽ được chuyển sang và biểu diễn ở dạng gần gũi với người sử dụng như đồ thị, cây, bảng biểu, luật, v. v.

Trong 5 công đoạn trên của quá trình khám phá tri thức từ CSDL, công đoạn 4 là quan trọng nhất.

Các kết quả nghiên cứu cùng với những ứng dụng thành công thời gian qua cho thấy, khám phá tri thức từ CSDL là một lĩnh vực khoa học tiềm năng, mang lại nhiều lợi ích, đồng thời có ưu thế hơn hẳn so với các công cụ phân tích dữ liệu truyền thống. Tuy nhiên, với tốc độ tăng trưởng của dữ liệu hiện nay, việc nghiên cứu và ứng dụng các kỹ thuật khai phá dữ liệu cũng đang gặp nhiều khó khăn, thách thức, đòi hỏi các nhà nghiên cứu phải không ngừng nỗ lực nhằm tìm ra những công cụ để giải quyết các khó khăn, thách thức này.

Một trong những khó khăn, thách thức quan trọng đó chính là, cùng với sự bùng nổ nhanh chóng của công nghệ, kích thước của những tập dữ liệu con người thu thập được ngày càng lớn. Có thể thấy, trong hầu hết các ứng dụng như dữ liệu gen, phân lớp văn bản, truy xuất hình ảnh và truy xuất thông tin, chúng ta thường phải đối mặt với các tập dữ liệu có số lượng lớn các thuộc tính (hay đặc trưng). Điều này có thể dẫn đến các thuật toán khai phá hoặc học từ dữ liệu truyền thống trở nên chậm lại và không thể xử lý thông tin một cách hiệu quả. Vấn đề đặt ra là trước khi triển khai các thuật toán khai phá dữ liệu cần phải có phương pháp rút gọn thuộc tính của CSDL mà vẫn bảo toàn được những thông tin cần khai thác. Rút gọn thuộc tính có thể được thực hiện bằng cách sử dụng các kỹ thuật phù hợp, tùy thuộc vào yêu cầu của bài toán khai phá dữ liệu đặt ra. Những kỹ thuật này có thể được chia thành hai loại chính, đó là *biến đổi thuộc tính* và *lựa chọn thuộc tính* [1, 3, 4, 5].

Phép biến đổi thuộc tính cố gắng xây dựng một *không gian thuộc tính mới* bằng cách biến đổi không gian thuộc tính ban đầu thành không gian có số chiều thấp hơn. Phân tích thành phần chính và phân tích thành phần độc lập là hai phương pháp biến đổi thuộc tính được sử dụng rộng rãi [1, 4, 5].

Lựa chọn thuộc tính (hay còn gọi là rút gọn thuộc tính) là quá trình chọn ra một tập hợp con thuộc tính từ tập hợp các thuộc tính ban đầu, với mục tiêu loại bỏ càng nhiều càng tốt các thuộc tính không liên quan và dư thừa nhằm cải thiện chất lượng dữ liệu và giảm độ phức tạp về thời gian và không gian cho việc phân tích. Lựa chọn thuộc tính là vấn đề rất quan trọng: thứ nhất là do các thuộc tính không liên quan không góp phần vào việc làm tăng độ chính xác dự đoán; thứ hai là do hầu hết thông tin mà nó có thể cung cấp cho việc dự đoán đã được chứa trong các thuộc tính khác. Lựa chọn thuộc tính được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, chẳng hạn như phân loại văn bản (text categorization), truy cập hình ảnh (image retrieval), Tin-sinh học (bioinformatics), phát hiện xâm nhập mạng (intrusion detection), v. v. [1, 3, 5].

Trong công đoạn 4 của quá trình khai phá dữ liệu, hai kỹ thuật quan trọng, thường được sử dụng nhất là kỹ thuật phân lớp (Classification) và kỹ thuật gom cụm dữ liệu (Data clustering) [1].

Phân lớp là phương pháp phân tích dữ liệu để trích xuất các quy tắc sắp xếp các đối tượng vào một trong các lớp đã biết dựa trên các giá trị sẵn có của các thuộc tính. Phân lớp còn được gọi là học có giám sát (supervised learning). Một số kỹ thuật cơ bản để phân lớp dữ liệu là quy nạp cây quyết định (decision tree induction), phân lớp Bayes, mạng nơ-ron nhân tạo (Neural network), và phương pháp máy véc tơ hỗ trợ (Support vector machines - SVM).

Gom cụm dữ liệu là phương pháp nhóm các đối tượng tương tự nhau trong tập dữ liệu vào các cụm sao cho các đối tượng thuộc cùng một cụm là tương đồng còn các đối tượng thuộc các cụm khác nhau sẽ không tương đồng. Gom cụm dữ liệu là một phương pháp học không có giám sát (unsupervised learning). Không giống như phân lớp dữ liệu, gom cụm dữ liệu không đòi hỏi phải biết trước nhãn lớp của các mẫu dữ liệu huấn luyện. Khi bắt đầu quá trình ta không biết trước các cụm dữ liệu sẽ như thế nào. Vì vậy, thông

thường cần có các chuyên gia về lĩnh vực giúp đánh giá các cụm thu được sau khi thực hiện một kỹ thuật gom cụm. Gom cụm dữ liệu được sử dụng nhiều trong các ứng dụng, chẳng hạn trong phân loại các loài thực vật, phân đoạn khách hàng, phân loại trang web v.v. Ngoài ra, gom cụm dữ liệu còn có thể được sử dụng như một kỹ thuật trong bước tiền xử lý cho các thuật toán khai phá dữ liệu khác.

Bài toán gom cụm dữ liệu cũng là bài toán NP-khó. Cho đến nay, có nhiều kỹ thuật gom cụm heuristic đã được đề xuất và giới thiệu trong các tài liệu về phân tích thống kê, khai phá dữ liệu, học máy [1, 6, 7]. Hầu hết các kỹ thuật gom cụm trong các tài liệu đều tập trung vào các tập dữ liệu số, trong đó mỗi thuộc tính mô tả các đối tượng đều có miền giá trị là một khoảng giá trị thực liên tục, mỗi đối tượng dữ liệu số được coi là một điểm trong không gian metric đa chiều với một metric đo khoảng cách giữa các đối tượng, chẳng hạn như metric Euclide hoặc metric Mahalanobis. Tuy nhiên, trong các ứng dụng thực tiễn thường gặp phải những tập dữ liệu với các thuộc tính là những thuộc tính phân loại hay phạm trù (categorical), tức là những thuộc tính có miền giá trị D hữu hạn và không có thứ tự (chẳng hạn như màu tóc, quốc tịch v.v.); trong D chỉ được phép so sánh giữa các giá trị, với bất kỳ $a, b \in D$ hoặc $a = b$ hoặc $a \neq b$. Với dữ liệu phân loại ta không thể định nghĩa hàm khoảng cách một cách tự nhiên.

Lý thuyết tập thô - do Zdzisaw Pawlak [8] đề xuất vào những năm đầu thập niên tám mươi của thế kỷ hai mươi - được xem là công cụ hữu hiệu để giải quyết các bài toán xử lý thông tin có chứa dữ liệu mơ hồ, không chắc chắn. Tính từ mơ hồ, không chắc chắn liên quan đến sự không nhất quán hoặc không rõ ràng. Do tư duy mới lạ, phương pháp độc đáo và dễ cài đặt, trong hơn ba mươi năm qua, lý thuyết tập thô đã được nghiên cứu, ứng dụng và trở thành một công cụ quan trọng trong lĩnh vực xử lý thông tin thông minh [2, 9, 10, 11, 12, 13]. Nó đã được áp dụng thành công trong một số lĩnh vực như học máy, hệ chuyên gia, nhận dạng mẫu, hệ thống hỗ trợ quyết định, khám phá tri thức trong cơ sở dữ liệu v.v. Trong nghiên cứu tính toán hạt (granular computing), lý thuyết tập thô đã trở thành một trong những mô hình và công cụ chính [10]. Triển vọng ứng dụng của lý thuyết tập hợp thô là rất rộng. Các tập thô không chỉ có thể được sử dụng để giải quyết vấn đề thông tin không chắc chắn, mà còn có thể giúp tối ưu hóa nhiều phương pháp tính toán mềm hiện

có. Ưu điểm chính của cách tiếp cận tập thô là nó không cần bất kỳ thông tin sơ bộ hoặc bổ sung nào về dữ liệu, như các giá trị xác suất trong thống kê, mức độ thuộc thành viên (degrees of membership) của các phần tử trong lý thuyết tập mờ.

Trong hơn ba mươi năm qua, nghiên cứu về các thuật toán và ứng dụng của lý thuyết tập thô luôn là đề tài phát triển mạnh mẽ và sôi động. Trong xu thế đó, nhiều nhóm nhà khoa học, trong đó có cả các nhà khoa học Việt nam, đã và đang quan tâm đến nghiên cứu vấn đề rút gọn thuộc tính trong bảng quyết định và gom cụm dữ liệu. Luận án tiến sĩ của Hoàng Thị Lan Giao [14] đã đề xuất các thuật toán heuristic tìm tập rút gọn và tìm tập rút gọn xấp xỉ của bảng quyết định nhất quán, bao gồm thuật toán sử dụng các phép toán trong đại số quan hệ và thuật toán sử dụng ma trận phân biệt. Luận án tiến sĩ của Nguyễn Đức Thuận [15] đề xuất thuật toán heuristic tìm tập rút gọn của bảng quyết định đầy đủ nhất quán dựa vào phủ tập thô. Luận án tiến sĩ của Nguyễn Long Giang [16] nghiên cứu phương pháp rút gọn thuộc tính trong bảng quyết định đầy đủ sử dụng metric.

Có thể thấy, ứng dụng lý thuyết tập thô trong khám phá tri thức từ CSDL trong thời gian qua đã thu hút sự quan tâm của các nhà nghiên cứu trong và ngoài nước. Tuy nhiên, đối với hai bài toán quan trọng là lựa chọn thuộc tính và gom cụm dữ liệu vẫn còn một số vấn đề lớn cần được tiếp tục thảo luận và cải tiến. Đó là:

Đối với bài toán lựa chọn thuộc tính, nhiều thuật toán lựa chọn thuộc tính hiện nay có thể loại bỏ thành công các thuộc tính không liên quan nhưng không thể loại bỏ các thuộc tính dư thừa [17, 18, 19, 20, 21]. Thuộc tính dư thừa không giúp cho quá trình dự đoán tốt hơn vì hầu hết các thông tin cần thiết đã được cung cấp bởi các thuộc tính còn lại. Điều này làm ảnh hưởng nghiêm trọng đến độ chính xác của một máy học. Vì vậy, yêu cầu đặt ra là phải nghiên cứu phương pháp lựa chọn thuộc tính mới, có thể loại bỏ hiệu quả đồng thời các thuộc tính không liên quan và cả các thuộc tính dư thừa [6, 7, 22, 23, 24].

Đối với bài toán gom cụm dữ liệu phân loại, mặc dù các thuật toán gom cụm đã được đề xuất có những đóng góp quan trọng trong vấn đề gom cụm dữ liệu phân loại nhưng chúng cũng có một số hạn chế như thường có độ chính xác thấp và độ phức tạp tính toán cao. Đặc biệt, trên một số tập dữ liệu chúng không thành công hoặc khó chọn được thuộc tính gom cụm tốt nhất [6, 7]. Vì vậy, cải tiến các thuật toán gom cụm dữ liệu phân loại

nhằm cho kết quả gom cụm tốt hơn các thuật toán cơ bản hiện có cũng là bài toán quan trọng cần giải quyết trong khám phá tri thức.

Với là lý do này, nghiên cứu sinh chọn đề tài nghiên cứu: “Phương pháp lựa chọn thuộc tính và kỹ thuật gom cụm dữ liệu phân loại sử dụng lý thuyết tập thô”.

Mục tiêu nghiên cứu của luận án tập trung giải quyết hai vấn đề của đề tài:

Mục tiêu thứ nhất là nghiên cứu phương pháp lựa chọn thuộc tính có thể loại bỏ hiệu quả đồng thời các thuộc tính không liên quan và cả các thuộc tính dư thừa.

Mục tiêu thứ hai là cải tiến các thuật toán gom cụm dữ liệu phân loại, đặc biệt là bài toán lựa chọn thuộc tính nhằm cho kết quả gom cụm tốt hơn các thuật toán cơ bản hiện có

Đối tượng nghiên cứu của luận án là các hệ thông tin, bảng quyết định có thể chứa dữ liệu mơ hồ, không chắc chắn.

Phạm vi nghiên cứu của luận án bao gồm việc nghiên cứu các phương pháp khai phá dữ liệu theo hướng tiếp cận tập thô, tập trung vào hai vấn đề chính nêu trong mục tiêu của luận án.

Phương pháp nghiên cứu các vấn đề nghiên cứu đặt ra được thực hiện bằng cách tổng hợp và đánh giá các kết quả nghiên cứu đã đạt được về lý thuyết tập thô trong khai phá dữ liệu từ các công trình đăng trên các tạp chí khoa học chuyên ngành uy tín trong và ngoài nước. Từ đó đề xuất các kỹ thuật, thuật toán mới, cài đặt, tính toán, so sánh và đánh giá kết quả thực nghiệm, chứng minh tính hiệu quả của các thuật toán.

Bố cục của luận án bao gồm chương mở đầu, ba chương nội dung chính, chương kết luận, các công trình nghiên cứu đã thực hiện và danh mục tài liệu tham khảo. Chương 2 trình bày các khái niệm cơ bản của lý thuyết tập thô cùng với một số khái niệm liên quan từ lý thuyết thông tin, khái quát về khai phá dữ liệu và tiềm năng ứng dụng lý thuyết tập thô trong khai phá dữ liệu. Chương 3 trình bày bài toán lựa chọn thuộc tính và một số thuật toán hiệu quả hiện có theo tiếp cận tập thô, những khó khăn thách thức; trên cơ sở đó đề xuất thuật toán mới rút gọn thuộc tính sử dụng phương pháp gom cụm thuộc tính. Chương 4 trình bày bài toán gom cụm trong khai phá dữ liệu, một số phương pháp gom cụm hiệu quả hiện có; hạn chế của chúng và đề xuất thuật toán gom cụm dữ liệu phân loại sử dụng

lý thuyết tập thô kết hợp các khái niệm entropy trong lý thuyết thông. Cuối cùng, chương kết luận nêu những đóng góp của luận án và các hướng phát triển.

Đóng góp chính của luận án được trình bày trong chương 3, chương 4.

Chương 3 đề xuất một thuật toán tìm toán tập rút gọn trong bảng quyết định bằng cách sử dụng phép gom cụm thuộc tính với tên gọi ACBRC (Attribute Clustering Based Reduct Computing – Tính toán tập rút gọn dựa vào gom cụm thuộc tính). Thuật toán đề xuất hoạt động trong ba công đoạn chính. Trong công đoạn đầu, các thuộc tính không liên quan sẽ bị loại bỏ. Tại công đoạn thứ hai, các thuộc tính có liên quan được phân chia thành một số cụm thích hợp bằng phương pháp gom cụm Phân hoạch Xung quanh Medoids (Partitioning Around Medoids - PAM) với một metric đặc biệt trong không gian thuộc tính là Biến thể Thông tin Chuẩn hóa (Normalized Variation of Information). Trong công đoạn thứ ba, một thuộc tính đại diện cho mỗi cụm được chọn là thuộc tính có độ liên quan lớn nhất với thuộc tính quyết định; các thuộc tính được lựa chọn tạo thành một tập rút gọn xấp xỉ.

Vì trong mỗi cụm gom được các thuộc tính là tương tự nhau, việc chỉ chọn một thuộc tính từ mỗi cụm đưa vào tập rút gọn tại công đoạn ba của thuật toán cho phép loại bỏ được các thuộc tính dư thừa đối với nhiệm vụ phân lớp dữ liệu. Đồng thời, bằng cách lấy tất cả đại diện của các cụm làm tập rút gọn thuật toán đã xét đến tất cả các thuộc tính liên quan, trong đó có thể có các thuộc tính kết hợp với nhau tác động đến kết quả phân lớp.

Để đánh giá thuật toán ACBRC, luận án đã tiến hành cài đặt, tính toán thực nghiệm trên các tập dữ liệu chuẩn lấy từ kho dữ liệu UCI [25]. Kết quả thực nghiệm cho thấy thuật toán đề xuất có khả năng tính toán tập rút gọn xấp xỉ có kích thước nhỏ và độ chính xác phân lớp cao so với các thuật toán đem so sánh, khi số cụm dùng để phân chia các thuộc tính được lựa chọn một cách thích hợp.

Chương 4 luận án đề xuất một thuật toán mới gom cụm dữ liệu phân loại với tên gọi MMNVI (Minimum Mean Normalized Variation of Information - Biến thể Thông tin Chuẩn hóa Trung bình Nhỏ nhất (MMNVI)). MMNVI thuộc loại phương pháp gom cụm

phân cấp, phân phân đôi dần tập các đối tượng thành các cụm. Tại mỗi bước lặp thuật toán thực hiện ba bước chính sau:

- Loại bỏ tất cả các thuộc tính chỉ nhận một giá trị;
- Chọn thuộc tính phân cụm là thuộc tính có giá trị biến thể thông tin chuẩn hóa trung bình (MNVI) nhỏ nhất;
- Lấy lớp tương đương sinh ra bởi thuộc tính phân cụm có tổng entropy của mỗi thuộc tính nhỏ nhất làm một cụm và hợp của tất cả các lớp tương đương còn lại làm tập dữ liệu cần phân chia tiếp.

Tại bước lặp đầu tiên, MMNVI lấy tập tất cả các đối tượng ban đầu làm tập dữ liệu cần phân chia. Quá trình phân cụm trên lặp lại cho đến khi đạt được số cụm quy định trước. Để thực hiện bước thứ hai, MMNVI sử dụng khái niệm “biến thể chuẩn hóa của thông tin” trong lý thuyết thông tin, một độ đo khoảng cách phổ quát trong không gian thuộc tính.

Kết quả thử nghiệm trên các tập dữ liệu thực từ UCI cho thấy thuật toán MMNVI có thể được sử dụng thành công trong việc gom cụm dữ liệu phân loại. Nó tạo ra kết quả gom cụm tốt hơn hoặc tương đương hơn với các thuật toán cơ bản đem so sánh.

Các đóng góp chính trên đây đã được đăng trong hai bài báo trên *Journal of Computer Science and Cybernetics*, năm 2022 và năm 2023. Ngoài các đóng góp chính trình bày trong luận án, nghiên cứu sinh là đồng tác giả của có một số kết quả khác liên quan đến đề tài luận án, bao gồm một bài báo quốc tế và ba báo cáo hội thảo khoa học trong nước.

CHƯƠNG 2. KHÁI QUÁT VỀ LÝ THUYẾT TẬP THÔ VÀ ỨNG DỤNG TRONG KHAI PHÁ DỮ LIỆU

2.1 Mở đầu

Lý thuyết tập thô – do Zdzisaw Pawlak [8] đề xuất vào những năm đầu thập niên tám mươi của thế kỷ hai mươi – được xem là công cụ hữu hiệu để giải quyết các bài toán chứa dữ liệu mơ hồ, không chắc chắn. Từ khi ra đời cho đến nay, lý thuyết tập thô được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau của khoa học máy tính như trí tuệ nhân tạo, hệ chuyên gia, hệ hỗ trợ quyết định, khám phá tri thức từ cơ sở dữ liệu, v.v.

Trong lý thuyết tập thô, mọi đối tượng của tập vũ trụ U đều hàm chứa một lượng thông tin nhất định (dữ liệu, tri thức) liên quan. Thông tin này có thể được thể hiện bằng một số thuộc tính (attribute) hay còn gọi là đặc trưng (feature). Các thuộc tính mô tả đối tượng. Các đối tượng có mô tả giống nhau được coi là không thể phân biệt được đối với thông tin có sẵn. Mỗi quan hệ không phân biệt là cơ sở toán học của lý thuyết tập thô. Nó tạo ra sự phân chia tập vũ trụ thành các khối đối tượng không thể phân biệt được, được gọi là các tập hợp cơ bản, có thể được sử dụng để xây dựng tri thức về một thế giới thực hoặc trừu tượng. Bất kỳ tập con X nào của tập vũ trụ U đều có thể được biểu thị theo các khối này một cách chính xác hoặc xấp xỉ.

Chương này trình bày các khái niệm cơ bản của lý thuyết tập thô, quy trình khám phá tri thức từ cơ sở dữ liệu và khả năng ứng dụng của của lý thuyết về tập thô trong khai phá dữ liệu. Các vấn đề cơ bản trình bày trong chương này là cơ sở cho việc nghiên cứu đề xuất các phương pháp mới rút gọn thuộc tính, gom cụm dữ liệu phân loại trình bày các chương sau.

2.2 Các khái niệm cơ bản của lý thuyết tập thô

2.2.1 Hệ thông tin

Một tập dữ liệu có thể được biểu diễn dưới dạng một bảng, trong đó mỗi hàng biểu diễn một đối tượng, một trường hợp hay một sự kiện, mỗi cột biểu diễn một thuộc tính, một tính chất hay một số đo có thể đo được trên mỗi đối tượng. Trong lý thuyết tập thô, một

bảng dữ liệu như vậy được gọi là một hệ thông tin. Một cách hình thức, người ta định nghĩa hệ thông tin như sau:

Định nghĩa 2.1. [8] *Hệ thông tin là một bộ đôi $IS = (U, A)$, trong đó U là một tập hữu hạn, không rỗng các đối tượng, A là một tập hữu hạn, không rỗng các thuộc tính, mỗi $a \in A$ là một ánh xạ $a : U \rightarrow V_a$, trong đó V_a ký hiệu miền giá trị của a .*

2.2.2 Quan hệ không phân biệt được và các xấp xỉ của một tập hợp

Định nghĩa 2.2. [8] *Cho hệ thông tin là một bộ tứ $IS = (U, A)$. Mỗi tập con các thuộc tính $B \subseteq A$ xác định một quan hệ, ký hiệu là $IND(B)$, gọi là quan hệ không phân biệt được, như sau:*

$$IND(B) = \{(u, v) \in U^2 : a(u) = a(v) \forall a \in B\} \quad (2.1)$$

Nếu hai đối tượng $(u, v) \in IND(B)$ thì hai đối tượng này sẽ không phân biệt được bởi các thuộc tính thuộc tập B .

Rõ ràng $IND(B)$ là một quan hệ tương đương, nó phân chia U thành các lớp tương đương rời nhau, trong đó hai đối tượng thuộc cùng một lớp nếu chúng có cùng giá trị đối với B . Gọi $U/IND(B)$ (hay viết tắt U/B) là họ của tất cả các lớp tương đương của $IND(B)$. Với mọi đối tượng $x \in U$, ký hiệu $[x]_B$ là lớp tương đương của quan hệ $IND(B)$ chứa phần tử x , và gọi $[x]_B$ là lớp tương đương của x trong quan hệ $IND(B)$.

Định nghĩa 2.3. [8] *Cho hệ thông tin là một bộ tứ $IS = (U, A, V, f)$, $B \subseteq A$ và $X \subseteq U$, B -xấp xỉ dưới của X , ký hiệu là $\underline{B}(X)$, và B -xấp xỉ trên của X , ký hiệu là $\overline{B}(X)$, được định nghĩa tương ứng như sau:*

$$\underline{B}X = \{u \in U : [u]_B \subseteq X\} \quad (2.2)$$

$$\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\} \quad (2.3)$$

Định nghĩa trên cho thấy nếu đối tượng $x \in \underline{B}X$ thì nó chắc chắn thuộc vào tập X , còn khi $x \in \overline{B}X$ thì nó có thể thuộc vào tập X . Hiển nhiên, ta có $\underline{B}X \subseteq X \subseteq \overline{B}X$. X được gọi là định nghĩa được nếu $\underline{B}X = \overline{B}X$, trường hợp ngược lại, X được gọi là tập thô với B -biên

$BN_B(X) = \overline{BX} - \underline{BX}$. Một cách tự nhiên, một tập thô X có thể được xấp xỉ bằng $\underline{BX}$ và/hoặc $\overline{BX}$.

Định nghĩa 2.4. [8] Cho hệ thông tin $IS = (U, A)$, $B \subseteq A$ và $X \subseteq U$. Độ chính xác của xấp xỉ X thông qua B được định nghĩa bởi

$$\alpha_B(X) = \frac{|\underline{BX}|}{|\overline{BX}|} \quad (2.4)$$

Trong suốt luận án này, $|X|$ ký hiệu số phần tử của tập X .

Định nghĩa 2.5. [8] Cho hệ thông tin $IS = (U, A)$, $B \subseteq A$ và $X \subseteq U$. Độ thô (roughness) của X đối với B được định nghĩa là

$$R_B(X) = \alpha_B(X) = 1 - \frac{|\underline{B}(X)|}{|\overline{B}(X)|} \quad (2.5)$$

Hiển nhiên, $0 \leq R_B(X) \leq 1$. Nếu $R_B(X) = 0$, thì $\underline{BX} = \overline{BX}$, B -biên của X là tập rỗng, và X là tập rõ đối với B . Nếu $R_B(X) < 1$, thì $\underline{BX} \subset \overline{BX}$, B -biên của X là khác rỗng, và X là tập thô đối với B .

2.2.3 Bảng quyết định

Định nghĩa 2.6. [8, 10] Bảng quyết định là một hệ thông tin dạng $DT = (U, C \cup \{d\})$, trong đó $d \notin C$ là một thuộc tính riêng biệt được gọi là thuộc tính quyết định. Các thuộc tính trong C được gọi là các thuộc tính điều kiện.

Định nghĩa 2.7. [8, 10] Cho $DT = (U, C \cup \{d\})$ là một bảng quyết định và tập con thuộc tính điều kiện $B \subseteq C$. Vùng dương của d đối với B , ký hiệu là $POS_B(d)$, được xác định như sau

$$POS_B(d) = \bigcup_{x \in U/d} (\underline{BX}) \quad (2.6)$$

Vùng dương $POS_B(d)$ bao gồm những đối tượng chắc chắn có thể được phân vào một số lớp quyết định bằng cách kiểm tra tất cả các thuộc tính có trong B . Nếu $POS_B(d) = U$, thì bảng quyết định DT là nhất quán, ngược lại DT là không nhất quán.

Định nghĩa 2.8. [8, 10] Cho $DT = (U, C \cup \{d\})$ là một bảng quyết định, thuộc tính $c \in C$ được gọi là không cần thiết trong bảng quyết định DT nếu

$$POS_c(d) = POS_{(C \setminus \{c\})}(d) \quad (2.7)$$

ngược lại, c được gọi là cần thiết.

Định nghĩa 2.9. [8, 10] Bảng quyết định $DT = (U, C \cup \{d\})$ được gọi là độc lập nếu mọi thuộc tính $c \in C$ đều cần thiết. Tập tất cả các thuộc tính cần thiết trong DT được gọi là tập lõi và được ký hiệu $Core(C)$. Lúc đó, một thuộc tính cần thiết còn được gọi là thuộc tính lõi.

Định nghĩa 2.10. [8, 10] Tập các thuộc tính $R \subseteq A$ được gọi là một rút gọn của bảng quyết định $DT = (U, C \cup \{d\})$ nếu nó là tập con tối thiểu thỏa mãn $POS_R(d) = POS_C(d)$. Như vậy, tập rút gọn là tập con tối thiểu các thuộc tính có khả năng phân lớp đúng các đối tượng trong U như toàn bộ tập thuộc tính C .

Rõ ràng là có thể có nhiều tập rút gọn của C . Tập tất cả các tập rút gọn của bảng quyết định DT được ký hiệu là $Red(C)$. Một thuộc tính là cần thiết khi và chỉ khi nó thuộc vào mọi tập rút gọn của C . Điều đó được thể hiện trong mệnh đề sau.

Mệnh đề 2.1. [8, 10] Cho bảng quyết định $DT = (U, C \cup \{d\})$. Ta có:

$$CORE(C) = \bigcap_{R \in red(C)} R. \quad (2.8)$$

Định nghĩa 2.11. [8, 10] Cho bảng quyết định $DT = (U, C \cup \{d\})$. Với tập con $B \subseteq C$, độ phụ thuộc $\gamma_B(d)$ của d vào B được định nghĩa như sau:

$$\gamma_B(d) = \frac{|POS_B(d)|}{|U|}. \quad (2.9)$$

Rõ ràng, $0 \leq \gamma_B(d) \leq 1$. Nếu $\gamma_B(d) = 1$, thì ta nói rằng d phụ thuộc hoàn toàn vào B , còn nếu $0 < \gamma_B(d) < 1$, thì d phụ thuộc vào B với mức độ $\gamma_B(d)$. Khi $\gamma_B(d) = 0$, ta nói rằng d không phụ thuộc vào B .

2.2.4 Các khái niệm lý thuyết thông tin liên quan

Cho $IS = (U, A)$ là một hệ thống thông tin, thuộc tính $a \in A$. Hệ thống thông tin IS có thể được xem như một quần thể thống kê và a là một biến ngẫu nhiên rời rạc. Giả sử $V_a = \{x_1, x_2, \dots, x_m\}$, $U/IND(a) = \{X_1, X_2, \dots, X_m\}$. Khi đó, phân phối xác suất của a có thể được xác định bởi:

$$P(a = x_i) = P(x_i) = |X_i|/|U|, \quad i = 1, \dots, m. \quad (2.10)$$

Các phân phối xác suất liên quan khác có thể được xác định tương tự. Cụ thể, $P(a, b)$ là phân phối xác suất chung của a và b , và $P(a|b)$ là phân phối xác suất có điều kiện của a cho trước b . Giả sử $U/IND(a) = \{X_1, X_2, \dots, X_m\}$ và $U/IND(b) = \{Y_1, Y_2, \dots, Y_n\}$, khi đó

$$\begin{aligned} P(a = x_i, b = y_j) &= P(x_i, y_j) = |X_i \cap Y_j|/|U|, \\ P(a = x_i | b = y_j) &= P(x_i|y_j) = |X_i \cap Y_j|/|Y_j|, \end{aligned}$$

$$i = 1, \dots, m, j = 1, \dots, n.$$

Định nghĩa 2.12. [26] Cho hệ thống thông tin $IS = (U, A)$ và thuộc tính $a \in A$. Shannon entropy (gọi tắt là entropy) của a là một đại lượng $H(a)$ xác định theo công thức sau:

$$H(a) = - \sum_{i=1}^m P(a = x_i) \log_2 P(a = x_i). \quad (2.11)$$

với quy ước $0 \times \log_2 0 = 0$.

Đối với thuộc tính, Entropy $H(a)$ là thước đo đo mức độ hỗn loạn (không chắc chắn) trong vector cột liên kết với thuộc tính a . Giá trị nhỏ nhất của entropy $H(a)$ là 0, giá trị này xảy ra khi tất cả các thành phần trong vector liên kết là như nhau, không có sự rối loạn. Giá trị lớn nhất của entropy là $\log_2 |V_a|$, xảy ra khi tất cả các thành phần trong vector liên kết

đều khác nhau. Giá trị entropy càng lớn thì mức độ hỗn loạn càng cao. Khái niệm về entropy có thể được khái quát cho trường hợp có hai và nhiều thuộc tính.

Định nghĩa 2.13. [26] Cho hệ thông tin $IS = (U, A)$ và hai thuộc tính $a, b \in A$. Entropy đồng thời của a và b là một đại lượng $H(a, b)$ xác định theo công thức sau:

$$H(a, b) = - \sum_{i=1}^m \sum_{j=1}^n P(a = x_i, b = y_j) \log_2 P(a = x_i, b = y_j). \quad (2.12)$$

Entropy $H(a, b)$ biểu thị mức độ không chắc chắn của hai thuộc tính a và b .

Định nghĩa 2.14. [26] Cho hệ thông tin $IS = (U, A)$ và hai thuộc tính $a, b \in A$. Entropy có điều kiện của a khi đã biết b là đại lượng $H(a|b)$ xác định bởi:

$$H(a|b) = - \sum_{j=1}^n P(b = y_j) \sum_{i=1}^m P(a = x_i | b = y_j) \log_2 P(a = x_i | b = y_j). \quad (2.13)$$

$H(a|b)$ xác định lượng entropy (tức là độ không chắc chắn) còn lại của thuộc tính a khi đã biết giá trị của một thuộc tính b . Áp dụng các công thức (2.11), (2.12) và (2.13) ta có:

$$H(a|b) = H(a, b) - H(b) \quad (2.14)$$

Định nghĩa 2.15. [26] Cho hệ thông tin $IS = (U, A)$ và hai thuộc tính $a, b \in A$. Thông tin tương hỗ giữa hai thuộc tính a và b được định nghĩa:

$$I(a; b) = H(a) - H(a|b) = H(b) - H(b|a) \quad (2.15)$$

Thông tin tin tương hỗ $I(a; b)$ là hàm không âm và đối xứng, tức là $I(a; b) \geq 0$ và $I(a; b) = I(b; a)$. $I(a; b)$ là lượng thông tin mà a và b chia sẻ cho nhau; nó cho biết thông tin về thuộc tính này sẽ làm giảm được bao nhiêu độ không chắc chắn của thuộc tính kia. Thông tin tin tương hỗ giữa a và b còn được gọi là thông tin có thêm được về a khi biết b .

Định nghĩa 2.16. [26, 27] Cho hệ thông tin $IS = (U, A)$ và hai thuộc tính $a, b \in A$. Biến thể thông tin chuẩn hóa $NVI(a, b)$ giữa a và b được xác định như sau:

$$NVI(a, b) = 1 - \frac{I(a; b)}{H(a, b)} = \frac{H(a|b) + H(b|a)}{H(a, b)}. \quad (2.16)$$

Định lý 2.1. [27] $NVI(a, b)$ là một metric trên không gian của các thuộc tính, nghĩa là đối với mọi $a, b, c \in A$, ta đều có:

- (i) $NVI(a, b) \geq 0$ và đẳng thức xảy ra khi và chỉ khi $a = b$,
- (ii) $NVI(a, b) = NVI(b, a)$,
- (iii) $NVI(a, b) + NVI(b, c) \geq NVI(a, c)$.

Để chứng minh NVI là một metric, trước hết ta chứng minh bất đẳng thức sau

$$H(a|b) \leq H(a|c) + H(c|b) \quad (2.17)$$

trong đó a, b và c là 3 thuộc tính bất kỳ.

Thật vậy, ta có $H(a|c) \leq H(a, c|b) = H(a|c, b) + H(c|b) \leq H(a|c) + H(c|b)$ (bất đẳng thức cuối cùng đúng vì khi có thêm điều kiện luôn làm giảm entropy).

Dễ thấy $NVI(a, b) \geq 0$, dấu bằng xảy ra khi $a = b$, và $NVI(a, b) = NVI(b, a)$. Do đó để chứng tỏ NVI là một metric, ta chỉ cần chứng minh NVI thỏa mãn bất đẳng thức tam giác, nghĩa là $NVI(a, b) \leq NVI(a, b) + NVI(c, a)$.

Sử dụng bất đẳng thức (2.17) và các phép tính đại số đơn giản ta có:

$$\begin{aligned} \frac{H(a|b)}{H(a, b)} &= \frac{H(a|b)}{H(b) + H(a|b)} \leq \frac{H(a|c) + H(c|b)}{H(b) + H(a|c) + H(c|b)} \\ &= \frac{H(a|c) + H(c|b)}{H(a|c) + H(b, c)} = \frac{H(a|c)}{H(a|c) + H(b, c)} + \frac{H(c|b)}{H(a|c) + H(b, c)} \\ &\leq \frac{H(a|c)}{H(a|c) + H(c)} + \frac{H(c|b)}{H(b, c)} = \frac{H(a|c)}{H(a, c)} + \frac{H(c|b)}{H(c, b)}. \end{aligned} \quad (2.18)$$

Hoán đổi a và b để thu được một bất đẳng thức tương tự khác:

$$\frac{H(b|a)}{H(b,a)} \leq \frac{H(b|c)}{H(b,c)} + \frac{H(c|a)}{H(c,a)} \quad (2.19)$$

Cộng (2.18) và (2.19) lại với nhau chúng ta được:

$$\begin{aligned} \frac{H(a|b)}{H(a,b)} + \frac{H(b|a)}{H(b,a)} &\leq \frac{H(a|c)}{H(a,c)} + \frac{H(c|b)}{H(c,b)} + \frac{H(b|c)}{H(b,c)} + \frac{H(c|a)}{H(c,a)} \\ \frac{H(a|b) + H(b|a)}{H(a,b)} &\leq \frac{H(b|c) + H(c|b)}{H(b,c)} + \frac{H(c|a) + H(a|c)}{H(c,a)} \end{aligned} \quad (2.20)$$

Vì

$$\frac{H(a|b) + H(b|a)}{H(a,b)} = 1 - \frac{I(a,b)}{H(a,b)} = NVI(a,b),$$

bất đẳng thức (2.20) có nghĩa là:

$$1 - \frac{I(a,b)}{H(a,b)} \leq \left(1 - \frac{I(b,c)}{H(b,c)}\right) + \left(1 - \frac{I(c,a)}{H(c,a)}\right).$$

$$NVI(a,b) \leq NVI(a,b) + NVI(c,a). \blacksquare$$

Giá trị của $NVI(a,b)$ nằm trong khoảng $[0,1]$. $NVI(a,b)$ cũng là một metric phổ quát theo nghĩa nếu một độ đo khoảng cách nào đó khác xác định a và b là gần nhau, thì NVI cũng sẽ đánh giá chúng gần nhau.

Mặc dù các độ đo entropy trên đây được định nghĩa cho các thuộc tính phân loại hoặc rời rạc, chúng cũng có thể được xác định cho các thuộc tính liên tục, nếu miền giá trị của các thuộc tính này được rời rạc hóa trước một cách thích hợp [27].

2.3 Một số thuật toán hiệu quả của lý thuyết tập thô

Phần này trình bày khái quát một số thuật toán hiệu quả trên các bảng dữ liệu lớn, đó là các thuật toán tìm lớp tương đương, tập xấp xỉ trên, tập xấp xỉ dưới và miền dương.

Thuật toán 2.1 Thuật toán xác định lớp tương đương

Đầu vào: Tập đối tượng U , tập thuộc tính B .

Đầu ra: Tập các lớp tương đương L trong U theo quan hệ $IND(B)$, (tức là phân hoạch của U theo $IND(B)$).

Thuật toán:

Bước 1: $L = \emptyset$

Bước 2: Nếu $U = \emptyset$

Thì: Thực hiện bước 5;

Ngược lại: Thực hiện bước 3.

Hết nếu.

Bước 3: Xét $x \in U$

$P = \{x\}$.

$U = U \setminus \{x\}$.

$\forall y \in U$:

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

Thì: $P = P \cup \{y\}$;

$U = U \setminus \{y\}$.

Hết nếu.

Hết với mọi.

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

Bước 4: Thực hiện bước 2.

Bước 5: Kết thúc.

Thuật toán 2.2 Thuật toán xác định xấp xỉ dưới

Đầu vào: Tập đối tượng U , tập thuộc tính B , tập các đối tượng X .

Đầu ra: Tập các đối tượng $\underline{B}X$.

Thuật toán:

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

Xác định phân hoạch P của tập vũ trụ U theo quan hệ $IND(B)$.

Bước 2: $U_1 = U$.

Nếu: $U_1 \neq \emptyset$

Thì: Thực hiện bước 3;

Ngược lại: Thực hiện bước 5.

Hết nếu.

Bước 3: Xét $x \in U_1$

Tìm lớp tương đương $P_i \in P$ sao cho: $x \in P_i$.

Nếu: $P_i \subseteq X$

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

Hết nếu.

$$U_1 = U_1 \setminus P_i.$$

Bước 4: Thực hiện bước 2.

Bước 5: Kết thúc.

Thuật toán 2.3 Thuật toán xác định xấp xỉ trên

Đầu vào: Tập đối tượng U , tập thuộc tính B , tập các đối tượng X .

Đầu ra: Tập các đối tượng $\bar{B}X$.

Thuật toán:

Bước 1: Khởi tạo $\bar{B}X = \emptyset$;

Xác định phân hoạch P của tập vũ trụ U theo quan hệ $IND(B)$ s.

Bước 2: $U_1 = U$

Nếu: $U_1 \neq \emptyset$

Thì: Thực hiện bước 3;

Ngược lại: Thực hiện bước 5.

Hết nếu.

Bước 3: Xét $x \in U_1$

Tìm lớp tương đương $P_i \in P$ sao cho: $x \in P_i$.

$$\bar{B}X = \bar{B}X \cup P_i.$$

Với mọi: $p \in P_i \cap U_1$.

$$U_1 = U_1 \setminus \{p\}.$$

Hết với mọi.

Bước 4: Thực hiện bước 2.

Bước 5: Kết thúc.

Thuật toán 2.4 Thuật toán xác định miền dương

Đầu vào: Hệ thông tin $S = (U, A, V, f)$, $A = C \cup D$.

Đầu ra: Tập các đối tượng $POS_C(D)$.

Thuật toán:

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

Bước 2: $POS_C(D) = \emptyset$.

Bước 3:

Với mọi: $j = 1, 2, \dots, m$

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

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

Hết nếu.

Hết với mọi.

Các thuật toán trên có độ phức tạp thời gian $O(kn \log n)$ và độ phức tạp không gian là $O(n)$, với n là số đối tượng của tập U , k là số thuộc tính của tập A .

2.4 Ứng dụng của lý thuyết tập thô trong khám phá tri thức từ cơ sở dữ liệu

Lý thuyết tập thô có thể được ứng dụng vào hầu hết các công đoạn của quá trình khám phá tri thức từ dữ liệu. Dưới đây là một số ứng dụng cụ thể của lý thuyết tập thô trong quá trình khám phá tri thức từ cơ sở dữ liệu [9, 10, 11, 13, 28].

(1) Tiền xử lý dữ liệu. Với giả thiết mô hình tối thiểu, lý thuyết tập thô được sử dụng để rút gọn và làm sạch dữ liệu cho các phân tích tiếp theo. Một cách cụ thể, đối với công đoạn tiền xử lý dữ liệu, lý thuyết tập thô là công cụ hữu hiệu giải quyết các vấn đề dưới đây [9, 10, 11].

- Xử lý các giá trị thiếu .
- Rời rạc hóa dữ liệu. Lý thuyết tập thô cho phép tạo ra các phép rời rạc hóa dữ liệu bảo toàn các lớp quyết định trong một bảng quyết định.
- Rút gọn dữ liệu.

Trong lý thuyết tập thô vấn đề lựa chọn thuộc tính trong khai phá dữ liệu được đưa về bài toán tìm tập thuộc tính rút gọn. Các công cụ sử dụng để tìm tập rút gọn là quan hệ không phân biệt giữa các cá thể và các thuật toán tìm tập rút gọn. Sử dụng các công cụ này người ta có thể tìm được tập các thuộc tính nhỏ nhất nhằm loại bỏ những thuộc tính dư thừa, không cần thiết cho nhiệm vụ khai phá; sau đó, dựa vào tập thuộc tính rút gọn này có thể tìm ra các quy luật chung hoặc các mẫu biểu diễn dữ liệu.

(2) Khai phá dữ liệu. Trong công đoạn khai phá dữ liệu, lý thuyết tập thô có thể được sử dụng giải quyết các vấn đề sau [9, 10, 11, 13, 28]:

- Phân lớp dữ liệu. Là mục đích đầu tiên lý thuyết tập thô hướng tới. Hiện nay, các công cụ tập thô có khả năng giải quyết bài toán phân lớp trong cả hai trường hợp, bảng thông tin nhất quán và không nhất quán.
- Gom cụm dữ liệu. Ngoài khả năng giải quyết hiệu quả bài toán phân lớp, gần đây một số nghiên cứu ứng dụng lý thuyết tập thô vào vấn đề gom cụm cũng đã được thực hiện
- Phát hiện luật kết hợp. Phép phân tích sự phụ thuộc giữa các thuộc tính trong lý thuyết tập thô có thể được sử dụng để phát hiện luật kết hợp, lượng hóa mức độ kết hợp giữa các tập thuộc tính.

Có thể nói lý thuyết tập thô là công cụ hữu hiệu cho quá trình khám phá tri thức từ cơ sở dữ liệu. Tuy vậy, các kết quả nghiên lý thuyết và ứng dụng đến nay vẫn còn những hạn chế. Những hạn chế nổi bật của lý thuyết tập thô kinh điển là [9, 10, 11, 13]:

- Dữ liệu khai phá phải là rời rạc, trong khi phần lớn các cơ sở dữ liệu thực tiễn thường chứa cả các thuộc tính liên tục.
- Dữ liệu khai phá phải đầy đủ, không bị nhiễu trong khi dữ liệu của phần lớn các cơ sở dữ liệu thực tiễn thường bị thiếu và/hoặc chứa nhiễu.
- Tri thức khám phá được dựa trên lý thuyết tập thô thường nhạy cảm với sự biến động của dữ liệu.
- Các thuật toán khai phá dữ liệu dựa vào lý thuyết tập thô thường có độ phức tạp cao.

Có thể thấy, lý thuyết tập thô đã được ứng dụng vào hầu hết các công đoạn của quá trình khám phá tri thức từ dữ liệu. Trong đó, rút gọn thuộc tính được xem là ứng dụng quan trọng nhất của lý thuyết tập thô trong khai phá dữ liệu. Mục tiêu của rút gọn thuộc tính là loại bỏ các thuộc tính dư thừa để tìm ra tập con các thuộc tính cốt yếu và cần thiết trong cơ sở dữ liệu. Đối với một bảng quyết định (tập dữ liệu dành cho bài toán phân lớp, có các thuộc tính điều kiện và thuộc tính quyết định), rút gọn thuộc tính là tìm tập con nhỏ nhất của tập thuộc tính điều kiện bảo toàn thông tin cho mục đích phân lớp các đối tượng như tập tất cả các thuộc tính điều kiện ban đầu. Các tập hợp con thuộc tính như vậy được gọi là các tập rút gọn. Nói chung, trong một bảng quyết định có thể tồn tại nhiều tập rút gọn. Trong những năm qua, nhiều phương pháp tính toán tập rút gọn đã được nghiên cứu và đề xuất trong cộng đồng các nhà nghiên cứu lý thuyết tập thô. Các phương pháp chính bao gồm: phương pháp sử dụng ma trận phân biệt, phương pháp dựa trên miền dương, phương pháp sử dụng các phép toán trong đại số quan hệ, phương pháp sử dụng entropy thông tin.

Bên cạnh đó, gom cụm dữ liệu cũng là một ứng dụng quan trọng trong lý thuyết tập thô trong khai phá dữ liệu. Trong những năm gần đây, gom cụm dữ liệu phân loại sử dụng tập thô đã thu hút nhiều sự chú ý từ cộng đồng nghiên cứu khai phá dữ liệu [29, 22, 30, 31, 24, 23]. Lý do là vì:

(1) Lý thuyết tập thô là công cụ phân tích hiệu quả dữ liệu phân loại;

(2) Lý thuyết tập thô cho phép xử lý sự không chắc chắn của dữ liệu. Mặc dù trong những năm qua, một số thuật toán gom cụm dữ liệu phân loại đã được đề xuất, nhưng chúng không được thiết kế để xử lý sự không chắc chắn trong quá trình gom cụm. Xử lý sự không chắc chắn trong quá trình gom cụm là một vấn đề quan trọng, bởi vì trong nhiều ứng dụng thực tế thường không có ranh giới rõ ràng giữa các cụm.

2.5 Kết luận chương 2

Nội dung chương 2 bao gồm 3 phần chính: khái quát về lý thuyết về tập thô với các khái niệm liên quan, quy trình khám phá tri thức từ cơ sở dữ liệu với các kỹ thuật khai phá dữ liệu cơ bản và ứng dụng của của lý thuyết về tập thô trong khai phá dữ liệu.

Các khái niệm cơ bản trình bày trong chương này là cơ sở để nghiên cứu đề xuất các phương pháp mới tìm tập rút gọn trong một bảng quyết định và gom cụm dữ liệu phân loại sử dụng tập thô, trình bày ở các chương sau.

CHƯƠNG 3. LỰA CHỌN THUỘC TÍNH SỬ DỤNG LÝ THUYẾT TẬP THÔ

3.1 Mở đầu

Như đã trình bày trong Chương 1, trong khai phá dữ liệu, các CSDL thực tế thường có kích thước rất lớn. Điều này làm cho quá trình khai phá dữ liệu gặp nhiều khó khăn, thậm chí là bất khả thi. Vấn đề đặt ra là trước khi thực hiện thuật toán khai thác dữ liệu cần phải có phương pháp rút gọn thuộc tính của cơ sở dữ liệu mà vẫn bảo toàn được những thông tin cần khai thác. Rút gọn thuộc tính có thể được thực hiện bằng cách sử dụng các kỹ thuật phù hợp, tùy thuộc vào yêu cầu của bài toán khai phá dữ liệu đặt ra. Những kỹ thuật này có thể được chia thành hai loại chính: biến đổi thuộc tính (attribute transformation) và lựa chọn thuộc tính (attribute selection) [1, 9, 10, 11]. Kỹ thuật biến đổi thuộc tính, hay còn gọi là trích xuất thuộc tính (attribute extraction), là việc tạo ra một số nhỏ hơn các thuộc tính mới bằng cách biến đổi các thuộc tính ban đầu sao cho các thuộc tính được tạo ra chứa thông tin hữu ích nhất cho mục tiêu khai phá. Ngược lại, kỹ thuật lựa chọn thuộc tính chỉ loại bỏ những thuộc tính không cần thiết hoặc không quan trọng và giữ nguyên các tính năng còn lại. Trong hai loại kỹ thuật rút gọn thuộc tính, kỹ thuật trích xuất thuộc tính là phức tạp hơn và cho kết quả khó giải thích cho người dùng. Tuy nhiên, thật khó có thể so sánh hiệu quả của hai phương pháp vì chúng được sử dụng trong những tình huống khác nhau.

Các nghiên cứu gần đây cho thấy, lý thuyết tập thô là một công cụ rất hiệu quả giải quyết nhiều vấn đề quan trọng trong khai phá dữ liệu, trong đó có bài toán lựa chọn thuộc tính. Lựa chọn thuộc tính là một phần quan trọng được nghiên cứu trong lý thuyết tập thô và được xem là ứng dụng quan trọng nhất của lý thuyết tập thô trong khai phá dữ liệu. Đối với một bảng quyết định, lựa chọn thuộc tính là việc tìm tập con nhỏ nhất của tập thuộc tính điều kiện, bảo toàn thông tin cho mục đích phân lớp các đối tượng như tập tất cả các thuộc tính điều kiện ban đầu. Các tập hợp con thuộc tính như vậy được gọi là các *tập rút gọn (reducts)* [8].

Trong những năm qua, nhiều phương pháp tính toán tập rút gọn một bảng quyết định đã được nghiên cứu đề xuất trong cộng đồng các nhà nghiên cứu lý thuyết tập thô. Các

phương pháp chính bao gồm: phương pháp sử dụng ma trận phân biệt, phương pháp dựa vào độ phụ thuộc, phương pháp sử dụng các phép toán trong đại số quan hệ, phương pháp sử dụng entropy thông tin.

Chương 3 này trình bày khái quát về vấn đề lựa chọn thuộc tính, các phương pháp chính tìm tập rút gọn của một bảng quyết định và đề xuất một thuật toán mới, với tên gọi ACBRC, dựa trên gom cụm các thuộc tính.

3.2 Khái quát về bài toán lựa chọn thuộc tính

Lựa chọn thuộc tính có thể được thực hiện bằng cách sử dụng các kỹ thuật phù hợp, tùy thuộc vào yêu cầu của bài toán khai phá dữ liệu đặt ra. Những kỹ thuật này có thể được chia thành hai loại chính, đó là biến đổi thuộc tính và lựa chọn thuộc tính [1, 32, 33].

Biến đổi thuộc tính là quá trình biến đổi không gian thuộc tính ban đầu thành không gian thuộc tính mới có số chiều thấp hơn. Với các kỹ thuật biến đổi thuộc tính, tập thuộc tính mới được tạo ra thường không mang ý nghĩa vật lý đối với người sử dụng và thường khó hiểu.

Lựa chọn thuộc tính là quá trình chọn ra một tập hợp con thuộc tính từ tập hợp các thuộc tính ban đầu, với mục tiêu loại bỏ càng nhiều càng tốt các thuộc tính không liên quan và dư thừa nhằm cải thiện chất lượng dữ liệu và giảm độ phức tạp về thời gian và không gian cho việc phân tích. Thật không may, việc tính toán tất cả các tập rút gọn hay tính toán một tập rút gọn tối ưu (theo nghĩa có số thuộc tính nhỏ nhất) là một bài toán NP- khó [3, 5]. Tuy nhiên, trong thực hành thường không yêu cầu tìm tất cả các tập rút gọn mà chỉ cần tìm được một tập rút gọn tốt nhất theo một tiêu chuẩn đánh giá nào đó là đủ. Do đó, nhiều thuật toán heuristic tìm kiếm một tập rút gọn xấp xỉ đã được nghiên cứu và đề xuất [1, 3, 4, 5]. Các thuật toán này giảm thiểu đáng kể khối lượng tính toán, nhờ đó có thể áp dụng đối với các bài toán có khối lượng dữ liệu lớn. Nội dung dưới đây trình bày khái quát về các kỹ thuật này.

Nhìn chung, một thuật toán lựa chọn thuộc tính thường bao gồm bốn bước cơ bản sau [1, 32, 33]:

- (1) Tạo lập tập con để đánh giá.

(2) Đánh giá tập con.

(3) Kiểm tra điều kiện dừng.

(4) Kiểm chứng kết quả.

Hiện nay có hai cách tiếp cận chính đối với bài toán lựa chọn thuộc tính bao gồm tiếp cận lọc (filter) và đóng gói (wrapper) [1, 3, 32]. Mỗi cách tiếp cận có những chú trọng riêng dành cho việc rút gọn kích thước dữ liệu hay để nâng cao độ chính xác.

Với cách tiếp cận filter, các thuộc tính được chọn chỉ dựa trên độ quan trọng của chúng trong việc mô tả dữ liệu, gọi là độ quan trọng của thuộc tính. Cho đến nay, Nhiều phương pháp nhiều đánh giá độ quan trọng của của các thuộc tính đã được đề xuất.

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

Cách tiếp cận filter có ưu điểm là thời gian tính toán nhanh, nhưng do không sử dụng thông tin nhãn lớp (học không có giám sát) của các bộ dữ liệu nên kết quả thường có độ chính xác không cao. Gần đây, nhiều nhà nghiên cứu đã đề xuất một số cách tiếp cận lựa chọn thuộc tính mới, chẳng hạn cách tiếp cận lai ghép (hybrid approach) nhằm kết hợp các ưu điểm của cả hai cách tiếp cận filter và wrapper [33].

Cũng có thể phân chia các cách tiếp cận bài toán lựa chọn thuộc tính thành hai loại: có giám sát (supervised) và không có giám sát (unsupervised), tùy theo việc lựa chọn có sử dụng hay không sử dụng thông tin nhãn lớp của các đối tượng.

Quy trình tạo lập các tập con là vấn đề quan trọng trong quá trình lựa chọn thuộc tính. Tạo lập tập con thuộc tính là quá trình tìm kiếm liên tiếp nhằm tạo ra các tập con để tiến hành đánh giá và lựa chọn. Quy trình này bao gồm việc chọn điểm xuất phát, chọn hướng tìm kiếm và chiến lược tìm kiếm tập con. Giả sử có n thuộc tính trong tập dữ liệu ban đầu, khi đó số tất cả các tập con khác rỗng từ n thuộc tính sẽ là $2^n - 1$. Có thể thấy, việc tìm tập con tối ưu theo một tiêu chuẩn nào đó, ngay cả khi n 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 [32, 33]:

- Đã thu được số thuộc tính quy định;
- Số bước lặp quy định cho quá trình lựa chọn đã hết;
- Việc thêm vào hay loại bớt một thuộc tính nào đó không cho một tập con tốt hơn;
- 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.

Thông thường có hai phương pháp tạo lập các tập con cho việc chọn lựa thuộc tính, bao gồm [32, 33]: phương pháp bổ sung dần (Forward Generation) và phương pháp loại bỏ dần (Backward Generation).

Tạo lập theo phương pháp bổ sung dần bắt đầu bằng tập rỗng. Sau đó, tại mỗi bước lặp một thuộc tính tốt nhất (theo tiêu chuẩn đánh giá) trong số các thuộc tính còn lại sẽ được thêm vào. Quá trình tạo lập dừng lại khi đã vét cạn tất cả các thuộc tính của tập dữ liệu ban đầu hoặc đã tìm được tập con tối ưu.

Ngược lại với phương pháp bổ sung dần, phương pháp loại bỏ dần bắt đầu bằng tập tất cả các thuộc tính. Tại mỗi bước lặp, một thuộc tính tồi nhất (theo tiêu chuẩn đánh giá) sẽ bị loại. Tập thuộc tính ban đầu sẽ nhỏ dần cho đến khi chỉ còn lại một thuộc tính hoặc khi điều kiện dừng thỏa mãn.

Một phương pháp khác để tạo lập các tập con là bắt đầu bằng một tập con thuộc tính chọn ngẫu nhiên, sau đó tại mỗi bước lặp lần lượt thêm vào hoặc loại bớt một thuộc tính cũng được chọn một cách ngẫu nhiên.

Một vấn đề quan trọng khác trong lựa chọn thuộc tính là xác định cách thức đánh mức độ phù hợp của mỗi tập con. Để đánh giá một tập con thuộc tính được chọn là tối ưu phải

dựa trên một tiêu chuẩn đánh giá nhất định, một tập con là tối ưu theo tiêu chuẩn này chưa chắc sẽ tối ưu theo tiêu chuẩn khác. Các tiêu chuẩn đánh giá có thể phân thành hai loại: tiêu chuẩn độc lập và tiêu chuẩn phụ thuộc [32, 33].

Tiêu chuẩn độc lập (thường được dùng trong cách tiếp cận filter) đánh giá mức độ phù hợp của một hay một tập con thuộc tính một cách độc lập, không thông qua áp dụng một thuật học. Các tiêu chuẩn độc lập thường được sử dụng để đánh giá các tập con thuộc tính để lựa chọn là: số đo khoảng cách, số đo lượng thông tin thu thêm, số đo độ phụ thuộc, số đo độ nhất quán và số đo độ tương tự.

Tiêu chuẩn phụ thuộc (thường được dùng trong cách tiếp cận wrapper) đánh giá một tập con thuộc tính thông qua độ hiệu quả của một thuật học áp dụng trên chính tập thuộc tính cần đánh giá. Trong học có giám sát, mục đích đầu tiên là cực tiểu hóa sai số dự báo. Do đó, sai số dự báo (hay độ chính xác của dự báo) thường được chọn làm tiêu chuẩn để đánh giá các tập con thuộc tính. Kết quả tập con thuộc tính được chọn dựa trên tiêu chuẩn này có khả năng dự báo cao tuy nhiên điểm hạn chế là nó sẽ mất nhiều thời gian tính toán.

3.3 Các phương pháp lựa chọn thuộc tính sử dụng lý thuyết tập thô

Trong cộng đồng tập thô, các thuật toán lựa chọn thuộc tính được thực hiện bằng việc tìm kiếm các rút gọn (reducts) của tập các thuộc tính, nghĩa là tìm cách rút gọn tối đa tập các thuộc tính ban đầu mà vẫn đảm bảo được những thông tin cần thiết đối với nhiệm vụ khai phá dữ liệu. Thật không may, việc tìm kiếm tất cả các tập rút gọn là không thể thực hiện được trong hầu hết các trường hợp vì với tập dữ liệu có n thuộc tính sẽ có $2^n - 1$ tập hợp con, khi n tăng số tập con thuộc tính sẽ tăng theo cấp số nhân. Tìm kiếm tất cả các tập rút gọn chỉ có thể được khi n tương đối nhỏ.

Tuy nhiên, trong ứng dụng thực tiễn thường không đòi hỏi tìm tất cả các tập rút gọn mà chỉ cần tìm một tập rút gọn tốt nhất theo một nghĩa nào đó là đủ. Vì vậy, trong những năm qua nhiều thuật toán heuristic tìm một tập rút gọn xấp xỉ đã được các nhà nghiên cứu đề xuất. Các thuật toán này nhằm giảm khối lượng tính toán, nhờ đó có thể áp dụng đối với các tập dữ liệu lớn. Với cách tiếp cận này, các khái niệm của lý thuyết tập thô được sử dụng để xác một tiêu chuẩn đánh giá mức độ cần thiết hay quan trọng của các thuộc tính, sau đó

chuẩn đánh giá này được sử dụng như là các hàm heuristic định hướng cho quá trình lựa chọn thuộc tính trong các thuật toán.

Các phương pháp heuristic thường áp dụng một trong hai chiến lược cơ bản tìm kiếm tập rút gọn, đó là bổ sung dần và loại bỏ dần [2, 9, 13, 10]. Chiến lược bổ sung dần bắt đầu với tập rỗng hoặc tập lõi Core và liên tục bổ sung thêm một thuộc tính tại mỗi thời điểm cho đến khi có được một tập rút gọn, hoặc một tập cha của một tập rút gọn. Chiến lược loại bỏ dần bắt đầu với tập hợp đầy đủ các thuộc tính và liên tục xóa đi một thuộc tính tại mỗi thời điểm cho đến khi có được một tập rút gọn. Từ tính chất của tập rút gọn, có thể thấy các thuật toán áp dụng chiến lược loại bỏ dần luôn dẫn đến một tập rút gọn.

Mục này trước hết trình bày thuật toán kinh điển tìm tất cả các tập rút gọn sử dụng ma trận không phân biệt, sau đó là một số thuật toán heuristic tìm tập rút gọn xấp xỉ của bảng quyết định bao gồm: phương pháp dựa trên hàm đo độ phụ thuộc, phương pháp sử dụng các phép toán trong đại số quan hệ, phương pháp sử dụng entropy thông tin. Các thuật toán heuristic có độ phức tạp tính toán theo thời gian là đa thức, và do đó có thể áp dụng được trên bảng dữ liệu với kích thước lớn.

3.3.1 Phương pháp lựa chọn thuộc tính sử dụng ma trận phân biệt

Phương pháp lựa chọn thuộc tính sử dụng ma trận phân biệt là phương pháp nhằm xác định tất cả các tập rút gọn trong một bảng quyết định có số thuộc tính tương đối nhỏ.

Cho bảng quyết định $DT = (U, C \cup \{d\})$ với tập các đối tượng $U = \{u_1, u_2, \dots, u_n\}$, tập các thuộc tính điều kiện $C = \{c_1, c_2, \dots, c_m\}$ và thuộc tính điều kiện d . Để tìm tất cả các tập rút gọn của một bảng quyết định, trong [28] Skowron đã đề xuất thuật toán sử dụng khái niệm ma trận phân biệt và hàm phân biệt định nghĩa dưới đây.

Định nghĩa 3.1. [28] *Ma trận phân biệt của DT là ma trận $M(DT)$ cỡ $n \times n$ với các phần tử m_{ij} xác định:*

$$m_{ij} = \begin{cases} \{c \in C | c(u_i) \neq c(u_j) \text{ khi } d(u_i) \neq d(u_j)\} \\ \emptyset & \text{khi } d(u_i) = d(u_j) \end{cases} \quad (3.1)$$

Có thể thấy, $M(DT)$ là một ma trận đối xứng. Mỗi phần tử m_{ij} của $M(DT)$ hoặc là một tập bao gồm các thuộc tính điều kiện cho phép phân biệt hai đối tượng u_i, u_j nếu $d(u_i) \neq d(u_j)$ hoặc là một tập rỗng nếu $d(u_i) = d(u_j)$.

Hàm phân biệt của DT là hàm f_{DT} của các biến Bool $c_1^*, c_2^*, \dots, c_m^*$ cho tương ứng với các biến điều kiện $c_1, c_2, \dots, c_m$ trong DT và được xây dựng như sau [28]:

- Với mỗi $u_i \in U$, lập biểu thức $f_{DT}(u_i) = \bigwedge_{i \neq j} \bigvee m_{ij}$, trong đó

- $\bigvee m_{ij}$ là biểu thức tuyển của tất cả các biến Bool ứng với các biến $c \in m_{ij}$, nếu $m_{ij} \neq \emptyset$.
- $\bigvee m_{ij} = true$ nếu $m_{ij} = \emptyset$ và $d(u_i) = d(u_j)$.
- $\bigvee m_{ij} = false$ nếu $m_{ij} = \emptyset$ và $d(u_i) \neq d(u_j)$. (Trường hợp thứ ba này xảy

ra đối với bảng quyết định không nhất quán)

- Khi đã có các biểu thức $f_{DT}(u_i)$, tính

$$f_{DT} = \bigwedge_{1 \leq i \leq n} f_{DT}(u_i) \quad (3.2)$$

Để đơn giản hóa ký hiệu, người ta thường bỏ đi các dấu * chỉ các biến Bool.

Mệnh đề 3.1 [28] Cho bảng quyết định $DT = (U, C \cup \{d\})$ với n đối tượng và m thuộc tính và có ma trận phân biệt $M(DT) = (m_{ij})$. Khi đó:

1) $Core(DT) = \{c \in C : m_{ij} = \{c\} \text{ với cặp } i, j \text{ nào đó}\}.$

2) Tập tất cả các tiền đề nguyên tố của hàm phân biệt f_{DT} chính là tập tất cả các tập rút gọn của bảng quyết định DT .

Trên cơ sở khẳng định 2) của mệnh đề 3.1, quá trình tìm tất cả các tập rút gọn của một bảng quyết định bao gồm hai bước sau:

Bước 1:

- Thiết lập ma trận phân biệt $M(DT) = (m_{ij})$ theo công thức (3.1);

- Xây dựng hàm phân biệt theo công thức (3.2);

Bước 2:

- Sử dụng luật hấp thụ của đại số Bool biến đổi hàm phân biệt từ dạng hội chuẩn tắc (Conjunctive Normal Form – CNF) về dạng tuyển chuẩn tắc (Disjunctive Normal Form – DNF).

- Xác định các tập rút gọn: Mỗi đơn thức trong dạng tuyển chuẩn tắc thu được sẽ là một tiền đề nguyên tố của f_{DT} và do đó là một tập rút gọn của bảng quyết định DT .

Khi m cố định, bài toán tìm $Core(DT)$ có độ phức tạp thời gian là $O(n^2)$ [28]. Vì vấn đề tìm tất cả các tiền đề nguyên tố của f_{DT} là bài toán có độ phức tạp là NP-khó, suy ra bài toán tìm tất cả các tập rút gọn cũng như bài toán tìm tập rút gọn có kích thước nhỏ nhất của một bảng quyết định là các bài toán có độ phức tạp là NP-khó [28].

Ví dụ 3.1. Xét bảng quyết định như trong Bảng 3.1.

Bảng 3.1 Bảng quyết định ví dụ 3.1.

U	a	b	c	d
u_1	1	1	0	0
u_2	1	1	1	1
u_3	1	1	2	1
u_4	0	1	0	0
u_5	0	0	1	0
u_6	0	1	2	1

Ta có ma trận phân biệt của bảng này là Bảng 3.2. Chú ý rằng, đây là ma trận đối xứng nên chúng ta chỉ trình bày ma trận tam giác dưới.

Bảng 3.2 Ma trận phân biệt của Bảng quyết định 3.1.

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	$\emptyset$					
u_2	$\{c\}$	$\emptyset$				
u_3	$\{c\}$	$\emptyset$	$\emptyset$			
u_4	$\emptyset$	$\{a, c\}$	$\{a, c\}$	$\emptyset$		
u_5	$\emptyset$	$\{a, b\}$	$\{a, b, c\}$	$\emptyset$	$\emptyset$	
u_6	$\{a, c\}$	$\emptyset$	$\emptyset$	$\{c\}$	$\{b, c\}$	$\emptyset$

Từ ma trận phân biệt Bảng 3.2, ta có:

$$\forall c_{12} = c; \forall c_{13} = c; \forall c_{14} = true; \forall c_{15} = true; \forall c_{16} = a \vee c.$$

$$\text{Vì vậy, } fs(u_1) = c \wedge c \wedge true \wedge true \wedge (a \vee c) = c.$$

Tính toán tương tự ta có

$$fs(u_2) = c \wedge (a \vee b);$$

$$fs(u_2) = c;$$

$$fs(u_3) = c;$$

$$fs(u_4) = c;$$

$$fs(u_5) = (a \vee b) \wedge (b \vee c);$$

$$fs(u_6) = c;$$

Vậy,

$$f_s = \bigwedge_{1 \leq i \leq 6} fs(u_i) = c \wedge (c \wedge (a \vee b)) \wedge c \wedge c \wedge c \wedge c = c \wedge (a \vee b) = (a \wedge c) \vee (b \wedge c)$$

Suy ra hai tập rút gọn của bảng quyết định đã cho là: $R_1 = \{a, c\}$ và $R_2 = \{b, c\}$. Tập lỗi là $\{c\}$.

Có thể thấy việc tìm tất cả các tập rút gọn bằng phương pháp sử dụng ma trận phân biệt trên đây là rất tốn kém, không áp dụng được đối với các tập dữ liệu lớn.

3.3.2 Phương pháp rút gọn thuộc tính dựa vào độ phụ thuộc

Phương pháp dựa vào độ phụ thuộc sử dụng hàm đo độ phụ thuộc định nghĩa bởi Pawlak làm tiêu chuẩn đánh giá mức ý nghĩa của các thuộc tính. Nó còn được gọi là phương pháp dựa vào vùng dương, bởi vì giá trị độ phụ thuộc này hoàn toàn được xác định bởi vùng dương như định nghĩa sau đây.

Định nghĩa 3.2. [8, 10] Cho bảng quyết định $DT = (U, C \cup \{d\})$, Với tập con $B \subseteq C$, độ phụ thuộc $\gamma_B(d)$ của d vào B được định nghĩa như sau:

$$\gamma_B(d) = \frac{|POS_B(d)|}{|U|} \quad (3.3)$$

trong đó $POS_B(d)$ là miền B -dương của d được xác định theo công thức 2.6 đã được trình bày trong chương 2.

Rõ ràng, $0 \leq \gamma_B(d) \leq 1$. Nếu $\gamma_B(d) = 1$, thì ta nói rằng d phụ thuộc hoàn toàn vào B , còn nếu $0 < \gamma_B(d) < 1$, thì d phụ thuộc vào B với mức độ $\gamma_B(d)$. Khi $\gamma_B(d) = 0$, ta nói rằng d không phụ thuộc vào B .

Mệnh đề 3.2. [30] Cho bảng quyết định $DT = (U, C \cup \{d\})$. Khi đó:

- 1) Nếu $DT = (U, C \cup \{d\})$ là nhất quán thì $POS_C(d) = U$
- 2) $\gamma_B(d) \leq \gamma_{B'}(d)$ với mọi B, B' thỏa mãn $B \subset B' \subseteq C$.

Thuật toán QuickReduct [34, 35] thực hiện tính toán một tập rút gọn mà không cần phải sinh ra tất cả các tập con có thể (tập con ứng viên). Thực hiện chiến lược tìm kiếm bổ sung dần, nó bắt đầu với một tập rỗng và lần lượt thêm vào, tại mỗi thời điểm, một thuộc tính làm gia tăng nhiều nhất số đo độ phụ thuộc, cho đến khi đạt được giá trị độ phụ thuộc tối đa có thể đối với tập dữ liệu.

Định nghĩa 3.3. [34, 35] Cho bảng quyết định $DT = (U, C \cup \{d\})$, $B \subseteq C$. Với mỗi thuộc tính $a \in B$, mức ý nghĩa của a đối với B và d trong DT được đo bằng:

$$SIG(a, B, d) = \gamma_B(d) - \gamma_{B-\{a\}}(d). \quad (3.4)$$

$SIG(a, B, d)$ là số đo mức biến động giá trị của hàm đo độ phụ thuộc $\gamma_B(d)$ khi loại bỏ thuộc tính a khỏi tập B .

Thuật toán QuickReduct [34, 35] dưới đây là thuật toán tiêu biểu sử dụng độ phụ thuộc theo nghĩa Pawlak (Định nghĩa 3.2) và chiến thuật tìm kiếm bổ sung dần tìm tập rút gọn xấp xỉ.

Thuật toán 3.1 Thuật toán QuickReduct

Đầu vào: Bảng quyết định $DT = (U, C \cup \{d\})$

Đầu ra: Tập rút gọn R

Thuật toán:

(1) Tính độ phụ thuộc Pawlak: $\gamma_C(d)$

(2) $R \leftarrow \emptyset$

(3) **Thực hiện**

(4) $T \leftarrow R$

(5) $\forall x \in (C - R)$

(6) **Nếu** $\gamma_{R \cup \{x\}}(d) > \gamma_T(d)$

(7) $T \leftarrow R \cup \{x\}$

(8) $R \leftarrow T$

(9) **Cho đến khi** $\gamma_R(d) = \gamma_C(d)$

(10) **Trả về** R

Ví dụ 3.2. Xét bảng quyết định cho trong Bảng 3.3. Với $C = \{c1, c2, c3, c4\}$ là tập các thuộc tính điều kiện và d là thuộc tính quyết định. Đây là một bảng quyết định nhất quán.

Bảng 3.3 Bảng quyết định

U	a	b	c	d	e
0	0	1	2	2	1
1	1	0	0	0	2
2	2	1	1	0	0
3	0	0	1	2	2
4	0	1	2	1	0
5	2	2	1	0	0
6	2	0	0	0	2
7	1	0	0	1	0

Theo thuật toán, độ phụ thuộc Pawlak của từng thuộc tính sẽ được tính toán như sau:

$$\gamma_{\{c1,c2,c3,c4\}}(d) = 1;$$

$$\gamma_{\{c1\}}(d) = \frac{0}{8}; \quad \gamma_{\{c2\}}(d) = \frac{1}{8}; \quad \gamma_{\{c3\}}(d) = \frac{0}{8}; \quad \gamma_{\{d\}}(d) = \frac{2}{8}.$$

Thuộc tính $c4$ được chọn vì có độ phụ thuộc cao nhất. Quá trình này tiếp tục lặp lại cho đến khi độ phụ thuộc của tập rút gọn bằng độ phụ thuộc của tập tất cả các thuộc tính điều kiện vì bảng quyết định là nhất quán.

Như vậy, các tập $\{c1, c4\}$, $\{c2, c4\}$ và $\{c3, c4\}$ tiếp tục được đưa ra đánh giá. Ta có:

$$\gamma_{\{c1,c4\}}(d) = \frac{3}{8}; \quad \gamma_{\{c2,c4\}}(d) = \frac{8}{8} = 1.$$

Với $\gamma_{\{c2,c4\}}(d) = 1 = \gamma_{\{c1,c2,c3,c4\}}(d)$, thuật toán sẽ kết thúc và tập con $\{c2, c4\}$ là tập rút gọn được chọn.

3.3.3 Phương pháp rút gọn thuộc tính sử dụng sử dụng độ phụ thuộc tương đối

Nhằm tránh phải tính toán hàm phân biệt hay miền khẳng định, một công việc tiêu tốn nhiều thời gian mà lại không tối ưu, trong [36], Han và các cộng sự đã thay thế số đo độ

phụ thuộc trong lý thuyết tập thô truyền thống bằng một số đo khác, gọi là độ phụ thuộc tương đối, dựa vào đại số quan hệ.

Định nghĩa 3.4. [36] Cho bảng quyết định $DT = (U, C \cup \{d\})$ và $B \subseteq C \cup \{d\}$. Phép chiếu của U lên B , ký hiệu là $\Pi_B(U)$, cho kết quả là một tập con của U thu được bằng cách:

- 1) Loại bỏ tất cả các thuộc tính không thuộc B ;
- 2) Hợp nhất tất cả các đối tượng không phân biệt theo B thành một.

Định nghĩa 3.5. [36] Cho bảng quyết định nhất quán $DT = (U, C \cup \{d\})$ và $B \subseteq C \cup \{d\}$, độ phụ thuộc tương đối của d vào R là đại lượng $k_B(d)$ xác định bởi:

$$k_B(d) = \frac{\text{card}(U/\text{IND}(B))}{\text{card}(U/\text{IND}(B \cup \{d\}))} = \frac{\text{card}(\Pi_B(U))}{\text{card}(\Pi_{B \cup \{d\}}(U))} \quad (3.5)$$

trong đó $\text{card}(\cdot)$ chỉ số phần tử của một tập hợp, $\Pi_X(U)$ chỉ phép chiếu DT lên tập thuộc tính X . Tính hợp lý của định nghĩa này được thể hiện qua mệnh đề dưới đây:

Mệnh đề 3.3. [36, 37] Cho bảng quyết định $DT = (U, C \cup \{d\})$. Với mọi tập $B \subseteq C$, ta có:

- a) $k_B(d) \leq 1$
- b) $k_B(d) = 1 \Leftrightarrow$ phụ thuộc hàm $B \rightarrow d$ đúng trên U .
- c) DT nhất quán $\Leftrightarrow k_C(d) = 1$.
- d) Nếu $k_B(d) = 1$ thì $k_{B'}(d) = 1$ với mọi $B \subseteq B' \subseteq C$.

Khi phân hoạch U theo $R \subseteq C$ trùng với phân hoạch U theo $R \cup \{d\}$ thì tập R có thể được lựa chọn thay cho tập điều kiện C . Vì vậy ta đi đến định nghĩa về tập rút gọn như sau:

Định nghĩa 3.6. [36, 37] Cho bảng quyết định $DT = (U, C \cup \{d\})$. Tập con các thuộc tính điều kiện $R \subseteq C$ được coi là một tập rút gọn của tập thuộc tính điều kiện C trong bảng quyết định DT nếu :

$$k_R(d) = k_C(d) \text{ và } k_{R'}(d) < k_C(d), \quad \forall R' \subset R$$

Sử dụng độ đo phụ thuộc tương đối trên đây làm tiêu chuẩn đo mức ý nghĩa của các thuộc tính, Han và các cộng sự đã đề xuất một thuật toán tính toán tập rút gọn xấp xỉ trong bảng quyết định nhất quán. Thuật toán thực hiện loại bỏ dần các thuộc tính khỏi tập các thuộc tính C theo cách một thuộc tính sẽ được loại bỏ nếu sau khi loại bỏ nó độ phụ thuộc tương đối thu được vẫn có giá trị bằng 1.

Xuất phát từ $R = C$, thuật toán thực hiện tối đa k vòng lặp để thực hiện loại bỏ $R \leftarrow R - \{c\}$. Trong mỗi vòng lặp này, chúng ta phải kiểm tra điều kiện $k_{R-\{c\}}(d) = 1$. Điều đó đòi hỏi sắp xếp lại các đối tượng trong U theo thứ tự tăng/giảm giá trị của các thuộc tính trong $R \setminus \{c_j\}$ và d . Vậy, độ phức tạp tính toán của thuật toán này sẽ là $O(kn \log n)$ [36, 37].

Thuật toán 3.2 Thuật toán RelativeReduct

Đầu vào: Bảng quyết định $DT = (U, C \cup \{d\})$

Đầu ra: Tập rút gọn R .

Thuật toán:

Bước 1: $R \leftarrow C$;

Bước 2: Với mọi $c \in C$

Nếu: $k_{R-\{c\}}(d) == 1$

Thì: $R \leftarrow R - \{c\}$;

Hết nếu.

Hết với mọi

Bước 3: Kết thúc.

Ví dụ 3.3. Xét lại bảng quyết định cho trong Bảng 3.3 với tập các thuộc tính điều kiện $C = \{c1, c2, c3, c4\}$ và thuộc tính quyết định là d .

Thuật toán bắt đầu với tập ứng viên là tập R bao gồm tất cả các thuộc tính điều kiện, $\{c1, c2, c3, c4\}$. Sau đó, thuật toán xem xét việc loại bỏ thuộc tính đầu tiên là thuộc tính $c1$:

$$\begin{aligned}
k_{\{c2, c3, c4\}}(d) &= \frac{\text{card}(U/\text{IND}(\{c2, c3, c4\}))}{\text{card}(U/\text{IND}(\{c2, c3, c4, d\}))} \\
&= \frac{\text{card}(\{\{0\}, \{1,6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}\})}{\text{card}(\{\{0\}, \{1,6\}, \{2\}, \{3\}, \{4\}, \{5\}, \{7\}\})} = 1.
\end{aligned}$$

Vì độ phụ thuộc tương đối bằng 1, thuộc tính $c1$ có thể được loại bỏ khỏi tập rút gọn ứng viên hiện tại, $R \leftarrow \{c2, c3, c4\}$.

Thuật toán tiếp tục xem xét việc loại bỏ thuộc tính $c2$ khỏi R :

$$\begin{aligned}
k_{\{c3, c4\}}(d) &= \frac{\text{card}(U/\text{IND}(\{c3, c4\}))}{\text{card}(U/\text{IND}(\{c3, c4, d\}))} \\
&= \frac{\text{card}(\{\{0\}, \{1,6\}, \{2,5\}, \{3\}, \{4\}, \{7\}\})}{\text{card}(\{\{0\}, \{1,6\}, \{2,5\}, \{3\}, \{4\}, \{7\}\})} = 1.
\end{aligned}$$

Một lần nữa, độ phụ thuộc tương đối của d vào $\{c3, c4\}$ vẫn bằng 1, do đó $c2$ được loại bỏ khỏi R , $R \leftarrow \{c3, c4\}$. Bước tiếp theo xem xét việc loại bỏ $c3$ khỏi tập rút gọn ứng viên:

$$k_{\{c3\}}(d) = \frac{\text{card}(U/\text{IND}(\{c3\}))}{\text{card}(U/\text{IND}(\{c3, d\}))} = \frac{\text{card}(\{\{0,3\}, \{1,2,5,6\}, \{4,7\}\})}{\text{card}(\{\{0\}, \{3\}, \{1,6\}, \{2,5\}, \{4,7\}\})} = \frac{3}{5}.$$

Vì giá trị độ phụ thuộc tương đối nhỏ hơn 1, thuộc tính $c3$ không bị loại bỏ khỏi R . Thuật toán tiếp tục xem xét việc loại bỏ thuộc tính $c4$ khỏi $R = \{c3, c4\}$:

$$k_{\{c4\}}(d) = \frac{\text{card}(U/\text{IND}(\{c4\}))}{\text{card}(U/\text{IND}(\{c4, d\}))} = \frac{\text{card}(\{\{0,3\}, \{1,2,5,6\}, \{4,7\}\})}{\text{card}(\{\{0\}, \{3\}, \{1,6\}, \{2,5\}, \{4,7\}\})} = \frac{3}{5}.$$

Vì độ phụ thuộc tương đối không bằng 1; thuộc tính $c4$ được giữ lại trong tập rút gọn ứng viên. Do không còn các thuộc tính phải xem xét tiếp, thuật toán kết thúc và cho ra kết quả là tập rút gọn $\{c3, c4\}$.

3.3.4 Phương pháp rút gọn thuộc tính sử dụng Entropy thông tin

Các công trình nghiên cứu về lý thuyết tập thô hiện nay tập trung vào hai hướng chính; một là xây dựng mô hình lý thuyết tập thô mở rộng dựa trên lý thuyết tập thô của Pawlak

nhằm giải quyết các bài toán trong thực tế; hai là tích hợp lý thuyết tập thô với các lý thuyết khác như lý thuyết thông tin, lý thuyết tập mờ, lý thuyết xác suất thống kê, v.v.

Như đã trình bày ở 2.2.4, Entropy trong lý thuyết thông tin là một đại lượng toán học dùng để đo độ không chắc chắn của một đại lượng ngẫu nhiên. Entropy thông tin được xem là công cụ hiệu quả để giải quyết các bài toán trong lý thuyết tập thô, đặc biệt là bài toán rút gọn thuộc tính trên các hệ thông tin không nhất quán.

Định nghĩa 3.7. [35] Cho bảng quyết định $DT = (U, C \cup \{d\})$, thuộc tính $a \in C$ gọi là dư thừa dựa trên Entropy nếu $H(D|C) = H(D|C - \{a\})$.

Trong mọi trường hợp, thuộc tính a dư thừa dựa trên Entropy thì a cũng dư thừa dựa trên miền dương của Pawlak. Tuy nhiên nếu DT không nhất quán, mệnh đề ngược lại không đúng. Trong trường hợp DT nhất quán, hai khái niệm dư thừa này là như nhau, nghĩa là $\gamma_{C-\{a\}}(d) = \gamma_C(d)$ khi và chỉ khi $H(d|C) = H(d|C - \{a\})$ [35].

Định nghĩa 3.8. [35] Cho bảng quyết định $DT = (U, C \cup \{d\})$, tập thuộc tính $R \subseteq C$ gọi là một rút gọn của DT dựa trên Entropy nếu thỏa mãn hai điều kiện sau:

- (1) $H(D|C) = H(D|R)$
- (2) $\forall r \in R, H(D|R) \neq H(D|R - \{r\})$

Dựa trên entropy Shannon có điều kiện, Wang và cộng sự [35] đã đưa ra khái niệm tập rút gọn và tập lõi của bảng quyết định và đề xuất thuật toán heuristic tìm tập rút gọn của bảng quyết định: thuật toán CEBARKNC. CEBARKNC là thuật toán heuristic với độ phức tạp $O(|C|^2|U| + |C||U|^3)$. Cấu trúc của thuật toán CEBARKNC tương tự như thuật toán QuickReduct ngoại trừ việc sử dụng độ đo mức ý nghĩa thuộc tính dựa trên entropy có điều kiện và được minh họa ở thuật toán 3.3.

Thuật toán 3.3 Thuật toán CEBARKNC

Đầu vào: Bảng quyết định $DT = (U, C \cup \{D\})$

Đầu ra: Tập rút gọn R

Thuật toán:

(1) *Tính độ phụ thuộc dựa trên entropy có điều kiện: $H(D|C)$*

(2) $R \leftarrow C$

(3) **Thực hiện**

(5) Với mọi $x \in C$

(6) Nếu $H(D|R - \{x\}) = H(D|C)$

(7) $R \leftarrow R - \{x\}$

(9) Hết với mọi

(10) **Trả về** R

3.3.5 Phương pháp lựa chọn thuộc tính dựa trên gom cụm

Các thuật toán heuristic lựa chọn thuộc tính được trình bày trong mục trên đây là các thuật toán tìm tập rút gọn xấp xỉ với độ phức tạp thời gian hợp lý. Có thể thấy, các thuật toán này có thể loại bỏ thành công các đặc trưng không liên quan nhưng có thể không thể loại bỏ các thuộc tính dư thừa [17, 18, 19, 20, 21]. Để khắc phục vấn đề này, một số phương pháp lựa chọn thuộc tính sử dụng gom cụm đã được đề xuất [19, 20, 21, 38, 39, 40]. Lưu ý rằng gom cụm thuộc tính khác với gom cụm đối tượng; gom cụm thuộc tính nhằm nhóm các thuộc tính thành các cụm chứ không phải các đối tượng. Gom cụm thuộc tính nhóm các thuộc tính thành các cụm sao cho các đặc trưng trong cùng một cụm được mong đợi có độ tương đồng cao, nhưng trong các cụm khác nhau có độ tương đồng thấp.

Trong phần này, luận án tập trung nghiên cứu một số phương pháp lựa chọn thuộc tính dựa trên phép gom cụm và đề xuất một thuật toán tính toán tập rút gọn bằng cách sử dụng phép gom cụm thuộc tính.

Các thuật toán lựa chọn thuộc tính dựa trên gom cụm xuất phát từ một ý tưởng đơn giản. Nó chia không gian thuộc tính ban đầu thành một họ các cụm. Nhìn chung, các hàm đo độ tương quan thường được sử dụng để đánh giá mức độ tương tự giữa hai thuộc tính trong quá trình gom cụm. Điều này làm cho các thuộc tính trong cùng một cụm là tương quan với nhau, dẫn đến việc có thể chỉ cần lựa chọn một thuộc tính để đại diện cho mỗi

cụm, các thuộc tính còn lại được coi là dư thừa. Tập hợp các thuộc tính đại diện là tập các thuộc tính có liên quan, không dư thừa và được lấy làm một tập rút gọn.

Đối với cách tiếp cận bài toán lựa chọn thuộc tính dựa trên gom cụm này, có hai vấn đề cốt lõi cần được xem xét kỹ lưỡng, đó là việc lựa chọn hàm đo độ tương tự và phương pháp gom cụm để sử dụng. Hàm tương tự đo mức độ giống nhau giữa hai thuộc tính trong không gian thuộc tính; phương pháp gom cụm phân chia các thuộc tính thành các cụm bằng cách sử dụng hàm đo độ tương tự đã chọn.

Trong [41] Hong và cộng sự đề xuất thuật toán lựa chọn thuộc tính dựa trên gom cụm có tên gọi là MNF (Most Neighbors First). MNF sử dụng độ phụ thuộc tương đối của một thuộc tính vào một thuộc tính khác để xây dựng thước đo độ tương tự cho một cặp thuộc tính. Cũng như trong định nghĩa 3.5, trong một bảng quyết định $DT = (U, C \cup \{d\})$, độ phụ thuộc tương đối của thuộc tính của thuộc tính a_i vào thuộc tính a_j được xác định bởi:

$$k_{a_j}(a_i) = \frac{|\Pi_{a_j}(U)|}{|\Pi_{a_j \cup a_i}(U)|} \quad (3.6)$$

Trong đó $|\Pi_{a_j}(U)|$ là phép chiếu của U trên thuộc tính a_j .

Lưu ý rằng độ phụ thuộc tương đối của thuộc tính a_i vào thuộc tính a_j không có tính chất đối xứng, nghĩa là $k_{a_i}(a_j)$ và $k_{a_j}(a_i)$ thường không bằng nhau. Do đó, Hong và các cộng sự đã sử dụng nghịch đảo giá trị trung bình của $k_{a_i}(a_j)$ và $k_{a_j}(a_i)$ để biểu diễn độ tương đồng giữa hai thuộc tính a_i và a_j . Như vậy, độ tương đồng giữa 2 thuộc tính được định nghĩa như sau:

$$d(a_i, a_j) = \frac{1}{\text{Avg}(k_{a_i}(a_j), k_{a_j}(a_i))} \quad (3.7)$$

Quá trình bắt đầu chọn ngẫu nhiên k thuộc tính đại diện làm trung tâm cụm. Tại mỗi bước lặp, MNF tính toán độ khác biệt giữa các thuộc tính không đại diện và các thuộc tính đại diện. Các thuộc tính không đại diện được phân bổ vào cụm có đại diện gần nhất, sau đó trung tâm cụm được cập nhật bằng thuộc tính có số láng giềng nhiều nhất.

Thuật toán 3.4 Thuật toán gom cụm thuộc tính MNF

Đầu vào: Bảng quyết định $DT = (U, C \cup \{d\})$ và Bảng quyết định k cụm cho trước.

Đầu ra: k cụm thuộc tính với các tâm cụm.

Thuật toán:

Bước 1: Chọn ngẫu nhiên k thuộc tính $A_1^c, A_2^c, \dots, A_k^c$ làm thuộc tính đại diện cho mỗi cụm

Bước 2: Với mỗi thuộc tính không phải là thuộc tính đại diện $A_i \in A - A_c$, tính khoảng cách $d(A_i, A_{j_c})$ giữa thuộc tính A_i và thuộc tính đại diện A_{j_c} của mỗi cụm với $1 \leq j \leq k$.

Bước 3: Phân bổ tất cả các thuộc tính không phải là thuộc tính tâm vào cụm có tâm gần nhất dựa vào độ đo khoảng cách được tính ở bước 2.

Bước 4: Với mỗi cụm C_t tính khoảng cách giữa các cặp thuộc tính trong cụm

Bước 5: Tính bán kính mỗi cụm theo công thức:

$$r_t = \frac{\sum_{i \neq j} d(A_{t,i}, A_{t,j})}{C_2^n} \quad (3.8)$$

Với $d(A_{t,i}, A_{t,j})$ là khoảng cách giữa hai thuộc tính $A_{t,i}$ và $A_{t,j}$ trong mỗi cụm C_t , n là số lượng thuộc tính không bao gồm C_t và C_2^n là số cặp thuộc tính trong mỗi cụm với $C_2^n = \frac{n(n-1)}{2}$.

Bước 6: Với mỗi thuộc tính $A_{t,i}$ trong mỗi cụm C_t (bao gồm cả tâm A_t^c), tìm tập các thuộc tính (gọi là $Near(A_{t,i})$) với khoảng cách $A_{t,i}$ trong khoảng r_t như sau:

$$Near(A_{t,i}) = \{A_{t,i} | A_{t,i} \in C_t \text{ and } d(A_{t,i}, A_{t,j}) \leq r_t\}$$

Như vậy, thuộc tính nào có tập $NEAR$ lớn hơn thì sẽ có khả năng cao trở thành thuộc tính đại diện cho cụm.

Bước 7: Với mỗi cụm C_t , tìm thuộc tính $A_{t,l}$ với tập $NEAR$ nhiều thuộc tính nhất và gán thuộc tính này thành tâm A_t^c của cụm C_t .

Bước 8: Lặp lại bước 2 đến bước 7 cho đến khi các cụm hội tụ (không còn sự thay đổi)

Thuật toán MNF về cơ bản đã đạt được hiệu quả trong việc giảm số thuộc tính của trong một hệ thông tin. Tuy nhiên, vì hàm đo độ tương đồng (khoảng cách) giữa 2 thuộc tính $d(a_i, a_j)$ trong MNF không phải là một metric, nên sự hội tụ của thuật toán MNF có thể không được đảm bảo. Đối với nhiều tập dữ liệu, MNF có thể phải lặp lại nhiều lần các bước thực hiện mới có thể tìm ra kết quả gom cụm [17].

Có thể thấy, gom cụm thuộc tính cũng là bài toán NP-khó giống với các thuật toán lựa chọn thuộc tính thông thường vì nó phải thực hiện tính toán độ tương tự giữa các cặp thuộc tính ở mỗi cụm.

Như chúng ta biết, giải thuật di truyền ngày càng trở nên quan trọng trong việc giải quyết các vấn đề NP-khó trong khai phá dữ liệu. Giải thuật di truyền có thể cung cấp các giải pháp hiệu quả với các bài toán yêu cầu thời gian tính toán [42]. Trên cơ sở đó, trong [17, 21, 38, 40] các tác giả đã đề xuất phương pháp tính toán tập rút gọn xấp xỉ dựa vào gom cụm thuộc tính bằng giải thuật di truyền.

Mặc dù các thuật toán rút gọn thuộc tính dựa trên gom cụm thuộc tính đã nhận được nhiều sự chú ý trong thời gian gần đây, tuy nhiên số lượng các nghiên cứu vẫn còn nhiều hạn chế. Trong phần tiếp theo, luận án đề xuất một phương pháp mới rút gọn thuộc tính trong bảng quyết định dựa vào gom cụm, với tên gọi ACBRC.

3.4 Đề xuất thuật toán rút gọn thuộc tính dựa vào gom cụm ACBRC

Trong phần này, luận án đề xuất một thuật toán để tìm tập rút gọn xấp xỉ trong một bảng quyết định. Thuật toán được đặt tên là ACBRC (Attribute Clustering Based Reduct Computing – Tính toán tập rút gọn dựa trên gom cụm thuộc tính).

3.4.1 Ý tưởng và những định nghĩa cơ bản

Thuật toán ACBRC hoạt động trong ba giai đoạn chính. Trong giai đoạn đầu, các thuộc tính không liên quan sẽ bị loại bỏ. Trong giai đoạn thứ hai, các thuộc tính có liên quan được phân chia thành một số cụm thích hợp bằng phương pháp gom cụm phân hoạch xung quanh medoids (Partitioning Around Medoids - PAM) kết hợp với một metric đặc biệt trong không gian thuộc tính là biến thể thông tin chuẩn hóa (Normalized Variation of Information). Trong giai đoạn thứ ba, một thuộc tính đại diện từ mỗi cụm được chọn là

thuộc tính có độ liên quan lớn nhất với thuộc tính quyết định. Các thuộc tính được lựa chọn tạo thành một tập rút gọn xấp xỉ.

3.4.2 Giới thiệu thuật toán k-medoids

Thuật toán k-medoids [1] là một cách tiếp cận gom cụm liên quan đến thuật toán gom cụm k-means [1] dùng để phân chia một tập dữ liệu thành k cụm. Trong gom cụm k-medoids, mỗi cụm được đại diện bởi một trong các đối tượng có trong cụm. Những điểm này được gọi là các medoid của các cụm. Thuật ngữ medoid dùng để chỉ một đối tượng trong một cụm mà khoảng cách trung bình giữa nó và tất cả các thành viên khác của cụm là nhỏ nhất. Nó tương ứng với điểm nằm ở vị trí trung tâm nhất trong cụm. Mỗi medoid của một cụm có thể được coi là một đại diện cho các thành viên của cụm đó. Chú ý rằng, đối với phương pháp gom cụm k-means, tâm của một cụm nào đó được tính là véc tơ trung bình của tất cả các véc tơ biểu diễn các đối tượng trong cụm. Việc làm này bị ảnh hưởng bởi các giá trị ngoại lai. Do sử dụng medoid làm trung tâm cụm thay cho trung bình, thuật toán k-medoids ít bị ảnh hưởng bởi các giá trị ngoại lai hơn thuật toán k-means và có thể hoạt động với bất kỳ ma trận khoảng cách nào [1].

Thuật toán gom cụm phổ biến nhất theo cách tiếp cận k-medoids là thuật toán PAM (Phân vùng xung quanh Medoids) [1]. Một cách ngắn gọn, thuật toán PAM tiến hành theo các bước sau:

Thuật toán 3.5 Thuật toán PAM

Đầu vào: Bảng quyết định $DT = (U, C \cup \{d\})$, số lượng các cụm k

Đầu ra: k cụm đối tượng.

Thuật toán PAM:

Bước 1: Chọn ngẫu nhiên k đối tượng làm k-medoids

Bước 2: Gán mọi đối tượng cho medoid gần nhất đối với nó.

Bước 3: Tính tổng chi phí E cho cấu hình gom cụm thu được bằng cách sử dụng công thức:

$$E = \sum_{i=1}^k \sum_{x \in C_i} |x - m_i| \quad (3.9)$$

Với x là một đối tượng trong cụm C_i , m_i là medoid hiện tại của C_i và giá trị tuyệt đối $|x - m_i|$ là khoảng cách giữa x và m_i .

Bước 4: Đối với mỗi medoid m :

Đối với mỗi điểm dữ liệu không phải là medoid x

Hoán đổi m và x và tính tổng chi phí E' của cấu hình cụm thu được;

Bước 5: Nếu $E' < E$, m được thay bằng x ;

Bước 6: Lặp lại các bước 4, 5 cho đến khi không có thay đổi trong các medoids

Độ phức tạp của PAM trong mỗi lần lặp (bước 3-4) là $O(k(n - k)^2)$ trong đó n là số đối tượng có trong tập dữ liệu, k là số cụm, còn độ phức tạp chung của PAM là $O(n^2 k^2)$ [1]. Vì vậy, độ phức tạp của phương pháp k-medoids nói chung cao hơn phương pháp k-means. Tuy nhiên phương pháp k-medoids có lợi thế là nó đảm bảo rằng tất cả các trung tâm của các cụm thu được chính là các đối tượng có trong dữ liệu. Đặc tính này rất quan trọng đối với vấn đề lựa chọn thuộc tính dựa vào gom cụm, vì ở đây không chỉ cần nhóm các thuộc tính thành các cụm mà còn phải tìm được thuộc tính đại diện cho mỗi cụm. Với lý do này, trong thuật toán lựa chọn thuộc tính bằng cách gom cụm ACBRC đề xuất luận án sử dụng phương pháp gom cụm k-medoid.

Thuật toán PAM được cài đặt sẵn trong ngôn ngữ lập trình R. Để tính PAM, ta gọi hàm `pam()` trong gói “cluster”. Đối với thuật toán PAM, người dùng phải chỉ định số cụm k . Tuy nhiên, trong R ta còn có một phiên bản nâng cao của hàm `pam()` là hàm `pamk()` trong gói “fpc”. `pamk()` không yêu cầu người dùng chọn số cụm k . Thay vào đó, nó thực hiện việc gom cụm bằng phương pháp phân vùng xung quanh các medoids với số lượng các cụm được ước tính bằng phương pháp “chiều rộng hình bóng trung bình tối ưu” (optimum average silhouette width) [43].

Nói một cách ngắn gọn, cách tiếp cận hình bóng trung bình (average silhouette) dựa vào việc đánh giá chất lượng của một phép gom cụm. Chất lượng này được xác định thông qua mức độ phù hợp của mỗi đối tượng nằm trong cụm mà nó được gom vào. Chiều rộng hình bóng trung bình càng cao thì kết quả gom cụm thu được càng tốt. Phương pháp hình bóng trung bình tính hình bóng trung bình của các kết quả gom cụm với số cụm k khác nhau. Số cụm tối ưu k là số cụm tối đa hóa hình bóng trung bình trong phạm vi có thể.

3.4.3 Thuật toán rút gọn thuộc tính dựa vào gom cụm ACBRC

Trong một bảng quyết định, các thuộc tính không liên quan không góp phần vào việc làm tăng độ chính xác của dự đoán và các thuộc tính thừa không giúp thêm được gì để có được một kết quả dự đoán tốt hơn vì hầu hết thông tin mà chúng cung cấp đã có trong các thuộc tính khác. Các thuộc tính không liên quan, cùng với các thuộc tính dư thừa, ảnh hưởng nghiêm trọng đến độ chính xác của một thuật toán phân lớp [1, 3]. Do đó, thuật toán rút gọn thuộc tính phải có thể xác định và loại bỏ càng nhiều càng tốt các thuộc tính không liên quan và thuộc tính dư thừa. Hơn nữa, một tập con thuộc tính tốt phải chứa các thuộc tính tương đồng với thuộc tính quyết định, nhưng không tương đồng nhau. Với những nhận xét quan trọng này, chương này của luận án đề xuất ACBRC, một thuật toán tính toán tập rút gọn xấp xỉ dựa trên gom cụm thuộc tính, có thể xử lý một cách hiệu quả các thuộc tính không liên quan và dư thừa trong một bảng quyết định.

Để mô tả chính xác hơn về thuật toán, trước tiên phần này trình bày một số khái niệm liên quan:

Định nghĩa 3.9. [1, 3] Cho $DT = (U, C \cup \{d\})$ là một bảng quyết định. Gom cụm thuộc tính trong DT là phép phân chia tập C các thuộc tính có điều kiện thành một họ gồm các tập con $C_X = \{C_1, C_2, \dots, C_k\}$ của C , sao cho $C_1 \cup C_2 \cup \dots \cup C_k = C$, $C_i \neq \emptyset$, và $C_i \cap C_j = \emptyset$, với $i \neq j$.

Định nghĩa 3.10. [1, 3] Mức độ không liên quan giữa thuộc tính điều kiện $a_i \in C$ và thuộc tính quyết định d được đo bằng giá trị khoảng cách $NVI(a_i, d)$, trong đó $NVI(a_i, a_j)$ là độ đo biến thể thông tin chuẩn hóa giữa thuộc tính a_i với thuộc tính a_j đã được định

nghĩa tại 2.16 Chương 2. Giá trị khoảng cách $NVI(a_i, d)$ càng lớn thì mức độ không liên quan giữa chúng càng nhỏ.

Định nghĩa 3.11 Cho G là một cụm thuộc tính. Thuộc tính $a^R \in G$ là thuộc tính đại diện của cụm nếu và chỉ nếu:

$$a^R = \operatorname{argmin}_{a \in G} NVI(a, d). \quad (3.10)$$

Công thức (3.10) nói lên a^R là thuộc tính có liên quan mạnh nhất và có thể được xem như một thuộc tính có liên quan tiêu biểu cho tất cả các thuộc tính trong cụm G .

Vì giá trị lớn nhất mà độ đo khoảng cách NVI có thể đạt được là 1, trong thuật toán đề xuất ACBRC, ta quy định nếu $NVI(X_i, d)$ lớn hơn giá trị ngưỡng $\delta = 0.98$, thì có thể khẳng định a_i là thuộc tính không liên quan với d ; trường hợp ngược lại, a_i là thuộc tính liên quan. Điều này được chứng minh trong tính toán thử nghiệm, với các giá trị ngưỡng thử nghiệm khác nhau, đặc biệt là $\delta = 0.95, 0.98$ và 0.99 làm tiêu chí để loại bỏ thuộc tính không liên quan. Kết quả cho thấy $\delta = 0.98$ là giá trị ngưỡng tốt nhất.

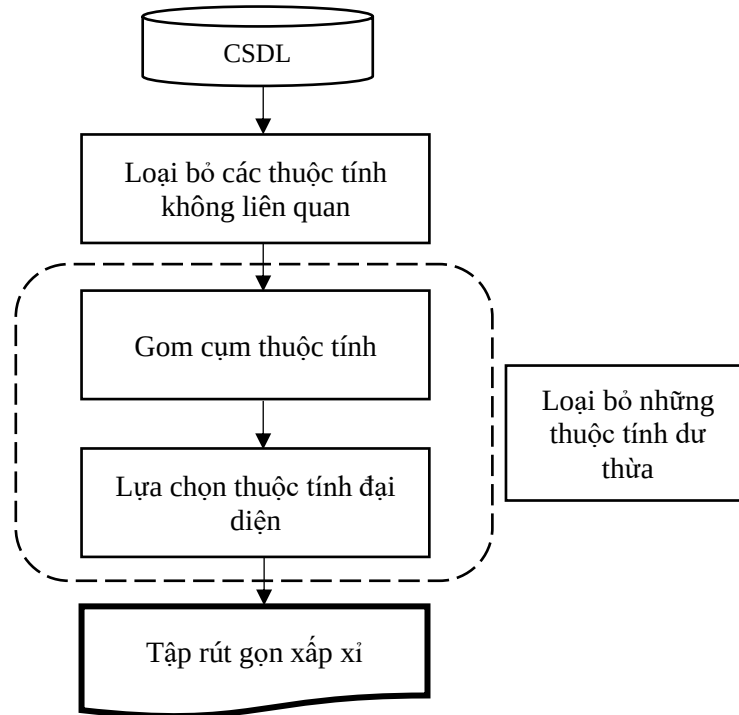
Sử dụng các định nghĩa trên, thuật toán ACBRC là quá trình bao gồm ba công đoạn liên tiếp, với khung hoạt động được thể hiện trong hình 3.1.

Các công đoạn là như sau:

(1) Đầu tiên, loại bỏ các thuộc tính không liên quan. Để thực hiện việc này, khoảng cách $NVI(a, d)$ được tính toán giữa mỗi thuộc tính điều kiện a và thuộc tính quyết định d . Như đã nói, một thuộc tính cho giá trị khoảng cách $NVI(a, d)$ càng lớn thì càng không liên quan với d , khả năng phân lớp đúng các đối tượng của nó càng thấp. Thuộc tính có mức độ không liên quan lớn hơn 0,98 sẽ bị loại bỏ khỏi tập thuộc tính điều kiện ban đầu.

(2) Gom cụm các thuộc tính có liên quan bằng cách sử dụng hàm `pamk()` trong gói R “fpc” của hệ thống R, sử dụng độ đo khoảng cách NVI . `pamk()` là phiên bản nâng cao của `pam()`, có thể hoạt động với bất kỳ ma trận khoảng cách nào và không yêu cầu người dùng phải cung cấp số cụm k . Thay vào đó, nó thực hiện thuật toán gom cụm PAM với số cụm được ước tính bằng phương pháp chiều rộng hình bóng trung bình tối ưu, được mô tả trong [43].

(3) Cuối cùng, từ mỗi cụm thuộc tính đã được gom chọn một thuộc tính có mức độ liên quan với thuộc tính quyết định mạnh nhất. Thuộc tính này được xem như là thuộc tính đại diện cho tất cả các thuộc tính trong cụm. Khi một thuộc tính được chọn làm đại diện, các thuộc tính khác trong cùng một cụm sẽ bị loại bỏ. Tập các thuộc tính đại diện chọn được từ các cụm tạo thành tập rút gọn xấp xỉ.



Hình 3.1 Hình minh họa thuật toán ACBRC

Tựa code của thuật toán ACBRC là như sau:

Thuật toán 3.6 Thuật toán ACBRC

Đầu vào: Bảng quyết định $DT = (U, C \cup \{d\})$, ngưỡng không liên quan $\delta = 0.98$.

Đầu ra: Tập rút gọn xấp xỉ Red .

Thuật toán:

// Loại bỏ các thuộc tính không liên quan.

(1) Với mỗi $a \in C$

(2) Gán $irrelevance = NVI(a, d)$.

(3) Nếu $irrelevance > \delta$ thì $C^R = C \setminus \{a\}$.

(4) Hết với mỗi

//Tính ma trận khoảng cách NVI giữa các cặp thuộc tính.

(5) Với mọi $(a_i, a_j) \in C^R$ tính

(6) $NVI[i, j] = NVI(a_i, a_j)$

(7) Hết với mọi

//Gom cụm thuộc tính

(8) Sử dụng hàm *pamk()* trong gói R “*fpc*” để gom cụm các thuộc tính trong C^R .

(9) Với mỗi cụm G

(10) $a^R = \operatorname{argmin}_{a \in G} NVI(X, d). Red = Red \cup a^R;$

(11) Hết với mỗi

3.4.4 Kết quả thực nghiệm thuật toán ACBRC

Thuật toán lựa chọn thuộc tính đề xuất ACBRC được cài đặt bằng ngôn ngữ lập trình R trên máy tính cá nhân. Tính toán thực nghiệm được thực hiện trên năm bộ dữ liệu chuẩn thu được từ kho dữ liệu chuẩn UCI. Mô tả của các bộ dữ liệu này được thể hiện trong bảng 3.4. Hai cột đầu tiên hiển thị tên và tên viết tắt của các tập dữ liệu, hai cột tiếp theo hiển thị số lượng cá thể và thuộc tính, cột cuối cùng hiển thị số lượng nhãn lớp của thuộc tính quyết định. Tất cả thuộc tính của các tập dữ liệu đều là thuộc tính phân lớp. Vì vậy, việc rời rạc hóa là không cần thiết.

Để đánh giá hiệu quả của thuật toán lựa chọn thuộc tính ACBRC, luận án đã tiến hành so sánh nó với thuật toán QuickReduct và CEBARKNC về số lượng thuộc tính được chọn, thời gian thực hiện và hiệu quả phân lớp của tập thuộc tính rút gọn được chọn. Để so sánh hiệu quả phân lớp của ACBRC, QuickReduct và CEBARKNC, luận án sử dụng C5.0 và Native Bayes, là hai thuật toán phân lớp phổ biến và được áp dụng rộng rãi trong nhiều lĩnh vực nghiên cứu khác nhau.

Bảng 3.4 Bảng mô tả các tập dữ liệu thực nghiệm

CSDL	Tên viết tắt	Số đối tượng	Số thuộc tính điều kiện	Số lớp
Chess	Chess	3196	36	2
Mushroom	Mushroom	8124	22	7
Soybean (small)	Soybean	47	35	4
Lung-cancer	Lung	32	56	3
Votes	Votes	435	16	2

Để sử dụng tốt nhất dữ liệu và thu được kết quả ổn định, chiến lược xác nhận chéo 3×10 được sử dụng. Nghĩa là, đối với mỗi tập dữ liệu, mỗi thuật toán rút gọn thuộc tính và mỗi thuật toán phân lớp, xác nhận chéo 10 lần được lặp lại 3 vòng. Tại mỗi vòng xác nhận chéo 10 lần thứ tự của các đối tượng trong tập dữ liệu được ngẫu nhiên hóa. Việc sắp xếp thứ tự ngẫu nhiên của các đối tượng nhằm giảm bớt tác động của thứ tự đến kết quả phân lớp. Với mỗi thuật toán phân lớp và mỗi tập dữ liệu luận án ghi nhận độ chính xác phân lớp thu được trong mỗi lần tính toán với các tập thuộc tính rút gọn cung cấp bởi ba thuật toán ACBRC, QuickReduct và CEBARKNC. Tính trung bình, ta thu được độ chính xác trung bình của từng thuật toán phân lớp theo từng thuật toán rút gọn thuộc tính trên mỗi tập dữ liệu.

Kết quả thực nghiệm thu được là như sau:

A. So sánh số lượng thuộc tính được chọn bởi ba thuật toán rút gọn thuộc tính

Bảng 3.5 cho thấy các thuộc tính chọn được bởi ba thuật toán rút gọn thuộc tính ACBRC, QuickReduct và CEBARKNC, khi chúng được áp dụng trên mỗi tập dữ liệu trong bảng 3.4.

Trong bảng, nếu chẳng hạn viết 13 20 5 9, điều có nghĩa là các thuộc tính số 13, số 20, số 5 và số 9 đã được chọn lập thành tập thuộc tính rút gọn.

Từ bảng 3.5, có thể thấy cả ba thuật toán đều rút gọn đáng kể số thuộc tính ban đầu. Tuy nhiên, số thuộc tính chọn được bởi thuật toán ACBRC nó chung là nhỏ hơn.

Bảng 3.5 Những thuộc tính được chọn bởi ba giải thuật rút gọn thuộc tính

CSDL	ACBRC	QuickReduct	CEBARKNC
Chess	7 29 32 8 10 18 33 14 16 21	21 10 29 14 28 1 15 16 6 33 7 35 18 34 11 5 17 23 36 26 20 30 4 24 12 27 25 31 3 9 13	21 10 33 32 6 35 15 1 34 7 16 23 17 4 2 30 5 27 3 9 20 25 31 12 13 24 18 28 26 36
Mushroom	13 20 5 9	15 6 22 11 12 5 1	5 20 22 21
Soybean	22 21	22 1	22 4
Lung	9 48	1 42 7 4	9 43 3 4
Votes	4 12	1 9 14 4 11 16 13 3 2 6	4 11 3 13 16 2 9 15 1

Bảng 3.6 Bảng so sánh thời gian thực hiện của các thuật toán (theo giây)

CSDL	ACBRC	QuickReduct	CEBARKNC
Chess	18.52	28.41	12.82
Mushroom	1.14	2.29	1.11
Soybean	0.64	0.12	0.36
Lung	0.84	0.31	0.44
Votes	0.64	0.73	0.53

Bảng 3.6, cho thấy thời gian thực hiện của ba thuật toán phụ thuộc vào đặc điểm của từng tập dữ liệu. Nhìn chung, thời gian thực hiện của thuật toán ACBRC lớn hơn một chút so với thuật toán QuickReduct và CEBARKNC, nhưng thời gian thực hiện của ACBRC là chấp nhận được.

B. Đánh giá hiệu suất phân lớp của thuật toán rút gọn thuộc tính ACBRC

Bảng 3.7 cho thấy khoảng tin cậy 95% của độ phân lớp chính xác trung bình thu được khi thực hiện xác nhận chéo 3×10 với hai thuật toán phân lớp C5.0 và Native Bayes trên 5 tập dữ liệu nguyên thủy (chưa rút gọn thuộc tính).

Bảng 3.7 Độ chính xác phân lớp khi chưa rút gọn thuộc tính

CSDL	C5.0	Bayes
Chess	0.9928 ± 0.0024	0.87868 ± 0.0126
Mushroom	1	0.94088 ± 0.0057
Soybean	0.975 ± 0.049	1
Lung	0.7667 ± 0.1701	0.56667 ± 0.2396
Votes	0.9674 ± 0.0139	0.90465 ± 0.0349

Bảng 3.8 Độ chính xác phân lớp với các thuộc tính được chọn bởi ACBRC

CSDL	C5.0	Bayes
Chess	0.9928 ± 0.0022	0.8906 ± 0.0129
Mushroom	1	0.9473 ± 0.0031
Soybean	1	1
Lung	0.8 ± 0.1996	0.6 ± 0.2134
Votes	0.9674 ± 0.0217	0.9581 ± 0.0164

Bảng 3.8 cho thấy khoảng tin cậy 95% thu được khi thực hiện xác nhận chéo 3×10 với hai thuật toán phân lớp trên 5 tập dữ liệu sau khi sử dụng thuật toán rút gọn thuộc tính ACBRC.

Kết quả thực nghiệm cho thấy, với cả năm bộ dữ liệu, độ chính xác phân lớp của các thuộc tính lựa chọn được bởi ACBRC lớn hơn độ chính xác phân lớp của các thuộc tính gốc (Bảng 3.7).

C. So sánh độ chính xác phân lớp của ba thuật toán rút gọn thuộc tính

Từ bảng 3.9, ta thấy đối với tập dữ liệu “Soybean”, “Lung” và “Votes”, độ chính xác phân lớp C5.0 với các thuộc tính chọn được bởi ACBRC cao hơn độ chính xác phân lớp với các thuộc tính chọn được bởi QuickReduct và CEBARKNC. Còn đối với hai tập

dữ liệu khác, kết quả phân lớp sau khi rút gọn thuộc tính bởi ACBRC gần bằng kết quả sau khi rút gọn thuộc tính bằng QuickReduct và CEBARKNC.

Bảng 3.9 Độ chính xác phân lớp bằng C5.0 sau khi sử dụng các phương pháp rút gọn thuộc tính khác nhau

CSLD	ACBRC	QuickReduct	CEBARKNC
Chess	0.9928 ± 0.0022	0.9931 ± 0.0024	0.9937 ± 0.0035
Mushroom	1	1	1
Soybean	1	0.975 ± 0.049	0.975 ± 0.049
Lung	0.8 ± 0.1996	0.8 ± 0.1445	0.7667 ± 0.1701
Votes	0.9674 ± 0.0217	0.9651 ± 0.014	0.9558 ± 0.0126

Bảng 3.10 cho thấy đối với hầu hết các tập dữ liệu, độ chính xác phân lớp Bayes với các thuộc tính được chọn bởi ACBRC lớn hơn độ chính xác phân lớp Bayes với các thuộc tính được chọn bởi QuickReduct và CEBARKNC.

Bảng 3.10 Độ chính xác phân lớp Bayes sử dụng các thuật toán rút gọn thuộc tính

CSDL	ACBRC	QuickReduct	CEBARKNC
Chess	0.8906 ± 0.0072	0.8881 ± 0.015	0.8947 ± 0.0061
Mushroom	0.9473 ± 0.0036	0.982 ± 0.0027	0.9807 ± 0.0031
Soybean	1	0.875 ± 0.1096	1
Lung	0.5334 ± 0.2425	0.4667 ± 0.2789	0.4667 ± 0.2425
Votes	0.9581 ± 0.0164	0.9279 ± 0.023	0.9302 ± 0.0192

3.5 Kết luận chương 3

Mục đích của lựa chọn thuộc tính là làm giảm số thuộc tính có trong tập dữ liệu, loại bỏ thuộc tính dư thừa, không liên quan mà vẫn không làm mất đi những thông tin cần thiết phục vụ nhiệm vụ khai phá dữ liệu. Chương 3 đã trình bày khái quát về vấn đề lựa chọn

thuộc tính, các phương pháp chính tìm tập rút gọn của một bảng quyết định và đề xuất một thuật toán mới, với tên gọi ACBRC, dựa trên gom cụm các thuộc tính. Đối với mỗi thuật toán, sau phần trình bày cơ sở lý thuyết là phần diễn giải quá trình thực hiện bằng tựa code và minh họa bằng ví dụ cụ thể. Thuật toán đề xuất ACBRC gồm 3 giai đoạn: Trong giai đoạn đầu, các thuộc tính không liên quan sẽ bị loại bỏ. Trong giai đoạn thứ hai, các thuộc tính có liên quan được phân chia thành một số cụm thích hợp bằng phương pháp gom cụm phân hoạch xung quanh Medoids PAM tích hợp với một metric đặc biệt trong không gian thuộc tính là Biến thể Thông tin Chuẩn hóa. Trong giai đoạn thứ ba, một thuộc tính đại diện từ mỗi cụm được chọn là thuộc tính có độ liên quan lớn nhất với thuộc tính quyết định. Các thuộc tính được lựa chọn tạo thành một tập rút gọn xấp xỉ.

Kết quả thử nghiệm cho thấy thuật toán đề xuất ACBRC là rất khả quan trong việc làm giảm số thuộc tính trong các tập dữ liệu cần khai phá các quy tắc phân lớp. Kết quả nghiên cứu trên đã được đăng trên tạp chí Journal of Computer Science and Cybernetics [CT2] năm 2022 và có thể làm tiền đề cho các hướng nghiên cứu phát triển tiếp theo.

CHƯƠNG 4. GOM CỤM DỮ LIỆU SỬ DỤNG LÝ THUYẾT TẬP THÔ

4.1 Mở đầu

Gom cụm là một kỹ thuật trong khai phá dữ liệu, nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tương tự nhau, đáng quan tâm trong tập dữ liệu lớn, từ đó cung cấp thông tin, tri thức hữu ích cho hỗ trợ cho việc ra quyết định [1].

Về bản chất ta có thể hiểu gom cụm là các qui trình tìm cách nhóm các đối tượng đã cho vào các cụm (clusters), sao cho các đối tượng trong cùng một cụm tương tự (similar) nhau và các đối tượng khác cụm thì không tương tự (dissimilar) [1]. Với phương pháp này, ta không biết trước các đối tượng sẽ thuộc vào các cụm như thế nào, vì thế gom cụm dữ liệu là một phương pháp học không có giám sát (unsupervised learning). Gom cụm dữ liệu được sử dụng nhiều trong các ứng dụng như phân loại động vật, thực vật, phân đoạn thị trường, phân loại khách hàng, trang web v.v. Ngoài ra, gom cụm dữ liệu còn có thể được sử dụng như một bước tiền xử lý cho các thuật toán khai thác dữ liệu khác.

Cho đến nay đã có nhiều phương pháp gom cụm đã được đề xuất. Tuy nhiên, theo các nghiên cứu, chưa có một phương pháp gom cụm tổng quát nào có thể giải quyết trọn vẹn cho tất cả các dạng cấu trúc cụm dữ liệu.

Gần đây, gom cụm dữ liệu phân loại đã thu hút nhiều sự chú ý từ cộng đồng nghiên cứu khai phá dữ liệu. Một số thuật toán gom cụm dữ liệu phân loại đã được đề xuất. Mặc dù những thuật toán này có những đóng góp quan trọng cho vấn đề gom cụm dữ liệu phân loại, nhưng chúng không được thiết kế để xử lý sự không chắc chắn trong quá trình gom cụm. Xử lý sự không chắc chắn trong quá trình gom cụm là một vấn đề quan trọng, bởi vì trong nhiều ứng dụng thực tế thường không có ranh giới rõ ràng giữa các cụm.

Một cách tiếp cận phổ biến để xử lý độ không chắc chắn là sử dụng Lý thuyết tập thô (Rough Sets Theory), đề xuất bởi Pawlak [8]. Trong những năm gần đây, một số tác giả đã đề xuất một cách tiếp cận mới bài toán gom cụm dữ liệu phân loại bằng cách sử dụng lý thuyết tập thô [29, 22, 30, 31, 24].

Trong chương này, luận án sẽ tập trung nghiên cứu kỹ thuật gom cụm dữ liệu phân loại sử dụng Lý thuyết tập thô mở rộng và đề xuất một thuật toán mới nhằm nâng độ cao hiệu quả trong gom cụm dữ liệu phân loại.

4.2 Khái quát bài toán gom cụm dữ liệu

Gom cụm dữ liệu (Data Clustering) là một kỹ thuật cơ bản trong khai thác dữ liệu và học máy. Nó có thể được định nghĩa như sau. Cho $D = \{x_1, x_2, \dots, x_n\}$ là tập hợp n đối tượng, trong đó mỗi x_i là một vectơ n chiều trong không gian đặc trưng đã cho. Bài toán gom cụm dữ liệu là hãy nhóm các đối tượng vào các cụm sao cho các đối tượng thuộc cùng một lớp là tương đồng còn các đối tượng thuộc các cụm khác nhau sẽ không tương đồng. Với kỹ thuật này, ta không biết trước các đối tượng sẽ thuộc vào các cụm như thế nào, vì thế gom cụm dữ liệu là một phương pháp học không có giám sát (unsupervised learning). Bài toán gom cụm xuất hiện trong nhiều lĩnh vực khác nhau như nhận dạng mẫu, tin sinh học, y học, truy xuất thông tin, thị giác máy tính, v.v. Ngoài ra, gom cụm dữ liệu còn có thể được sử dụng như một bước tiền xử lý cho các thuật toán khai thác dữ liệu khác.

4.2.1 Các bước giải bài toán gom cụm dữ liệu

Thông thường, quá trình giải bài toán gom cụm dữ liệu gồm các bước sau đây [1, 7, 44].

Bước 1. Lựa chọn các thuộc tính. Mục đích của bước này là lựa chọn những thuộc tính hợp lệ đối với mục tiêu gom cụm có trong tập dữ liệu, loại bỏ tối đa thuộc tính dư thừa, nhằm tối ưu hóa hiệu quả của thuật toán gom cụm.

Bước 2. Lựa chọn thuật toán phân cụm. Đây là bước lựa chọn thuật toán phân cụm sao cho nó phù hợp nhất với tập dữ liệu đầu vào và yêu cầu của bài toán. Xây dựng hàm tính độ tương tự (similarity measure) giữa hai đối tượng và tiêu chí gom cụm (clustering criterion) là những đặc điểm được xem xét chính để quyết định lựa chọn một thuật toán gom cụm hiệu quả.

Bước 3. Xác nhận tính hợp lệ của các kết quả (validation of results). Để xác nhận tính hợp lệ của các kết quả gom cụm, các phương pháp thường được sử dụng bao gồm kỹ thuật

trực quan, các tiêu chuẩn đánh giá và so sánh kết quả với các thuật toán khác. Mặc dù vậy, khi tập dữ liệu lớn, việc sử dụng các kỹ thuật trực quan là không thể.

Bước 4. Lý giải kết quả. Đây là bước phối hợp với các chuyên gia trong lĩnh vực ứng dụng để kiểm chứng, phân tích, giải thích kết quả và đưa ra những kết luận đúng đắn cho người sử dụng.

4.2.2 Các loại phương pháp gom cụm dữ liệu.

Cho đến nay có một số lượng lớn các thuật toán gom cụm trong các tài liệu. Việc lựa chọn thuật toán phân cụm phụ thuộc cả vào loại dữ liệu có sẵn cũng như mục đích và ứng dụng cụ thể. Đã có nhiều phương pháp gom cụm đã được đề xuất. Có thể phân loại chúng thành sáu loại, bao gồm phương pháp phân hoạch, phân cấp, dựa trên lưới, dựa trên mật độ, dựa trên mô hình [1, 7]. Trong 6 phương pháp gom cụm này, phương pháp phân hoạch và phân cấp là phổ biến nhất.

- *Phương pháp phân hoạch (Partitioning methods).* Cho tập dữ liệu gồm có n đối tượng, một phương pháp gom cụm phân hoạch tiến hành phân chia các đối tượng vào k cụm sao cho mỗi cụm chứa ít nhất một đối tượng và mỗi đối tượng thuộc về một cụm duy nhất. Chìa khóa của phương pháp này đó là sử dụng một hàm tiêu chuẩn để đánh giá chất lượng của các cụm cũng như để định hướng cho quá trình tìm kiếm phân hoạch dữ liệu. Với phương pháp này, thông thường người ta bắt đầu khởi tạo một phân hoạch ban đầu cho tập dữ liệu theo phép ngẫu nhiên, và liên tục tinh chỉnh nó cho đến khi thu được một phân hoạch mong muốn, thỏa mãn điều kiện cho trước bằng cách tính các giá trị đo độ tương tự hoặc không tương tự giữa các đối tượng. Có thể thấy ý tưởng chính của các thuật toán gom cụm dựa trên phương pháp phân hoạch là sử dụng chiến lược tham lam (Greedy) để tìm kiếm tất cả các phân hoạch. Do đó, độ phức tạp thuật toán là rất lớn.

Một số thuật toán gom cụm phân hoạch điển hình như k-means, k-modes, PAM (Partitioning Around Medoids), CLARA (Clustering Large Applications), CLARANS (Clustering Large Applications) v.v. [1].

- *Phương pháp phân cấp (Hierarchical methods).* Phương pháp gom cụm phân cấp tạo ra sự phân cấp các cụm đối tượng. Bằng kỹ thuật đệ quy, gom cụm phân cấp sắp xếp

một tập dữ liệu đã cho thành một cấu trúc có dạng hình cây. Cây gom cụm có thể được xây dựng theo hai phương pháp: phân chia dần từ trên xuống (Divisive - Top down) và hương pháp gộp dần từ dưới lên (Agglomerative - Bottom up). Phương pháp phân cấp phân chia dần sử dụng chiến lược chia để trị, các thuật toán gom cụm dữ liệu theo hướng tiếp cận này bắt đầu với trạng thái là tất cả các đối tượng được xếp trong cùng một cụm. Ứng với mỗi vòng lặp, cụm được chọn sẽ tiến hành phân chia thành các cụm nhỏ hơn dựa trên một phép đo độ tương tự nào đó giữa các cụm. Quá trình thực hiện cho đến khi điều kiện dừng thỏa mãn. Phương pháp gộp bắt đầu với mỗi đối tượng được khởi tạo tương ứng với các cụm riêng biệt, sau đó tiến hành nhóm các đối tượng theo một độ đo tương tự ví dụ như độ đo khoảng cách giữa hai tâm của hai cụm, quá trình này được thực hiện cho đến khi cho đến khi các điều kiện kết thúc thỏa mãn. Có thể thấy, cách tiếp cận này dựa trên chiến lược tham lam trong quá trình gom cụm.

Đặc điểm chung của lớp các thuật toán gom cụm phân cấp:

- Kết quả gom cụm là một cây các cụm;
- Thuật toán có tốc độ xử lý nhanh;
- Thuật toán không bắt buộc phải khai báo trước số cụm k đầu vào.

Một số thuật toán gom cụm phân cấp điển hình: BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), CURE (Clustering Using REpresentatives), ROCK, v.v. [1].

- *Phương pháp dựa trên mật độ (Density-based methods)*. Phương pháp gom cụm này dựa vào mật độ của các đối tượng để xác định các cụm dữ liệu. Mật độ thường được định nghĩa như là số các đối tượng lân cận của một đối tượng dữ liệu theo một ngưỡng nào đó, nhờ đó nó có thể phát hiện ra các cụm dữ liệu với hình dạng bất kỳ (chẳng hạn như cụm hình chữ “S” hoặc hình oval). Tuy nhiên, việc xác định các tham số mật độ của thuật toán rất khó khăn, trong khi các tham số này lại có tác động rất lớn đến kết quả gom cụm dữ liệu.

Một số thuật toán gom cụm dữ liệu dựa trên mật độ điển hình: DBSCAN (Density-Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify the Clustering Structure), DENCLUE (DENsity based LUStering), v.v. [1].

- *Phương pháp dựa trên lưới (Grid-based methods)*. Phương pháp gom cụm dựa trên mật độ không thích hợp khi xử lý dữ liệu nhiều chiều (high dimension datasets), để giải quyết cho vấn đề trên, các nhà nghiên cứu đã đề xuất phương pháp gom cụm dựa trên lưới. Mục tiêu của phương pháp này là lượng tử hoá không gian đối tượng thành các ô (Cell) dưới dạng cấu trúc dữ liệu lưới, sau đó các kỹ thuật gom cụm dữ liệu sẽ được thực hiện trên các đối tượng trong từng ô trên lưới. Cách tiếp cận này không di chuyển các đối tượng trong các ô mà xây dựng nhiều mức phân cấp của nhóm các đối tượng trong một cell. Ưu điểm của phương pháp gom cụm dữ liệu dựa trên lưới là thời gian xử lý nhanh và độc lập với số đối tượng dữ liệu trong tập dữ liệu ban đầu.

Gom cụm dữ liệu là vấn đề mở và khó, vì rằng người ta cần phải giải quyết nhiều vấn đề cơ bản như [1, 7]:

- Xây dựng hàm tính độ tương tự giữa các đối tượng nhiều dạng dữ liệu khác nhau, đặc biệt là đối với các dữ liệu phân loại (categorical data) và các dữ liệu hỗn hợp. Hiện nay, dữ liệu phân loại và dữ liệu hỗn hợp ngày càng gia tăng trong các CSDL cần gom cụm;
- Xây dựng thuật toán gom cụm và các xác lập các điều kiện khởi tạo;
- Xây dựng thuật toán gom cụm và các xác lập các điều kiện khởi tạo;
- Xây dựng các thủ tục đánh giá và biểu diễn kết quả gom cụm.

4.2.3 Các tiêu chí đánh giá một thuật toán gom cụm hiệu.

Mục đích của gom cụm là tìm ra bản chất tự nhiên bên trong các cụm dữ liệu. Tuy nhiên, không có tiêu chí nào để đánh hiệu quả của một phép gom cụm là tốt nhất. Chất lượng gom cụm cũng còn phụ thuộc vào công đoạn tiền xử lý dữ liệu. Dưới đây là một số tiêu chí để đánh giá một thuật toán gom cụm hiệu quả trong khai phá dữ liệu [1, 7].

- Có khả năng mở rộng. Thuật toán có khả năng thực hiện tốt với tập các đối tượng dữ liệu lớn.
- Có khả năng ứng dụng hiệu quả với các kiểu dữ liệu khác nhau. Không phải chỉ trên các tập dữ liệu có một kiểu dữ liệu liên tục hay phân loại, mà còn phải khả dụng trên cả các tập dữ liệu hỗn hợp.
- Có khả năng khám phá ra các cụm với hình dạng bất kỳ.

- Thuật toán yêu cầu tối thiểu các tham số đầu vào. Các giá trị đầu vào thường ảnh hưởng đến thuật toán gom cụm và làm tăng độ phức tạp tính toán khi tích hợp vào các CSDL lớn.
- Khả năng thích nghi với các dữ liệu có trong thực tế thường chứa dữ liệu nhiễu, dữ liệu không chắc chắn, không đầy đủ.
- Ít nhạy cảm với thứ tự sắp xếp dữ liệu đầu vào: Một thuật toán gom cụm hiệu quả sẽ không bị ảnh hưởng bởi thứ tự sắp xếp các đối tượng dữ liệu đầu vào.

Theo các tài liệu nghiên cứu, đến nay nhiều phương pháp gom cụm dữ liệu đã được đề xuất, tuy nhiên chưa có một phương pháp gom cụm tổng quát nào có thể giải quyết trọn vẹn cho tất cả các dạng cấu trúc cụm dữ liệu.

4.3 Gom cụm dữ liệu phân loại sử dụng Lý thuyết tập thô

Như đã trình bày ở trên, cho đến nay nhiều phương pháp gom cụm đã được đề xuất. Tuy nhiên, hầu hết các phương pháp gom cụm đều tập trung vào các tập dữ liệu số, trong đó mỗi thuộc tính mô tả các đối tượng đều có miền giá trị là một khoảng giá trị thực liên tục, mỗi đối tượng dữ liệu số được coi là một điểm trong không gian metric đa chiều với một metric đo khoảng cách giữa các đối tượng, chẳng hạn như metric Euclide hoặc metric Mahalanobis [1, 7, 22]. Tuy nhiên, trong các ứng dụng thực tiễn thường gặp phải những tập dữ liệu với các thuộc tính là những thuộc tính phân loại hay phạm trù (categorical), tức là những thuộc tính có miền giá trị D hữu hạn và không có thứ tự; (chẳng hạn như màu tóc, quốc tịch v.v.); trong D chỉ được phép so sánh giữa các giá trị, với bất kỳ $a, b \in D$ hoặc $a = b$ hoặc $a \neq b$. Với dữ liệu phân loại ta không thể định nghĩa hàm khoảng cách một cách tự nhiên.

Trong những năm gần đây, gom cụm dữ liệu phân loại đã thu hút nhiều sự chú ý từ cộng đồng nghiên cứu khai phá dữ liệu. Một số thuật toán gom cụm dữ liệu phân loại đã được đề xuất [1, 10, 11, 12, 45, 46]. Mặc dù những thuật toán này có những đóng góp quan trọng cho vấn đề gom cụm dữ liệu phân loại, nhưng chúng không được thiết kế để xử lý sự không chắc chắn trong quá trình gom cụm. Xử lý sự không chắc chắn trong quá trình gom cụm là một vấn đề quan trọng, bởi vì trong nhiều ứng dụng thực tế thường không có ranh

giới rõ ràng giữa các cụm. Để xử lý sự không chắc chắn trong quá trình gom cụm dữ liệu phân loại, gần đây nhiều nhà nghiên cứu đã nghiên cứu các thuật toán áp dụng lý thuyết tập mờ (Fuzzy set theory) do Lotfi Zadeh đề xuất vào năm 1965. Tuy nhiên, các thuật toán này yêu cầu nhiều lần chạy mới có thể thiết lập được giá trị ổn định cần thiết cho tham số sử dụng để kiểm soát mức độ thành viên mờ [7, 47]. Một cách tiếp cận phổ biến để xử lý độ không chắc chắn là sử dụng Lý thuyết tập thô (Rough Sets Theory), đề xuất bởi Pawlak [8]. Nền tảng cho sự thành công của lý thuyết tập thô là với nó không yêu cầu bất kỳ thông tin bổ sung nào về dữ liệu như phân bố xác suất tiên nghiệm trong thống kê và hàm thành viên trong lý thuyết tập mờ [2, 10].

Gần đây, một số tác giả đã đề xuất một cách tiếp cận mới bài toán gom cụm dữ liệu phân loại bằng cách sử dụng lý thuyết tập thô và kỹ thuật phân chia dần [29, 22, 30, 31, 24]. Ý tưởng chính của cách tiếp cận này là làm thế nào để chọn ra được các thuộc tính tốt nhất từ nhiều thuộc tính ứng viên để phân chia dần các đối tượng vào các cụm tại mỗi thời điểm. Vì vậy, điều quan trọng hàng đầu đối với phương pháp này là từ nhiều ứng viên có trong tập dữ liệu làm sao chọn được thuộc tính có thể phân chia các đối tượng một cách tốt nhất.

Công trình đầu tiên về gom cụm dữ liệu dựa trên tập thô được thực hiện bởi Mazlack và cộng sự. Trong [48], Mazlack và cộng sự đã đề xuất kỹ thuật sử dụng khái niệm gọi là *Tổng độ thô* (Total roughness TR), trong đó tổng độ thô càng cao thì độ chính xác của việc chọn thuộc tính phân cụm càng cao. Parmar và cộng sự [49] đã đề xuất thuật toán *min-roughness* (MMR), là thuật toán gom cụm phân cấp phân cấp dựa trên tập hợp thô "truyền thống" cho dữ liệu phân loại. Thuật toán MMR xác định thuộc tính phân cụm theo tiêu chí *Độ thô tối thiểu* (MR). Trong [31], Herawan và cộng sự đề xuất một kỹ thuật gọi là *Độ thuộc tính phụ thuộc tối đa* (MDA). MDA sử dụng hàm đo độ phụ thuộc trong lý thuyết tập thô để đánh giá mức độ phụ thuộc của một thuộc tính với các thuộc tính khác trong tập dữ liệu, từ đó chọn các thuộc tính phân cụm. Kỹ thuật MDA có thể được sử dụng để chọn các thuộc tính tạo ra các cụm, tuy nhiên nó không phải là một thuật toán gom cụm phân cấp hoàn chỉnh. Tiếp nối công trình của Herawan và cộng sự, một số nhà nghiên cứu khác cũng đã đề xuất các phương pháp mới để chọn thuộc tính phân cụm [22, 46, 50, 51, 52, 53]. Tuy

nhien, họ cũng chưa trình bày được một cách hoàn chỉnh, đầy đủ các thuật toán gom cụm cụ thể cũng như chưa đánh giá được hiệu quả thực tế của các kỹ thuật gom cụm dữ liệu phân loại của mình.

Trong [54] Qin và cộng sự đã đề xuất một thuật toán gom cụm phân cấp cho dữ liệu phân loại theo cách tiếp cận của lý thuyết tập thô nêu trên và sử dụng một số khái niệm của lý thuyết thông tin. Thuật toán này có tên gọi là MGR (Mean Gain Ratio). Trong MGR việc chọn thuộc tính phân cụm được thực hiện bằng cách sử dụng đại lượng *tỷ lệ gia tăng thông tin trung bình* (MGR) và sau đó sử dụng khái niệm *entropy của một cụm* để chọn ra một lớp trong số các lớp tương đương sinh bởi thuộc tính phân cụm để làm thành một cụm.

Gần đây, Trong [23], Wei và cộng sự đã phân tích một cách có hệ thống các *thuật toán gom cụm phân cấp* dựa trên tập thô hiện có cho dữ liệu phân loại và đưa ra một khung thống nhất. Khung thống nhất này bao gồm ba bước chính: (1) chọn một thuộc tính để phân hoạch nút cần tiếp tục chia cụm; (2) dựa trên thuộc tính đã chọn này, tạo ra một phép phân đôi nút cần tiếp tục phân chia; (3) xác định nút lá nào sẽ được phân chia thêm. Trong bước đầu tiên, một thuộc tính cho nhiều thông tin nhất được chọn để tạo ra các lớp tương đương ứng viên của nút cần phân chia. Trong bước thứ hai, một lớp tương đương ứng viên thích hợp được chọn để tạo thành một cụm từ các lớp ứng viên bằng một phương pháp đánh giá; các lớp tương đương còn lại được hợp lại làm thành nút cần tiếp tục phân chia. Ở bước thứ ba, một trong hai nút lá tạo ra từ phép phân đôi được chọn để phân chia tiếp trong vòng lặp tiếp theo. Việc áp dụng bước đầu tiên và bước thứ hai tạo ra một phép phân đôi nút cần phân cụm, do đó, có thể đạt được bất kỳ số cụm nhất định nào bằng cách thực hiện đệ quy quá trình phân đôi cụm.

4.3.1 Thuật toán lựa chọn thuộc tính gom cụm TR

Như đã trình bày ở trên, TR là công trình đầu tiên về gom cụm dữ liệu dựa trên tập thô, được thực hiện bởi Mazlack và cộng sự. Trong [48], Mazlack và cộng sự đã sử dụng khái niệm *Tổng độ thô* (Total roughness TR) để lựa chọn thuộc tính phân cụm, trong đó tổng độ thô càng cao thì độ chính xác của việc chọn thuộc tính phân cụm càng cao.

Định nghĩa 4.1. [48] Cho hệ thông tin $IS = (U, A)$, và $a_i, a_j \in A$, $U/a_i = \{X_1, X_2, \dots, X_h\}$. Độ thô trung bình của thuộc tính a_i đối với thuộc tính a_j , ký hiệu $Rough_{a_j}(a_i)$, được xác định bởi

$$Rough_{a_j}(a_i) = \frac{\sum_{k=1}^h R_{a_j}(X_k)}{|V(a_i)|} = \frac{\sum_{k=1}^h R_{a_j}(X_k)}{h}. \quad (4.1)$$

trong đó $R_{a_j}(X_k)$ là độ thô lớp tương đương X_k đối với a_j , nghĩa là (Định nghĩa 2.5. Chương 2):

$$R_{a_j}(X_k) = 1 - \frac{|a_j(X_k)|}{|\bar{a}_j(X_k)|}. \quad (4.2)$$

Định nghĩa 4.2. [48] Cho hệ thông tin $IS = (U, A)$ và $a_i \in A$. Tổng độ thô TR của a_i với mọi thuộc tính $a_j \in A$, với $i \neq j$ được xác định bởi

$$TR(a_i) = \frac{\sum_{(j=1) \wedge (i \neq j)}^{|A|} Rough_{a_j}(a_i)}{|A| - 1}. \quad (4.3)$$

Với các các định nghĩa trên, Thuật toán lựa chọn thuộc tính gom cụm TR là như sau

Thuật toán 4.1 Thuật toán TR (Total Roughness)

Đầu vào: Tập dữ liệu gom cụm (Hệ thông tin IS)

Đầu ra: Thuộc tính gom cụm.

Thuật toán:

Bước 1: Tính các lớp tương đương được sinh bởi quan hệ không phân biệt được trên mỗi thuộc tính.

Bước 2: Với mỗi thuộc tính a_i xác định độ thô trung bình $Rough_{a_j}(a_i)$ của nó đối với mỗi thuộc tính a_j , với $j \neq i$, theo công thức (4.1).

Bước 3: Với mỗi thuộc tính $a_i \in A$ tính độ thô toàn phần của nó với mọi thuộc tính a_j , với $i \neq j$, theo công thức (4.3).

Bước 4. Chọn thuộc tính a_i^* cho giá trị TR lớn nhất làm thuộc tính gom cụm, nghĩa là:

$$a_i^* = \operatorname{argmax}_{a_j \in A} \{TR(a_j)\}$$

4.3.2 Thuật toán lựa chọn thuộc tính gom cụm MDA

Trong [31], Herawan và cộng sự đã đề xuất một kỹ thuật lựa chọn thuộc tính phân cụm gọi là MDA (Maximum Dependency Attributes). MDA sử dụng độ phụ thuộc giữa các thuộc tính trong lý thuyết tập thô.

Định nghĩa 4.3. [31] Cho hệ thông tin $IS = (U, A)$, và $a_i, a_j \in A$, $U/a_i = \{X_1, X_2, \dots, X_h\}$. Độ phụ thuộc của thuộc tính a_i vào thuộc tính a_j , ký hiệu $\gamma_{a_j}(a_i)$, được định nghĩa như sau:

$$\gamma_{a_j}(a_i) = \frac{\sum_{k=1}^h |\underline{a_j X_k}|}{|U|}. \quad (4.4)$$

trong đó $\underline{a_j X_k}$ là a_j -xấp xỉ dưới của X_k .

Sử dụng định nghĩa 4.2 về độ phụ thuộc của một thuộc tính vào một thuộc tính khác trên đây, Herawan và cộng sự đã đề xuất thuật toán lựa chọn thuộc tính gom cụm MDA là như sau.

Thuật toán 4.2 Thuật toán MDA (Maximumdegree of Dependency of Attributes)

Đầu vào: Tập dữ liệu gom cụm (Hệ thông tin IS)

Đầu ra: Thuộc tính gom cụm.

Thuật toán:

Bước 1: Tính các lớp tương đương được sinh bởi quan hệ không phân biệt được trên mỗi thuộc tính.

Bước 2. Với mỗi thuộc tính a_i xác định độ phụ thuộc của thuộc tính a_i vào mỗi a_j , với $j \neq i$, theo công thức 4.4.

Bước 3. Chọn độ phụ thuộc lớn nhất $MD(a_i)$ của mỗi thuộc tính a_i ($a_i \in A$) như sau:

$$MD(a_i) = \max_{(a_j \in A) \wedge (i \neq j)} (\gamma_{a_j}(a_i))$$

Bước 4. Chọn thuộc tính a_i^ cho giá trị MD lớn nhất làm thuộc tính gom cụm, nghĩa là:*

$$a_i^* = \operatorname{argmax}_{a_j \in A} \{MD(a_j)\}$$

Trong [55], Hongwu Qin và cộng sự cho rằng các giá trị TR và MDA đều được xác định chủ yếu dựa vào số phần tử có trong xấp xỉ dưới của một thuộc tính đối với các thuộc tính khác, nên chúng thường chọn cùng một thuộc tính làm thuộc tính gom cụm trong hầu hết các trường hợp.

Cho đến nay, có hai trong những thuật toán gom cụm dựa vào lý thuyết tập thô và các khái niệm lý thuyết thông tin liên quan được cho là thành công nhất là thuật toán MMR (Minimum-Minimum Roughness) do Parmar và cộng sự đề xuất trong [49] và thuật toán MGR (Mean Gain Ratio) do Qin và các cộng sự đề xuất trong [54]. MMR và MGR là những thuật toán mạnh cho phép xử lý sự không chắc chắn trong quá trình gom cụm dữ liệu phân loại. Trong phần tiếp theo luận án sẽ trình bày về hai thuật toán cơ sở này và đề xuất một thuật toán mới gọi là MMNVI (Minimum Mean Normalized Variation of Information). Kết quả thực nghiệm cho thấy MMNVI là thuật toán gom cụm ổn định và tạo ra kết quả gom cụm tốt hơn hoặc ít ra cũng tương đương so với hai thuật toán cơ sở MMR và MGR.

4.3.3 Thuật toán MMR (Min-Min-Roughness)

Thuật toán MMR được đề xuất bởi Parmar và các cộng sự [49]. MMR là một thuật toán gom cụm hoàn chỉnh, thuộc loại phân cấp từ trên xuống. MMR được nhiều nhà nghiên cứu trích dẫn và xem nó là một thuật toán gom cụm tiên phong, thành công cho dữ liệu phân loại. MMR là một quá trình lặp không đảo ngược, thực hiện phân đôi dần tập U các

đối tượng ban đầu với mục tiêu đạt được một kết quả gom cụm tốt dần lên. MMR xác định thuộc tính gom cụm bằng cách sử dụng khái niệm độ thô (roughness) xác định như trong Định nghĩa 2.5, chương 2; điều này cho phép MMR có khả năng xử lý sự không chắc chắn trong quá trình gom cụm dữ liệu phân loại.

Định nghĩa 4.4. [49] Cho hệ thông tin $IS = (U, A)$ và $a_i \in A$. Độ thô cực tiểu (Min-Roughness - MR) của thuộc tính a_i là giá trị nhỏ nhất của độ thô trung bình của thuộc tính a_i đối với các thuộc tính khác trong A , nghĩa là:

$$MR(a_i) = \min_{(a_j \in A) \wedge (j \neq i)} \left(Rough_{a_j}(a_i) \right). \quad (4.4)$$

trong đó độ thô trung bình $Rough_{a_j}(a_i)$ được tính theo công thức (4.1).

Gọi $CDataset$ tập các đối tượng cần phân chia tiếp tại một vòng lặp nào đó với tập thuộc tính A . Với các định nghĩa trên, tại mỗi vòng lặp của quá trình gom cụm, thuật toán MMR thực hiện 6 bước sau.

Thuật toán 4.3 Thuật toán MMR (Min–Min–Mean–Roughness)

Đầu vào: Tập dữ liệu gom cụm (Hệ thông tin IS)

Đầu ra: Dữ liệu được gom cụm.

Thuật toán:

Bước 1: Với mỗi thuộc tính $a_i \in A$, tính phân hoạch của tập các đối tượng cần phân chia $CDataset$ sinh ra bởi a_i : $CDataset/a_i = \{X_1, X_2, \dots, X_g\}$.

Bước 2: Tính độ thô trung bình $Rough_{a_j}(a_i)$ của thuộc mỗi thuộc tính a_i đối với mỗi thuộc tính a_j khác theo công thức (4.1). Sau đó tính độ thô trung bình cực tiểu $MR(a_i)$ của thuộc mỗi tính a_i đối với các thuộc tính $a_j \in A, a_j \neq a_i$ theo công thức (4.5).

Bước 3: Chọn thuộc tính a_i^ cho giá trị MR nhỏ nhất làm thuộc tính gom cụm, nghĩa là:*

$$a_i^* = \operatorname{argmin}_{a_j \in A} \{MR(a_j)\}.$$

Bước 4. Tính phân hoạch $C\text{Dataset}/a^ = \{X_1, X_2, \dots, X_k\}$.*

Bước 5: Chọn lớp tương đương của phân hoạch $X_0 \in \text{Dataset}/a^$ thỏa mãn:*

$$X_0 = \operatorname{argmin}_{X_k} \left(\sum_{a_j \in A, a_j \neq a^*} R_{a_j}(X_k) \right)$$

và gán $X'_0 = C\text{Dataset} - X_0$.

Bước 6. Giữa hai tập đối tượng X_0 và X'_0 , chọn tập có số đối tượng nhỏ hơn làm một cụm và tập còn lại làm tập các đối tượng cần phân chia tiếp ở vòng lặp sau nếu chưa đạt được số cụm quy định.

Tại vòng lặp đầu tiên toàn bộ tập dữ liệu ban đầu là tập cần phân chia.

MMR được xem là một trong những thuật toán gom cụm sử dụng lý thuyết tập thành công. Tuy nhiên, *MMR* có hai hạn chế quan trọng sau đây [52, CT1]:

Thứ nhất, tại mỗi bước lặp, nó có xu hướng chọn thuộc tính có ít giá trị hơn. Đặc biệt, nếu một thuộc tính a thuộc A nào đó chỉ nhận một giá trị đơn lẻ, nó luôn được chọn làm thuộc tính phân hoạch. Điều này dẫn đến quá trình gom cụm bị dừng trước khi đạt được số cụm k cần thiết. Thật vậy, nếu một thuộc tính a chỉ nhận một giá trị đơn lẻ, thì $R_{a_j}(X) = 0$ với mọi $X \in U/Ind\{a\}$ và mọi $a_j \in A, a_j \neq a$. Do đó, giá trị *MR* của a luôn nhỏ nhất và nó sẽ được chọn làm thuộc tính phân hoạch. Tuy nhiên vì a chỉ nhận một giá trị duy nhất, quá trình phân đôi không thể tiếp tục thực hiện.

Thứ hai, thuật toán *MMR* chọn nút lá có nhiều đối tượng hơn để phân chia tiếp, do đó có thể tạo ra kết quả gom cụm không mong muốn.

4.3.4 Thuật toán MGR (Mean Gain Ratio)

Thuật toán MGR [54] được Qin và cộng sự đề xuất, đây là một thuật toán gom cụm phân cấp sử dụng các khái niệm lý thuyết thông tin liên quan với lý thuyết tập thô. Thuật toán MGR xác định thuộc tính gom cụm bằng cách sử dụng tỷ lệ lợi thông tin trung bình (the mean information gain ratio), điều này cho phép MGR có khả năng xử lý sự không chắc chắn trong quá trình gom cụm dữ liệu phân loại.

Định nghĩa 4.5. [54] Cho hệ thông tin $IS = (U, A)$ và $a_i, a_j \in A$, tỷ lệ lợi thông tin của a_i đối với a_j , ký hiệu bởi $IG_{a_j}(a_i)$, được xác định như sau

$$GR_b(a) = \frac{I(a; b)}{H(a)}. \quad (4.6)$$

trong đó $I(a; b)$ là thông tin tương hỗ giữa hai thuộc tính a và b , $H(a)$ là entropy của thuộc tính a (Định nghĩa 2.12 và 2.15 và Chương 2).

Định nghĩa 4.6. [54] Cho hệ thông tin $IS = (U, A)$ và $a_i, a_j \in A$. tỷ lệ lợi thông tin trung bình của a_i đối với mọi $a_j, j \neq i$, ký hiệu bởi $MGR(a_i)$, được xác định như sau

$$MGR(a_i) = \frac{1}{|A| - 1} \sum_{j=1, j \neq i}^{|A|} GR_{a_j}(a_i). \quad (4.7)$$

Cho tập dữ liệu gom cụm ($CDataset$) và A là tập các thuộc tính, trong mỗi vòng lặp của quá trình gom cụm, thuật toán MGR thực hiện các bước sau

Thuật toán 4.4 Thuật toán MGR (Mean Gain Ratio)

Đầu vào: Hệ thông tin $IS = (U, A, V, f)$

Đầu ra: Dữ liệu được gom cụm.

Thuật toán:

Bước 1: Với mỗi thuộc tính $a_i \in A$, tính phân hoạch của tập các đối tượng cần phân chia $CDataset$ sinh ra bởi a_i :

$$CDataset/a_i = \{X_1, X_2, \dots, X_g\}.$$

Bước 2: Tính tỷ lệ lợi thông tin của a_i đối với mỗi thuộc tính $a_j \in A, j \neq i_i$ theo công thức (4.6). Sau đó tính tỷ lệ lợi thông tin trung bình $MGR(a_i)$ của thuộc tính a_i đối với các thuộc tính $a_j \in A, j \neq i_i$ theo công thức (4.7).

Bước 3: Chọn thuộc tính a_i^ cho giá trị MGR nhỏ nhất làm thuộc tính gom cụm, nghĩa là:*

$$a^* = \operatorname{argmax}_{a_j \in A} \{MGR(a_j)\}.$$

Bước 4. Tính phân hoạch $CDataset/a^ = \{X_1, X_2, \dots, X_k\}$.*

Bước 5. Chọn lớp tương đương $X_0 \in Dataset/a^$ sao cho:*

$$X_0 = \operatorname{argmin}_{X_k \in CDataset/a^*} (Entropy(X_k)).$$

$$\text{lấy } X'_0 = CDataset - X_0.$$

Bước 6. Lấy tập đối tượng X_0 làm một cụm và tập X'_0 làm tập các đối tượng cần phân chia tiếp ở vòng lặp sau nếu chưa đạt được số cụm quy định.

Tại vòng lặp đầu tiên toàn bộ tập dữ liệu ban đầu là tập cần phân chia $CDataset$.

So sánh với MMR, thuật toán MGR có 2 ưu điểm [54]:

(1) MMG không có xu hướng chọn thuộc tính gom cụm có ít giá trị hơn;

(2) MMG cho ra một cụm trong mỗi vòng lặp dựa trên độ thuần nhất của các đối tượng trong mỗi cụm mà không quan tâm đến số lượng thành viên của cụm và thực hiện tiến trình gom cụm tiếp theo đối với tập các đối tượng còn lại, điều này hoàn toàn tự nhiên hơn so với thuật toán MMR.

Tuy nhiên, Wei và các cộng sự [23] đã chỉ ra rằng MGR có thể sẽ không phù hợp để lựa chọn thuộc tính đại diện cho việc phân tách cụm. Thật vậy, từ công thức (4.6) có thể thấy rằng độ lợi thông tin $GR_{a_j}(a_i)$ của thuộc tính a_i đối với thuộc tính a_j có thể sẽ rất lớn nếu cả entropy của a_i và thông tin tương hỗ $I(a_i; a_j)$ đều nhỏ. Nói cách khác, độ tương tự giữa hai thuộc tính a_i và a_j có thể sẽ thấp nếu $GR_{a_j}(a_i)$ cao. Như vậy, việc sử dụng MGR

để chọn thuộc tính phân chia cho quá trình gom cụm là không phù hợp. Đây là nhược điểm chính của thuật toán MGR.

Trong nỗ lực cải tiến chất lượng gom cụm dữ liệu, trên cơ sở đánh giá ưu nhược điểm của các thuật toán cơ sở, luận án đã đề xuất một thuật toán mới để gom cụm dữ liệu phân loại được gọi là MMNVI (Minimum Mean Normalized Variation of Information).

MMNVI sử dụng Biến thể thông tin chuẩn hóa trung bình của một thuộc tính đối với các thuộc tính khác để tìm ra thuộc tính gom cụm tốt nhất và entropy của các lớp tương đương được tạo bởi thuộc tính gom cụm được chọn để phân tách nhị phân tập dữ liệu gom cụm.

4.4 Đề xuất thuật toán MMNVI gom cụm dữ liệu phân loại

4.4.1 Ý tưởng và những định nghĩa cơ bản

Như chúng ta đã thấy trong Chương 2, trong hệ thống thông tin phân loại $IS = (U, A)$, mỗi thuộc tính trong A xác định duy nhất một phân hoạch (tức là một phép gom cụm) trên tập U các đối tượng. Một thuộc tính tạo ra một phân hoạch tốt tập các đối tượng nếu phân hoạch đó chia sẽ được nhiều thông tin nhất với các phân hoạch sinh ra bởi các thuộc tính khác trong A [27, 56]. Ngoài ra, entropy của tập dữ liệu càng nhỏ thì mức độ giống nhau của các các đối tượng trong đó càng cao. Trên cơ sở hai nhận xét này, trong thuật toán MMNVI, luận án đề xuất thực hiện bước 1 và bước 2 trong khung công việc của một thuật toán gom cụm phân cấp như sau: (1) tại mỗi vòng lặp, thuật toán sẽ chọn thuộc tính phân cụm là thuộc tính tạo ra phân hoạch gần nhất với các phân hoạch xác định bởi các thuộc tính khác; (2) trong phân hoạch được tạo bởi thuộc tính phân cụm đã chọn, chọn lớp tương đương có độ tương tự trong lớp cao nhất làm một cụm và lấy hợp của tất cả các lớp còn lại tạo thành tập dữ liệu cần chia cụm mới. Hai bước trên sẽ được lặp lại trên tập dữ liệu cần phân cụm mới cho đến khi số cụm thu được bằng số cụm mong muốn cho trước k .

Để đo khoảng cách giữa hai thuộc tính (hai phân hoạch), số liệu NVI (Biến thể thông tin chuẩn hóa) được được sử dụng.

Định nghĩa 4.7. Cho hệ thống thông tin $IS = (U, A)$ và $a_i \in A$. Biến thể thông tin chuẩn hóa trung bình giữa a_i với mỗi $a_j \in A, a_j \neq a_i$ được định nghĩa như sau:

$$MNVI(a_i) = \frac{1}{|A| - 1} \sum_{j=1, j \neq i}^{|A|} NVI(a_i, a_j). \quad (4.8)$$

trong đó $NVI(a_i, a_j)$ là độ đo biến thể thông tin chuẩn hóa giữa thuộc tính a_i với thuộc tính a_j được định nghĩa tại 2.16 Chương 2.

Vì $NVI(a, b)$ là một metric trong không gian các thuộc tính (điều này được chứng minh trong định lý 2.1, chương 2), nên $MNVI(a_i)$ được xem là khoảng cách trung bình giữa a_i với mỗi $a_j \in A, a_j \neq a_i$.

Định nghĩa 4.8 [54, 57] Cho hệ thống thông tin $IS = (U, A)$ và $A = \{a_1, a_2, \dots, a_p\}$. Giả sử các thuộc tính trong A là độc lập lẫn nhau, khi đó entropy của tập dữ liệu $X \subseteq U$, ký hiệu là $Entropy(X)$, được định nghĩa bởi:

$$Entropy(X) = H_X(a_1) + H_X(a_2) + \dots + H_X(a_p). \quad (4.9)$$

trong đó $H_X(a_i)$ là entropy trên X của thuộc tính $a_i, i = 1, \dots, p$, xác định bởi công thức (2.11, Chương 2). Entropy của X càng nhỏ thì các đối tượng trong X càng giống nhau. Do đó, entropy của một cụm đã được nhiều tác giả sử dụng làm thước đo độ tương tự của các đối tượng trong cùng một cụm [54, 57].

4.4.2 Thuật toán MMNVI

Với ý tưởng và định nghĩa trên, thuật toán MMNVI hoạt động như sau:

Ở vòng lặp đầu tiên, thuật toán lấy tất cả các đối tượng trong tập U làm tập dữ liệu cần gom cụm và tại mỗi vòng lặp thực hiện ba bước chính sau đây:

Bước 1. Loại bỏ tất cả các thuộc tính có giá trị đơn lẻ.

Bước 2. Chọn ra thuộc tính có giá trị MNVI nhỏ nhất làm thuộc tính gom cụm

Bước 3. Phân hoạch tập các đối tượng cần phân chia thành các lớp tương đương; lấy lớp tương đương có Entropy nhỏ nhất trở thành một cụm, đồng thời trả về tập dữ liệu cần phân cụm tiếp theo là tập chứa tất cả các tương đương còn lại.

Quá trình phân cụm trên được lặp lại cho đến khi số lượng nút lá thu được bằng với số lượng k cụm cho trước.

Thuật toán 4.5 Thuật MMNVI

Đầu vào: Tập dữ liệu gom cụm (Hệ thống tin) IS ; số cụm k cần gom.

Đầu ra: k cụm dữ liệu gom được.

Thuật toán:

Bước 1. $CNC = 1$ // Gán số cụm ban đầu bằng 1

$CDataset = U$ // $CDataset$ ký hiệu nút dữ liệu cần phân cụm.

Bước 2. $B = A$;

Với mọi $a_i \in A$

Tính phân hoạch $CDataset / Ind\{a_i\}$;

Nếu $|CDataset / Ind\{a_i\}| = 1$ // mọi cá thể trong $CNode$ có
// cùng một giá trị về a_i

$B = B - a_i$ // loại bỏ thuộc tính a_i

Hết nếu

Hết với mọi

Bước 3. Với mọi $a_i \in B$

Tính $MNVI\{a_i\}$ // sử dụng công thức 4.8

Hết với mọi

Bước 4. Xác định thuộc tính gom cụm a^* thỏa mãn

$$a^* = \operatorname{argmin}_{a_i \in B} MNVI(a_i)$$

Bước 5. Xác định phân hoạch $CDataset / Ind\{a^*\} = \{X_1, X_2, \dots, X_h\}$;

$$X = \operatorname{argmin}_{X_i} (Entropy(X_i))$$

// với $X_i \in CDataset / Ind\{a^*\}$

Bước 6. Trả về X là một cụm

$$CNC = CNC + 1;$$

Nếu $CNC < k$

$$CDataset = CDataset - X$$

Quay lại Bước 2;

Ngược lại: Trả về $Cdataset$ là một cụm;

Hết nếu;

Đối với thuật toán MMNVI có hai lưu ý cho sau đây:

(1) Ở bước 4, nếu có nhiều thuộc tính cùng cho giá trị MNVI nhỏ nhất, thuật toán sẽ chọn thuộc tính đầu tiên làm thuộc tính gom cụm.

(2) Ở bước 5 và 6, sau khi chọn được thuộc tính phân tách tập dữ liệu, MMNVI sẽ chọn lớp tương đương có Entropy thấp nhất làm một cụm và lấy hợp của các lớp tương đương còn lại làm tập dữ liệu cần phân cụm tiếp. Rõ ràng, entropy của các lớp tương đương còn lại sẽ lớn hơn entropy của lớp đã chọn, điều này được minh chứng bằng mệnh đề 4.1 dưới đây. Tuy nhiên, nếu có nhiều lớp tương đương có cùng giá trị Entropy thấp nhất, thuật toán sẽ ưu tiên chọn lớp có nhiều đối tượng nhất.

Mệnh đề 4.1. Cho một tập dữ liệu phân loại dưới dạng hệ thống thông tin $IS = (U, A)$, thuộc tính $a \in A$ và $U/IND\{a\} = \{X_1, X_2, \dots, X_m\}$. Nếu lớp $X_i \in U/a$ có entropy nhỏ nhất, nghĩa là nếu:

$$Entropy(X_i) = \min\{Entropy(X_1), Entropy(X_2), \dots, Entropy(X_h)\},$$

thì:

$$Entropy\left(\bigcup_{j \neq i} X_j\right) \geq Entropy(X_i).$$

Chứng minh: Không mất tính tổng quát, ta có thể giả thiết lớp X_1 là lớp có entropy nhỏ nhất. Đặt $U' = X_2 \cup X_3 \cup \dots \cup X_h = U \setminus X_1$.

Giả sử tập giá trị của a là $V_a = \{x_1, x_2, \dots, x_m\}$, b là một thuộc tính trong $b \in A - \{a\}$ có $V_b = \{y_1, y_2, \dots, y_n\}$ và $U/IND\{b\} = \{Y_1, Y_2, \dots, Y_n\}$, khi đó theo công thức (2.13) tính entropy có điều kiện tại Chương 2, ta có

$$H_{U'}(b|a) = - \sum_{i=2}^m P(a = x_i) \sum_{j=1}^n P(b = y_j | a = x_i) \log_2 P(b = y_j | a = x_i)$$

$$\begin{aligned}
&= \sum_{i=2}^m \frac{|X_i|}{|U'|} \left(- \sum_{j=1}^n P(b = y_j \mid a = x_i) \log_2 P(b = y_j \mid a = x_i) \right) \\
&= \sum_{i=2}^m \frac{|X_i|}{|U'|} H_{X_i}(b) .
\end{aligned}$$

Mặt khác, theo công thức (2.15) Chương 2, $H_{U'}(b) \geq H_{U'}(b|a)$. Vì vậy,

$$\begin{aligned}
H_{U'}(b) &\geq \sum_{i=2}^m \frac{|X_i|}{|U'|} H_{X_i}(b). \\
\sum_{b \in A - \{a\}} H_{U'}(b) &\geq \sum_{b \in A - \{a\}} \sum_{i=2}^m \frac{|X_i|}{|U'|} H_{X_i}(b) = \sum_{i=2}^m \frac{|X_i|}{|U'|} \sum_{b \in A - \{a\}} H_{X_i}(b) .
\end{aligned}$$

Lưu ý rằng, $H_{X_i}(a) = 0$ trên với mọi tương đương X_i , $i = 2, 3, \dots, m$, nên ta có

$$\sum_{b \in A - \{a\}} H_{X_i}(b) = \sum_{b \in A} H_{X_i}(b) .$$

Vậy

$$\sum_{b \in A} H_{U'}(b) \geq \sum_{i=1}^m \frac{|X_i|}{|U'|} \sum_{b \in A} H_{X_i}(b) = \sum_{i=1}^m \frac{|X_i|}{|U'|} Entropy(X_i) \geq Entropy(X_1) .$$

Vậy

$$Entropy(X_2 \cup X_3 \cup \dots \cup X_h) \geq Entropy(X_1) . \blacksquare$$

Ví dụ 4.1. Xét hệ thống thông tin gồm tám đối tượng với bảy thuộc tính phân loại cho ở bảng 4.1. Giả sử ta muốn phân các đối tượng vào 3 cụm. Ta có

$$U = \{1, 2, \dots, 8\};$$

$$A = \{\text{Học vị, Tiếng Anh, Kinh nghiệm, Tin học, Toán, Lập trình, Thống kê}\}$$

Tại bước lặp đầu tiên, MMNVI lấy cả 8 đối tượng của tập U làm tập dữ liệu gom cụm. Có thể thấy, cả bảy thuộc tính trên đều không có giá trị đơn lẻ, vì vậy không có thuộc tính nào bị loại bỏ. MMNVI xác định thuộc tính gom cụm tốt nhất ở bước lặp đầu tiên.

Bảng 4.1 Hệ thống tin về chất lượng đầu vào của sinh viên

Sinh viên	Bằng cấp a1	Tiếng Anh a2	Kinh nghiệm a3	Tin học a4	Toán a5	Lập trình a6	Thống kê a7
1	TS	Giỏi	TB	Giỏi	Giỏi	Giỏi	Giỏi
2	TS	TB	TB	Giỏi	Giỏi	Giỏi	Giỏi
3	ThS	TB	TB	TB	Giỏi	Giỏi	Giỏi
4	ThS	TB	TB	TB	Giỏi	Giỏi	TB
5	ThS	TB	TB	TB	TB	TB	TB
6	ThS	TB	TB	TB	TB	TB	TB
7	CN	TB	Cao	Giỏi	TB	TB	TB
8	CN	Kém	Cao	Giỏi	TB	TB	Giỏi

MMNVI xác định thuộc tính gom cụm tốt nhất ở bước lặp đầu tiên như sau:

Ta có:

$$U/Ind(\{a1\}) = \{X_1, X_2, X_3\} = \{\{1,2\}, \{3,4,5,6\}, \{7,8\}\}; \quad U/Ind(\{a2\}) \\ = \{Y_1, Y_2, Y_3\} = \{\{1\}, \{2,3,4,5,6,7\}, \{8\}\};$$

$$U/Ind(\{a1, a2\}) = \{Y_1, Y_2, Y_3\} = \{\{1\}, \{2\}, \{3,4,5,6\}, \{7\}, \{8\}\}.$$

$$H(a1) = 1.5, \quad H(a2) = 1.0613, \quad H(a1, a2) = 2, \quad I(a1; a2) = 0.5613.$$

$$\text{Vì vậy, } NVI(a1, a2) = 1 - I(a1; a2)/H(a1, a2) = 0.7194.$$

Tương tự, ta tính được biến thể thông tin chuẩn hóa của $a1$ với $a3, a4, a5, a6$, và $a7$.
Lần lượt là 0.4591, 0.3333, 0.7500, 0.7500, 0.8403.

Kết quả tính toán *biến thể thông tin chuẩn hóa* của mỗi thuộc tính ứng với các thuộc tính còn lại và *biến thể thông tin chuẩn hóa trung bình* (MNVI) của mỗi thuộc tính được thể hiện qua bảng 4.2.

Trong bảng 4.2, cột MNVI (cột cuối cùng) thể hiện giá trị *biến thể thông tin chuẩn hóa trung bình* của mỗi a_i ứng với $a_j \in A$ với $a_i \neq a_j$. Có thể thấy, thuộc tính $a1$ có MNVI nhỏ nhất và được chọn là thuộc tính gom cụm ở bước phân cụm đầu tiên. Phân hoạch của tập $C_{Dataset}$ là:

$$U/Ind(\{a1\}) = \{X_1, X_2, X_3\} = \{\{1,2\}, \{3,4,5,6\}, \{7,8\}\};$$

$$Entropy(X_1) = 1, \quad Entropy(X_2) = 2.8113, \quad Entropy(X_3) = 2.$$

Vì X_1 có entropy nhỏ nhất nên X_1 được chọn làm một cụm. Tập các đối tượng còn lại là $Z_1 = \{3,4,5,6,7,8\}$, sẽ tạo thành tập dữ liệu gom cụm ở bước phân tách thứ 2.

Bảng 4.2 Độ chắc chắn trung bình của các thuộc tính

Thuộc tính	NVI							MNVI
	a1	a2	a3	a4	a5	a6	a7	
a1		0.7194	0.4591	0.3333	0.7500	0.7500	0.8403	0.642
a2	0.7194		0.7910	0.8221	0.8620	0.8620	0.8221	0.8131
a3	0.4591	0.7910		0.7925	0.7925	0.7925	1.0000	0.7713
a4	0.3333	0.8221	0.7925		1.0000	1.0000	0.8958	0.8073
a5	0.7500	0.8620	0.7925	1.0000		0.0000	0.8958	0.7167
a6	0.7500	0.8620	0.7925	1.0000	0.0000		0.8958	0.7167
a7	0.8403	0.8221	1.0000	0.8958	0.8958	0.8958		0.8916

Ở bước phân tách thứ 2, có ba thuộc tính có cùng giá trị biến thể thông tin chuẩn hóa trung bình nhỏ nhất là $a1$, $a3$ và $a4$. Thuộc tính $a3$ được chọn làm thuộc tính gom cụm, ta có $Z_1/a3 = \{Y_1, Y_2\} = \{\{3,4,5,6\}, \{7,8\}\}$. Bởi vì, $Entropy(Y_1) = 2.8113 > Entropy(Y_2) = 2$, do đó, $Y_2 = \{7,8\}$ được chọn làm cụm thứ 2.

Với khai báo số cụm cần phân ban đầu là $k = 3$, thuật toán kết thúc với kết quả trả về gồm ba cụm dữ liệu: $C_1 = \{1,2\}$, $C_2 = \{7,8\}$ và $C_3 = \{3,4,5,6\}$.

4.4.3 Độ phức tạp của thuật toán MMNVI

Giả sử tập dữ liệu gom cụm có n đối tượng, m thuộc tính và k là số lượng cụm cần gom cho trước. Để chia tập dữ liệu vào k cụm, thuật toán sẽ có $k - 1$ bước lặp. Ở mỗi bước lặp, thuật toán MMNVI cần tính toán giá trị MNVI của tất cả m thuộc tính. Để tính MNVI của một thuộc tính bất kỳ, ta có thời gian xác định các lớp tương đương là n , và thời gian tính NVI giữa thuộc tính đó với các thuộc tính khác là $n(m - 1)$. Do đó, thời gian để tính toán giá trị MNVI của tất cả m thuộc tính là $m(n + n(m - 1)) = nm^2$. Ngoài ra, thuật toán MMNVI cần $n(k - 1)$ lần tính entropy của các lớp tương đương. Thời gian để tính

entropy của một lớp tương đương thì không lớn hơn m . Do đó, thời gian dành cho việc tính entropy sẽ ít hơn $(k - 1)nm$. Tổng cộng, độ phức tạp thời gian của MMNVI là đa thức, và là $O(knm^2)$.

4.4.4 Nhận xét thuật toán MMNVI

Tương tự MMR và MGR, thuật toán MMNVI có khả năng xử lý sự không chắc chắn trong quá trình gom cụm bằng cách sử dụng entropy thông tin để đo độ không chắc chắn của một tập đối tượng. MMNVI chỉ yêu cầu một dữ liệu đầu vào là số cụm cần gom cho trước k . MMNVI không phụ thuộc vào thứ tự của các đối tượng trong tập dữ liệu đầu vào và có thể cho kết quả gom cụm ổn định (cho cùng kết quả gom cụm với nhiều lần thực hiện thuật toán).

So sánh với MMR và MGR, thuật toán MMNVI có ba cải tiến chính:

(1) Trong mỗi vòng lặp, trước khi chọn thuộc tính gom cụm, MMNVI loại bỏ tất cả các thuộc tính có giá trị đơn lẻ, điều này giúp cho quá trình gom cụm không bị dừng quá sớm.

(2) Thuật toán MMNVI đánh giá các thuộc tính ứng viên bằng độ đo biến thể thông tin chuẩn hóa trung bình thay cho Min-Min-Roughness vì các phân vùng được tạo ra bởi một thuộc tính phải được phản ánh trên tất cả các thuộc tính sẽ hợp lý hơn thay vì chỉ được phản ánh bởi một thuộc tính tốt nhất.

(3) Thuật toán MMNVI chọn tập dữ liệu có entropy lớn hơn làm tập cần tiếp tục phân chia $CDataset$, do đó cải thiện được độ chính xác của kết quả gom cụm. Đây là ưu điểm vượt trội của MMNVI so với MMR, vì MMR chỉ biết chọn tập dữ liệu có nhiều đối tượng hơn để phân chia tiếp.

4.4.5 Kết quả thực nghiệm thuật toán MMNVI

Để đánh giá hiệu suất gom cụm, thuật toán gom cụm MMNVI được cài đặt bằng ngôn ngữ lập trình R và tính toán thực nghiệm trên các bộ dữ liệu thực tế. Bên cạnh đó, việc đánh giá còn được so sánh với hai thuật toán cơ sở là MMR và MGR.

4.4.5.1 Bộ dữ liệu đánh giá

Tính toán thực nghiệm của thuật toán MMNVI đã được thực hiện trên tám bộ dữ liệu chuẩn lấy từ kho dữ liệu chuẩn UCI bao gồm: Soybean small, Zoo, Votes, Breast Cancer Wisconsin, Mushroom, Balance Scale, Car Evaluation và Chess. Thông tin về các bộ dữ liệu được thể hiện trong bảng 4.3

Bảng 4.3 Tám bộ dữ liệu chuẩn UCI

CSDL	Tên viết tắt	Số đối tượng	Số thuộc tính	Số cụm
Soybean small	Soybean	47	359	4
Breast Cancer Wisconsin	Breast	683 (699)	9	2
Car Evalution	Car	1728	6	4
Vote	Vote	435	16	2
Chess	Chess	3196	36	2
Mushroom	Mushroom	8124	22	2
Balance scale	Balance	625	4	3
Zoo	Zoo	101	16	7

Trong bộ dữ liệu Breast Cancer Wisconsin có 16 đối tượng có giá trị thiếu, trong quá trình thực nghiệm đã tiến hành xóa các đối tượng này vì vậy số đối tượng còn lại là 683.

Cả ba thuật toán MMNVI, MMR và MGR đều được tiến hành thực nghiệm trên tất cả các tập dữ liệu. Mỗi thuật toán đều lấy số cụm cho trước của các đối tượng trong các tập dữ liệu làm tham số đầu vào. Ví dụ, đối với tập dữ liệu Car Evalution, số lượng cụm cho trước là 4.

4.4.5.2 Phương pháp đánh giá hiệu suất

Để đánh giá kết quả gom cụm một cách khách quan, ba tiêu chí thường được các nhà nghiên cứu sử dụng rộng rãi bao gồm: độ thuần khiết tổng thể (Overall Purity), chỉ số ngẫu nhiên hiệu chỉnh (Adjusted Rand Index - ARI), thông tin tương hỗ chuẩn hóa (Normalized Mutual Information - NMI).

Giả sử rằng mỗi tập dữ liệu có n đối tượng, $\Omega = \{\omega_1, \omega_2, \dots, \omega_I\}$ và $C = \{c_1, c_2, \dots, c_J\}$ đại diện cho kết quả gom cụm của thuật toán và số cụm dữ liệu thực tế tương ứng, cụm ω_i có a_i đối tượng, lớp c_j có b_j đối tượng, và n_{ij} là số đối tượng có trong cả cụm ω_i và lớp c_j . Trên cơ sở đó, ta có bảng dự phòng được cho ở bảng 4.4.

Bảng 4.4 Bảng dự phòng

$\Omega \setminus C$	c_1	c_2	...	c_J	$sums$
ω_1	n_{11}	n_{12}	...	n_{1J}	a_1
ω_2	n_{21}	n_{23}	...	n_{2J}	a_2
...	...	...	...	...	...
ω_I	n_{I1}	n_{I2}	...	n_{IJ}	a_I
$sums$	b_1	b_2	...	b_J	$\sum_{ij} n_{ij} = n$

Với những ký hiệu trên, ba tiêu chí đánh giá được định nghĩa như sau:

(1) Độ thuần khiết tổng thể [54, 58]:

Độ thuần khiết của cụm ω_i được định nghĩa là tỷ số giữa số đối tượng trong ω_i có nhãn lớp chiếm đa số:

$$Purity(\omega_i) = \frac{1}{n_i} \max_j(n_{ij}). \quad (4.10)$$

Độ thuần khiết tổng thể của kết quả gom cụm được định nghĩa là tỷ lệ các đối tượng được phân lớp đúng trong số tất cả các đối tượng có trong tập dữ liệu, nghĩa là

$$OP = \frac{\sum_{i=1}^k \max_j(n_{ij})}{n}. \quad (4.11)$$

Độ thuần khiết tổng thể có miền giá trị nằm trong khoảng $[0,1]$. Độ thuần khiết tổng thể càng cao thì chất lượng của kết quả gom cụm càng tốt. Một phép gom cụm là hoàn hảo sẽ cho giá trị độ thuần khiết tổng thể bằng 1.

(2) Chỉ số ngẫu nhiên hiệu chỉnh (The Adjusted Rand Index – ARI) [54, 58]:

$$ARI(\Omega, C) = \frac{\sum_{i,j} C_{n_{ij}}^2 - \left[\sum_i C_{a_i}^2 \sum_j C_{b_j}^2 \right] / C_n^2}{\frac{1}{2} \left[\sum_i C_{a_i}^2 + \sum_j C_{b_j}^2 \right] - \left[\sum_i C_{a_i}^2 \sum_j C_{b_j}^2 \right] / C_n^2} \quad (4.12)$$

trong đó $C_{a_i}^2 = a_i(a_i - 1)/2$, $C_{b_j}^2 = b_j(b_j - 1)/2$, $C_{n_{ij}}^2 = n_{ij}(n_{ij} - 1)/2$.

Chỉ số ngẫu nhiên hiệu chỉnh có giá trị nằm giữa 0 và 1. Một kết quả gom cụm và chất lượng các cụm được xem là hoàn hảo khi chỉ số ngẫu nhiên hiệu chỉnh có giá trị bằng 1.

(3) Thông tin tương hỗ chuẩn hóa (Normalized Mutual Information - NMI) được xây dựng dựa trên nền tảng lý thuyết thông tin cơ bản. Cho hai phép gom cụm Ω và C , entropy, entropy đồng thời và thông tin tương hỗ của chúng được xác định một cách tự nhiên thông qua phân phối biên duyên, phân phối đồng thời và thông tin tương hỗ [54, 58]:

$$H(\Omega) = - \sum_{i=1}^I \frac{a_i}{n} \log_2 \frac{a_i}{n} ;$$

$$H(\Omega, C) = - \sum_{i=1}^I \sum_{j=1}^J \frac{n_{ij}}{n} \log_2 \frac{n_{ij}}{n} ;$$

$$I(\Omega; C) = H(\Omega) + H(C) - H(\Omega, C)$$

Khi đó, độ đo Thông tin tương hỗ chuẩn hóa được xác định theo công thức [54, 58]:

$$NMI = \frac{2I(\Omega; C)}{H(\Omega) + H(C)} \quad (4.13)$$

NMI có giá trị nằm trong khoảng [0,1], nó có giá trị bằng 1 khi hai phép gom cụm phụ thuộc nhau và bằng 0 khi hai phép gom cụm là độc lập, nghĩa là không chia sẻ thông tin với nhau.

4.4.5.3 Kết quả gom cụm

Trước tiên, luận án trình bày kết quả gom cụm thu được bởi thuật toán đề xuất MMNVI, sau đó trình bày các kết quả tính toán độ thuần khiết tổng thể (Overall Purity), chỉ số ngẫu nhiên hiệu chỉnh (Adjusted Rand Index - ARI) và thông tin tương hỗ chuẩn hóa

(Normalized Mutual Information – NMI thu được bởi ba thuật toán MMR, MGR và MMNVI trên 8 bộ dữ liệu.

Đối với tập dữ liệu Soybean, vì số nhãn lớp đã biết trong tập dữ liệu này là 4, do đó, số cụm thực nghiệm được quy định là 4 cho cả ba thuật toán MMNVI, MMR và MGR. Số đối tượng của tập dữ liệu Soybean phân bố theo theo các cụm và các lớp được trình bày trong bảng 4.5. Thuật toán MMNVI thu được kết quả gom cụm với 47 đối tượng có nhãn lớp chiếm đa số. Do đó, Độ thuần khiết tổng thể (Overall Purity– OP) của các cụm bằng 1. Chỉ số ngẫu nhiên hiệu chỉnh và thông tin tương hỗ chuẩn hóa lần lượt bằng 0,4601 và 0,6511.

Bảng 4.5 Kết quả gom cụm MMNVI trên tập dữ liệu Soybean Small

$\Omega \backslash C$	Lớp 1	Lớp 2	Lớp 3	Lớp 4	CP	OP	ARI	NMI
1	10	0	0	0	1	1	0.4601	0.6511
2	10	0	0	0	1			
3	10	0	0	0	1			
4	0	0	0	17	1			

Bảng 4.6 – 4.12 cho kết quả gom cụm của thuật toán MMNVI trên 7 bộ dữ liệu UCI còn lại.

Bảng 4.6 Kết quả gom cụm MMNVI trên tập dữ liệu Breast Cancer Wisconsin.

$\Omega \backslash C$	Lớp 1	Lớp 2	CP	OP	ARI	NMI
1	369	4	0.9893	0.8843	0.59	0.5446
2	75	235	0.7581			

Bảng 4.7 Kết quả gom cụm MMNVI trên tập dữ liệu Car Evaluation.

$\Omega \backslash C$	Lớp 1	Lớp 2	Lớp 3	Lớp 4	CP	OP	ARI	NMI
1	108	0	324	0	0.75	0.7384	0.0071	0.0452
2	115	23	268	26	0.6204			
3	108	0	324	0	0.75			
4	72	0	360	0	0.8333			

Bảng 4.8 Kết quả gom cụm MMNVI trên tập dữ liệu Vote.

$\Omega \backslash C$	Lớp 1	Lớp 2	CP	OP	ARI	NMI
1	200	8	0.9615	0.8276	0.4279	0.4009
2	67	160	0.7048			

Bảng 4.9 Kết quả gom cụm MMNVI trên tập dữ liệu Chess.

$\Omega \backslash C$	Lớp 1	Lớp 2	CP	Overall purity	ARI	NMI
1	922	895	0.5074	0.5307	0.0036	0.0034
2	605	774	0.5613			

Bảng 4.10 Kết quả gom cụm MMNVI trên tập dữ liệu Mushroom.

$\Omega \backslash C$	Lớp 1	Lớp 2	CP	OP	ARI	NMI
1	0	36	1	0.5224	-0.0011	0.009
2	4208	3880	0.5203			

Bảng 4.11 Kết quả gom cụm MMNVI trên tập dữ liệu Balance Scale

$\Omega \backslash C$	Lớp 1	Lớp 2	Lớp 3	CP	OP	ARI	NMI
1	10	17	98	0.784	0.6784	0.1234	0.1346
2	10	17	98	0.784			
3	28	228	119	0.608			

Bảng 4.12 Kết quả gom cụm MMNVI trên tập dữ liệu Zoo

$\Omega \backslash C$	Lớp 1	Lớp 2	Lớp 3	Lớp 4	Lớp 5	Lớp 6	Lớp 7	CP	OP	ARI	NMI
1	41	0	0	0	0	0	0	1	0.7327	0.3211	0.3707
2	1	0	0	0	0	0	0	1			
3	9	6	0	2	1	2	0	0.45			
4	11	3	0	4	2	1	1	0.5			
5	3	0	0	3	0	0	0	0.5			
6	1	0	0	0	0	0	0	1			
7	0	0	0	0	0	8	2	0.8			

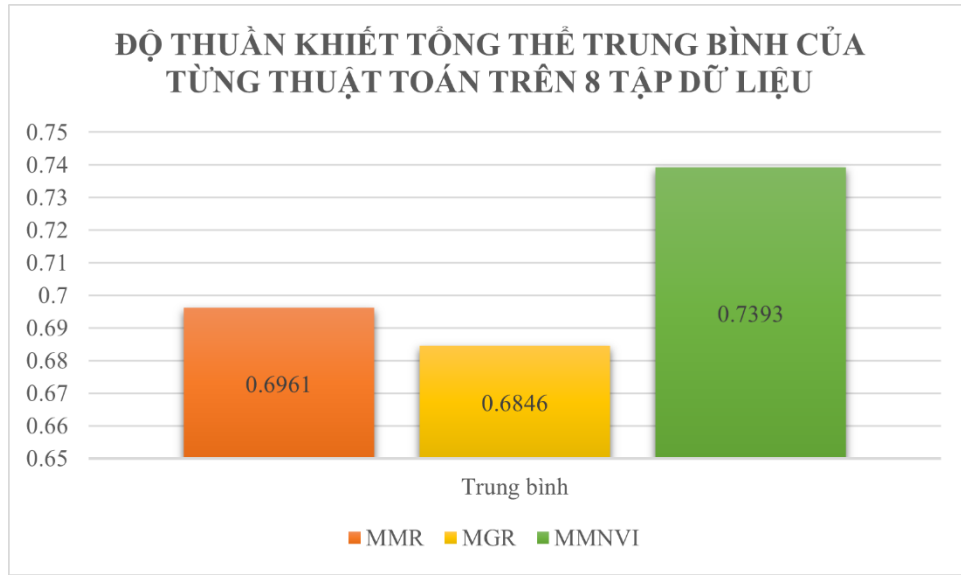
4.4.5.4 So sánh MMNVI với thuật toán MMR và MGR

Với quy trình tương tự, luận án áp dụng MMR và MGR cho 8 bộ dữ liệu trên. Độ tinh khiết tổng thể của ba thuật toán được tóm tắt trong Bảng 4.13.

Bảng 4.13 Độ thuần khiết tổng thể của 3 thuật toán trên 8 bộ dữ liệu.

Tập DL\ Thuật toán	MMR	MGR	MMNVI
Soybean	0.8298	1	1
Breast	0.6559	0.5	0.8843
Car	0.7002	0.6998	0.7384
Votes	0.6138	0.5	0.8276
Chess	0.5225	0.5338	0.5307
Mushroom	0.7002	0.6775	0.5224
Balance	0.6352	0.6352	0.6784
Zoo	0.9109	0.9307	0.7327
Trung bình	0.6961	0.6846	0.7393

Trong số 8 bộ dữ liệu thực nghiệm, MMNVI có độ thuần khiết tổng thể cao nhất trong năm tập dữ liệu, cụ thể là trên các tập Soybean Small, Breast Cancer Wisconsin, Car evaluation, Votes và Balance scale. MMR có độ thuần khiết tổng thể cao nhất đối với tập Mushroom. MGR có độ thuần khiết tổng thể cao nhất ở Soybean Small, Chess, và Zoo. Dòng cuối cùng của Bảng 4.13 hiển thị độ thuần khiết tổng thể trung bình của từng thuật toán trên 8 tập dữ liệu. Có thể thấy MMNVI đạt độ tinh khiết tổng thể trung bình cao nhất. Điều này được thể hiện qua hình 4.1.

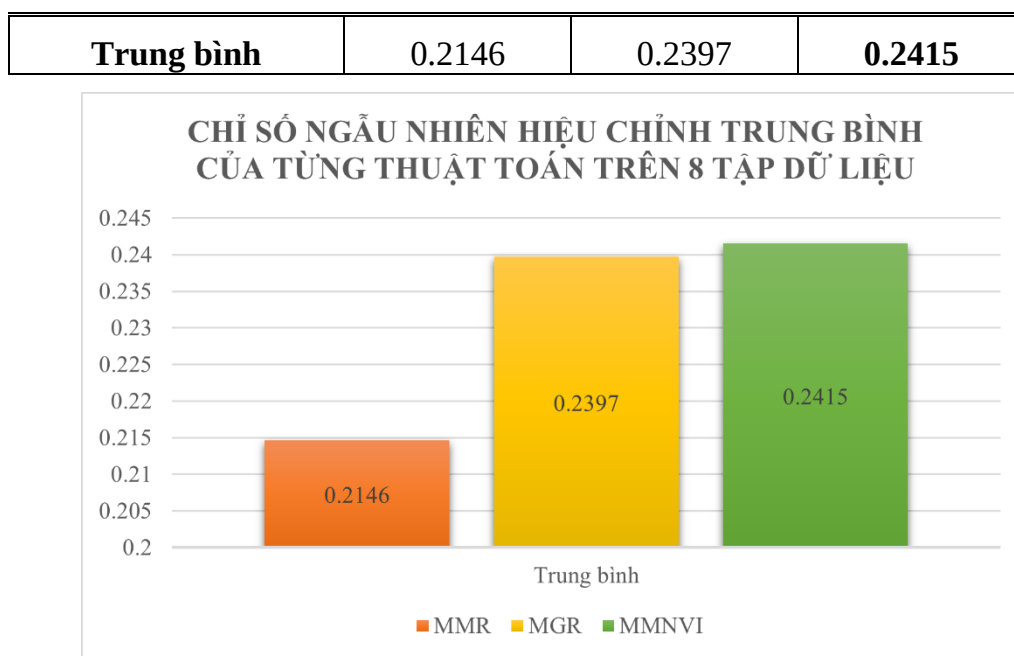


Hình 4.1 Hình minh họa so sánh độ thuần khiết tổng thể của ba thuật toán trên tám tập dữ liệu thực nghiệm

Chỉ số ngẫu nhiên hiệu chỉnh (ARI) của ba thuật toán được tóm tắt trong Bảng 4.14. Kết quả cho thấy, MMNVI cũng có giá trị ARI cao nhất trên bốn tập dữ liệu Breast Cancer Wisconsin, Votes, Chess và Balance. MMR có giá trị ARI thấp nhất trên tất cả 8 tập dữ liệu. MGR có giá trị ARI cao nhất ba tập Car evaluation, Mushroom và Zoo. Dòng cuối cùng của Bảng 4.14 hiển thị ARI trung bình của từng thuật toán trên 8 tập dữ liệu. Thuật toán MMNVI cũng đạt giá trị ARI trung bình cao nhất (Hình 4.2).

Bảng 4.14 Chỉ số ngẫu nhiên hiệu chỉnh (ARI) của ba thuật toán trên 8 tập dữ liệu.

Tập DL\ Thuật toán	MMR	MGR	MMNVI
Soybean	0.6738	0.4601	0.4601
Breast	0.0101	0.1465	0.59
Car	0.0129	0.0129	0.0071
Votes	- 0.0068	0.106	0.4279
Chess	0.0004	0.0036	0.0036
Mushroom	0.0129	0.1254	- 0.0011
Balance	0.1011	0.1011	0.1234
Zoo	0.913	0.9617	0.3211



Hình 4.2 Hình minh họa so sánh chỉ số ngẫu nhiên hiệu chỉnh trung bình của ba thuật toán trên tám tập dữ liệu thực nghiệm

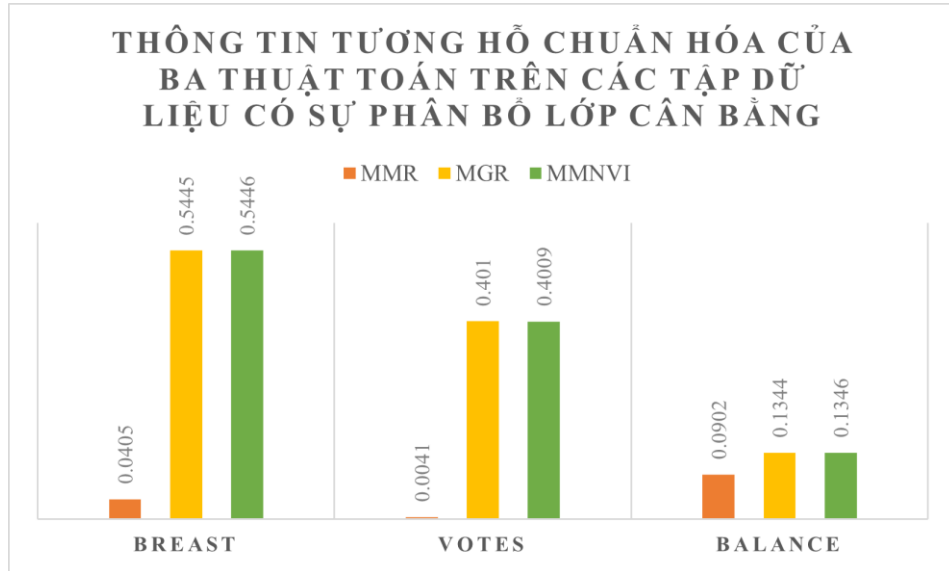
Bảng 4.15 Thông tin tương hỗ chuẩn hóa (NMI) của ba thuật toán trên 8 tập dữ liệu.

Tập DL\ Thuật toán	MMR	MGR	MMNVI
Soybean	0.8264	0.6511	0.6511
Breast	0.0405	0.5445	0.5446
Car	0.0621	0.0481	0.0452
Votes	0.0041	0.401	0.4009
Chess	0.0052	0.017	0.0034
Mushroom	0.0621	0.0246	0.009
Balance	0.0902	0.1344	0.1346
Zoo	0.913	0.9617	0.3707
Trung bình	0.3707	0.3478	0.2699

Thông tin tương hỗ chuẩn hóa (NMI) của ba thuật toán được tóm tắt trong Bảng 4.15. MMNVI có giá trị NMI cao nhất trên ba bộ dữ liệu Breast Cancer Wisconsin, Votes và Balance scale. MMR có NMI cao nhất trong Soybean Small, Car evaluation và Mushroom.

MGR có NMI cao nhất trong Chess và Zoo. Cột cuối cùng của Bảng 4.15 hiển thị NMI trung bình của từng thuật toán trên 8 bộ dữ liệu.

Một chú ý quan trọng là được thể hiện qua hình 4.3 là MMNVI hoạt động tốt hơn nhiều so với 2 thuật toán còn lại trên Breast Cancer, Votes và Balance scale. Breast Cancer và Votes là các tập dữ liệu có sự phân bố lớp cân bằng. MGR hoạt động tốt hơn nhiều so với các thuật toán khác trên tập Zoo.



Hình 4.3 Hình minh họa so sánh thông tin tương hỗ chuẩn hóa của ba thuật toán đối với các tập dữ liệu có sự phân bố lớp cân bằng

Kết quả thực nghiệm cho thấy, MMNVI cho thấy là thuật toán cho kết quả gom cụm tốt hơn hoặc ít ra là tương đương so với thuật toán MMR và MGR.

4.5 Kết luận chương 4

Gom cụm là một kỹ thuật quan trọng trong khai phá dữ liệu, và được ứng dụng trong nhiều vấn đề như phân loại động vật, thực vật, phân đoạn thị trường, phân loại khách hàng, văn bản, trang web v.v. Cho đến nay đã có nhiều phương pháp gom cụm đã được đề xuất. Tuy nhiên, theo các nghiên cứu, chưa có một phương pháp gom cụm tổng quát nào có thể giải quyết trọn vẹn cho tất cả các dạng cấu trúc cụm dữ liệu.

Gom cụm dữ liệu phân loại đặt ra nhiều thách thức hơn gom cụm dữ liệu số liên tục. Một số thuật toán gom cụm dữ liệu phân loại đã được đề xuất. Mặc dù những thuật toán

này có những đóng góp quan trọng cho việc giải quyết bài toán gom cụm dữ liệu phân loại, nhưng chúng không được thiết kế để xử lý sự không chắc chắn trong quá trình gom cụm. Xử lý sự không chắc chắn trong quá trình gom cụm là một vấn đề quan trọng, bởi vì trong nhiều ứng dụng thực tế thường không có ranh giới rõ ràng giữa các cụm.

Trong chương này, luận án tập trung nghiên cứu kỹ thuật gom cụm dữ liệu phân loại cho phép xử lý sự không chắc chắn trong quá trình gom cụm bằng cách sử dụng Lý thuyết tập thô kết hợp với các khái niệm entropy trong Lý thuyết thông tin, Trên cơ sở nghiên cứu các thuật toán cơ sở đã được đề xuất bởi các nhà nghiên cứu, tìm ra các thiếu sót, luận án đã đề xuất một thuật toán gom cụm phân cấp mới cho dữ liệu phân loại với tên gọi MMNVI. Kết quả thử nghiệm trên các tập dữ liệu thực tế lấy từ kho dữ liệu UCI cho thấy thuật toán MMNVI là một thuật toán ổn định, cho kết quả gom cụm tốt hơn hoặc ít ra là tương đương so với các thuật toán cơ sở. MMNVI là thuật toán có thể được sử dụng thành công trong việc gom cụm dữ liệu phân loại. Kết quả nghiên cứu này đã được công bố tại công trình [CT3] được đăng trên tạp chí *Journal of Computer Science and Cybernetics* năm 2023.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Khám phá tri thức từ CSDL là một lĩnh vực khoa học nhằm nghiên cứu để tạo ra những công cụ khai phá những thông tin, tri thức hữu ích, tiềm ẩn mang tính dự đoán trong các CSDL lớn. Tuy nhiên, với tốc độ tăng trưởng nhanh của dữ liệu ngày nay, việc nghiên cứu và ứng dụng các kỹ thuật khai phá dữ liệu còn gặp phải nhiều khó khăn, thách thức.

Một trong những khó khăn, thách thức là do sự phát triển nhanh chóng của công nghệ, kích thước của những tập dữ liệu con người thu thập được ngày càng lớn. Điều này có thể dẫn đến các thuật toán khai thác hoặc học từ dữ liệu truyền thống trở nên chậm lại và không thể xử lý thông tin một cách hiệu quả. Vấn đề đặt ra là trước khi thực hiện thuật toán khai thác dữ liệu cần phải có các phương pháp lựa chọn thuộc tính của cơ sở dữ liệu mà vẫn bảo toàn được những thông tin cần khai thác.

Bên cạnh đó, ngày nay trong các ứng dụng thực tiễn chúng ta thường gặp phải những cơ sở dữ liệu với các dạng cấu trúc cụm dữ liệu khác nhau. Việc khai phá những cơ sở dữ liệu này cũng là một thách thức.

Lý thuyết tập thô do Pawlak đề xuất, là một công cụ toán học mạnh để xử lý dữ liệu mơ hồ, không chính xác, không đầy đủ và không chắc chắn. Lý thuyết này đã được ứng dụng thành công trong khám phá tri thức trong cơ sở dữ liệu, cũng như trong học máy, hệ chuyên gia, nhận dạng mẫu.

Luận án tập trung vào việc ứng dụng Lý thuyết tập thô nhằm giải quyết hai vấn đề: (1) nghiên cứu thuật toán hiệu quả tìm tập rút gọn thuộc tính trong một bảng quyết định; (2) kỹ thuật gom cụm dữ liệu phân loại cho phép xử lý sự không chắc chắn trong quá trình gom cụm.

5.1 Những kết quả và đóng góp chính của luận án

(1) Bằng việc nghiên cứu các thuật toán đã được đề xuất bởi các nhà nghiên cứu, tìm ra các thiếu sót, luận án đã đề xuất một thuật toán mới lựa chọn thuộc tính trong một bảng quyết định dựa trên gom cụm. Thuật toán đề xuất, có tên gọi ACBRC, gồm 3 công đoạn:

- (i) Loại bỏ các thuộc tính không liên quan.

(ii) Gom các thuộc tính có liên quan thành một số cụm thích hợp bằng phương pháp gom cụm phân hoạch xung quanh Medoids PAM, kết hợp với một metric đặc biệt trong không gian thuộc tính là Biến thể Thông tin Chuẩn hóa.

(iii) Từ mỗi cụm thuộc tính gom được chọn ra một thuộc tính đại diện là thuộc tính có độ liên quan lớn nhất với thuộc tính quyết định. Các thuộc tính đại diện tạo thành một tập rút gọn xấp xỉ gồm những thuộc tính liên quan và không dư thừa.

Kết quả thử nghiệm trên các tập dữ liệu thực tế lấy từ kho dữ liệu UCI cho thấy thuật toán đề xuất ACBRC là rất khả quan trong việc làm giảm số thuộc tính trong các bảng quyết định, đồng thời nâng cao được độ chính xác phân lớp.

(2) Bằng việc nghiên cứu các thuật toán cơ sở đã được đề xuất bởi các nhà nghiên cứu, phân tích các thiếu sót, luận án đã đề xuất thuật toán gom cụm dữ liệu phân loại MMNVI theo phương pháp phân cấp. Thuật toán MMNVI sử dụng cách tiếp cận của Lý thuyết tập thô kết hợp với các khái niệm entropy trong Lý thuyết thông tin. MMNVI gồm 3 công đoạn:

(i) MMNVI loại bỏ tất cả các thuộc tính có giá trị đơn lẻ.

(ii) Chọn ra thuộc tính có giá trị MNVI nhỏ nhất làm thuộc tính gom cụm

(iii) Phân hoạch tập các đối tượng cần phân chia thành các lớp tương đương; lấy lớp tương đương có Entropy nhỏ nhất trở thành một cụm, đồng thời trả về tập dữ liệu cần phân cụm tiếp theo là tập chứa tất cả các tương đương còn lại.

Quá trình gom cụm trên được lặp lại cho đến khi thu được số cụm bằng số lượng cụm k ấn định trước.

Kết quả thử nghiệm trên các tập dữ liệu thực tế lấy từ kho dữ liệu UCI cho thấy thuật toán MMNVI là một thuật toán ổn định, cho kết quả gom cụm tốt hơn hoặc ít ra là tương đương so với các thuật toán cơ sở. MMNVI là thuật toán có thể được sử dụng thành công trong việc gom cụm dữ liệu phân loại.

5.2 Hướng phát triển của luận án

Đối với bài toán lựa chọn thuộc tính, tiếp tục nghiên cứu:

- Giải pháp tìm tập thuộc tính rút gọn trong bảng quyết định có thông tin bị thiếu;
- Ứng dụng thuật toán lựa chọn thuộc tính ACBRC vào phân loại văn bản.

Đối với bài toán gom cụm dữ liệu, tiếp tục nghiên cứu:

- Cải tiến thuật toán gom cụm MMNVI để nó có khả năng tự động xác định số lượng cụm thích hợp cho tập dữ liệu, thay vì phải chỉ định trước số lượng cụm cần gom. Chẳng hạn, chúng ta có thể để thuật toán MMNVI dừng quá trình lặp khi entropy của tất cả các nút lá thấp hơn một giá trị ngưỡng thích hợp;
- Phát triển MMNVI để có thể xử lý cả dữ liệu số và dữ liệu phân loại;
- Nghiên cứu cải thiện độ phức tạp tính toán.

DANH MỤC CÔNG TRÌNH ĐÃ CÔNG BỐ

Tạp chí quốc tế

[CT1] Do Si Truong, Nguyen Thanh Tung, Lam Thanh Hien, Improved minimum-minimum roughness algorithm for clustering categorical data, *International Journal of Advanced and Applied Sciences*, 8(10), Pages 43-50, 2021.

<https://doi.org/10.21833/ijaas.2021.10.006>

Tạp chí trong nước

[CT2] Do Si Truong, Lam Thanh Hien, Nguyen Thanh Tung, An effective algorithm for computing reducts decision table. *Journal of Computer Science and Cybernetics*, V.38, N.3 (2022), Pages 277–292, 2022.

DOI: <https://doi.org/10.15625/1813-9663/38/3/17450>.

[CT3] Do Si Truong, Lam Thanh Hien, Nguyen Thanh Tung, A new information theory based algorithm for clustering categorical data. *Journal of Computer Science and Cybernetics*, V.39, N.3 (2023), Pages 259-278, 2023.

DOI: <https://doi.org/10.15625/1813-9663/18568>

Hội thảo khoa học trong nước

[CT4] Pham Cong Xuyen, Do Si Truong, Nguyen Thanh Tung, An information-Theoretic metric based method for selecting clustering attribute. *Kỷ yếu Hội nghị Khoa học Quốc gia lần thứ IX “Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR’9)”*; Cần Thơ, ngày 4-5/8/2016, trang 31-40, 2016. DOI: 10.15625/vap.2016.0005.

[CT5] Nguyễn Minh Huy, Đỗ Sĩ Trường, Nguyễn Thanh Tùng, Về hàm đo độ phụ thuộc thuộc tính suy rộng. *Kỷ yếu Hội thảo quốc gia lần thứ XX: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông*, Quy Nhơn, 23-24/11/2017, trang 396-403, 2017.

[CT6] Đỗ Sĩ Trường, Trần Thanh Phương, Phạm Công Xuyên, Nguyễn Thanh Tùng, Thuật toán MMR cải tiến gom cụm dữ liệu phân loại, *Kỷ yếu Hội thảo “Fundamental and Applied IT Research 12/2021 (FAIR XIV)”*, Trường Đại học Công nghiệp Thực phẩm Tp.HCM, 2021.

TÀI LIỆU THAM KHẢO

- [1] J. Han, J. Pei, H. Tong, "Data Mining: Concepts and Techniques," *Morgan Kanufmann*, vol. 3th Edition, 2012.
- [2] A. Skowron, S. Dutta, "Rough sets: past, present, and future," *Natural computing*, vol. 17(4), pp. 855-876, 2018.
- [3] G. Chandrashekar, F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40 (1), no. 40th-year commemorative issue, p. 16 – 28, 2014.
- [4] D. Harris, A. V. Niekerk, "Feature clustering and ranking for selecting stable features from high dimensional remotely sensed data.," *International Journal of Remote Sensing*, 39(23), p. 8934–8949, 2018.
- [5] X. Zhu, Y. Wang, Y. Li, Y. Tan, G. Wang, Q. Song, "A new unsupervised feature selection algorithm using similarity-based feature clustering," *Computational Intelligence*, vol. 35 (1), p. 2–22, 2019.
- [6] S. Mesakar, M. S. Chaudhari, "Review Paper On Data Clustering Of Categorical Data," *International Journal of Engineering Research & Technology*, vol. 1, no. 10, December, 2012.
- [7] S. Naouali, S. B. Salem, Z. Chtourou, "Clustering Categorical Data: A Survey," *International Journal of Information Technology & Decision Making*, Vols. Vol. 19, No. 01, pp. 49-96, 2020.
- [8] Z. Pawlak, "Rough Sets - Theoretical Aspects of Reasoning about Data," *Kluwer Academic Publishers, Dordrecht*, 1991.
- [9] P. Pieta, T. Szmuc, "Applications of rough sets in big data analysis: An overview," *International Journal of Applied Mathematics and Computer Science*, vol. 31, no. 4, p. 659–683, 2021.
- [10] Q. Zhang, Q. Xie, G. Wang, "A survey on rough set theory and its applications," *CAAI Transactions on Intelligence Technology*, 1, pp. 323-333, 2016.
- [11] R. Bello, R. Falcon, "Rough sets in machine learning: a review," *Thriving Rough Sets*, pp. 87-118, 2017.
- [12] R. Słowiński, S. Greco, B. Matarazzo, "Rough-set-based decision support In Search Methodologies," *Springer, Boston, MA.*, pp. 557-609, 2014.
- [13] S. Pal, "Rough set and deep learning: Some concepts," *Academia Letters*, 1849, pp. 1-6, 2021.
- [14] Hoàng Thị Lan Giao, *Khía cạnh đại số và logic phát hiện luật theo tiếp cận tập thô*, Luận án Tiến sĩ Toán học, Viện Công Nghệ Thông Tin, 2007.
- [15] Nguyễn Đức Thuần, *Phủ tập thô và độ đo đánh giá hiệu năng tập luật quyết định*, Luận án Tiến sĩ Toán học, Viện Khoa học và Công nghệ Việt Nam, 2010.
- [16] Nguyễn Long Giang, *Nghiên cứu một số phương pháp khai phá dữ liệu theo hướng tiếp cận tập thô*, Luận án Tiến sĩ Toán học, Viện Công Nghệ Thông Tin, 2012.
- [17] M. Alimoussa, A. Porebski, N. Vandenbroucke, R. O. H. Thami, S. El Fkihi, "Clustering-based Sequential Feature Selection Approach for High Dimensional Data Classification," *VISIGRAPP (4: VISAPP)*, pp. 122-132, 2021.

- [18] S. Chormunge , S. Jena, "Correlation based feature selection with clustering for high dimensional data.," *Journal of Electrical Systems and Information Technology*, 5, p. 542–549, 2018.
- [19] T. P. Hong, Y. L. Liou, S. L. Wang, Bay Vo, "Feature selection and replacement by clustering attributes," *Vietnam J Comput Sci*, vol. 1, p. 47–55, 2014.
- [20] A. Janusz, D. Slezak, "Utilization of Attribute Clustering Methods for Scalable Computation of Reducts from High-Dimensional Data.," *Proceedings of the Federated Conference on Computer Science and Information Systems*, p. 295–302, 2012.
- [21] A. Janusz, D. Slezak, "Rough set methods for attribute clustering and selection," *Applied Artificial Intelligence*, vol. 28, p. 220–242, 2014.
- [22] J. Uddin, R. Ghazali, J. H. Abawajy, H. Shah, N. A. Husaini, A. Zeb, "Rough set based information theoretic approach for clustering uncertain categorical data," *PLOS ONE*, 2022.
- [23] W. Wei, J. Liang, X. Guo, P. Song, Y. Sun, "Hierarchical division clustering framework for categorical data," *Neurocomputing*, vol. 341, p. 118–134, 2019.
- [24] J. Uddin, R. Ghazali, M. M. Deris, "An empirical analysis of rough set categorical clustering techniques," *PloS one*, 12(1), 2017.
- [25] D. Dua, C. Graff, "UCI Machine Learning Repositories," <http://archive.ics.uci.edu/ml/>, 2019.
- [26] Y. Y. Yao, "Information-Theoretic Measures for Knowledge Discovery and Data Mining," *Part of the Studies in Fuzziness and Soft Computing book series (STUDFUZZ)*, vol. 119.
- [27] N. X. Vinh, J. Epps, J. Bailey, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837-2854, 2012.
- [28] A. Skowron, C. Rauszer, "The Discernibility Matrices and Functions in Information Systems. Intelligent Decision Support," *Handbook of Applications and Advances of the Rough Sets Theory*, Kluwer, Dordrecht, p. 331–362, 1992.
- [29] G. K. Singh, S. Mandal, "Cluster Analysis using Rough Set Theory," *Journal of Informatics and Mathematical Sciences*, Vols. 9, No. 3, pp. 509–520, , 2017.
- [30] S. Raheem, S. Al Shehabi, and A. M. Nassief, "MIGR: A Categorical Data Clustering Algorithm Based on Information Gain in Rough Set Theory," *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, Vols. 30, No. 05, pp. 757-771, 2022.
- [31] T. Herawan, M. M. Deris, J. H. Abawajy, "A rough set approach for selecting clustering attribute," *Knowledge-Based Systems*, vol. 23, p. 220–231, 2010.
- [32] U. Stańczyk, L. C. Jain, "Feature selection for data and pattern recognition: an introduction," *In Feature Selection for Data and Pattern Recognition*, vol. 584, pp. 1-7, 2015.
- [33] P. Dhal, C. Azad, "A comprehensive survey on feature selection in the various fields of machine learning," *Applied Intelligence*, pp. 1-39, 2021.
- [34] R. Jensen, Q. Shen, "A Rough Set-Aided System for Sorting WWW Bookmarks," *Lecture Notes in Computer Science*, vol. 2198, 2001.
- [35] G. Y. Wang, H. Yu, D. C. Yang, "Decision table reduction based on conditional information entropy," *Chinese Journal Of Computers-Chinese* , vol. 25 (7), pp. 759-766, 2002.

- [36] X. Hu, J. Han, T. Y. Lin, "A New Rough Sets Model Based on Database Systems," *Fundamenta Informaticae* 59 no. 2-3, pp. 135-152, 2004.
- [37] J. Han , R. Sanchez , X. Hu, "Feature Selection Based on Relative Attribute Dependency: An Experimental Study," *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pp. 214-223, 2005.
- [38] T. P. Hong, C. H. Chen, F. S. Lin, "Using group genetic algorithm to improve performance of attribute clustering, Appl.," *Soft Comput. J.*, p. <http://dx.doi.org/10.1016/j.asoc.2015.01.001>, 2015.
- [39] F. Pacheco, M. Cerrada, R. V. Sánchez, D. Cabrera, C. Li, J. V. de Oliveira, "Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery.," *Expert Systems with Application*, pp. 69-86, 2017.
- [40] T. P. Hong, P. C. Wang, C.K. Ting, "An evolutionary attribute clustering and selection method based on feature similarity," *The IEEE Congress on Evolutionary Computation*, 2010.
- [41] T. P. Hong, Y. L. Liou, "Attribute clustering in high dimensional feature spaces," *International Conference on Machine Learning and Cybernetics*, p. 19–22, 2007.
- [42] R. Jensen, Q. Shen, "Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches," *Wiley-IEEE Press eBook*, 2008.
- [43] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53-65, 1987.
- [44] S. R. A. Ahmed, I. A. Barazanchi, Z. A. Jaaz, H. R. Abdulshaheed, "Clustering algorithms subjected to K-mean and gaussian mixture model on multidimensional data set," *Periodicals of Engineering and Natural Sciences (PEN)*, 7(2), pp. 448-457, 2019.
- [45] G. Khandelwal, R. Sharma, "A simple yet fast clustering approach for categorical data.," *International Journal of Computer Applications*, 120(17)., pp. 25-30, 2015.
- [46] W. A. Hassanein, "Clustering algorithms for categorical data using concepts of significance and dependence of attributes," *European Scientific Journal*, vol. 10, pp. 381-400, 2014.
- [47] R. Jensen, Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Trans. Fuzzy Syst.* 17 (4), p. 824–838, 2009.
- [48] Mazlack, L. J., He, A., Zhu, Y., Coppock, S., "A rough set approach in choosing clustering attributes," *Proceedings of the ISCA 13th International Conference (CAINE 2000)*, p. 1–6, 2000.
- [49] D. Parmar, T. Wu, J. Blackhurst, "MMR: an algorithm for clustering categorical data using rough set theory," *Data and Knowledge Engineering*, vol. 63, p. 879–893, 2007.
- [50] M. M. Baroud, S. Z. M. Hashim, J. U. Ahsan, A. Zainal, "Positive region: An enhancement of partitioning attribute based rough set for categorical data," *Periodicals of Engineering and Natural Sciences*, Vols. Vol. 8, No. 4, December 2020.
- [51] M. M. Baroud, S. Z. m. Hashim, A. Zainul, J. Ahnad, "An New Algorithm-based Rough Set for Selecting Clustering Attribute in Categorical Data," *6th International Conference on Advanced Computing & Commucation Systems (ICACCS)*, pp. 1358-1364, 2020.
- [52] B. K. Tripathy, A. Goyal, R. Chowdhury, A. S. Patra, "MMeMeR: An algorithm for clustering heterogeneous data using rough set theory," *International Journal of Intelligent Systems and Applications*, 9(8), 25., pp. 25-33, 2017.

- [53] B. Tripathy, A. Ghosh, "SDR: An algorithm for clustering categorical data using rough set theory," *Recent Advances in Intelligent Computational Systems, IEEE*, pp. 867-872, 2011.
- [54] H. Qin, Xiuqin Ma, T. Herawan, J. M. Zain, "MGR: An information theory based hierarchical divisive clustering algorithm for categorical data," *Knowledge-Based Systems*, vol. 67, p. 401–411, 2014.
- [55] H. Qin, Xiuqin Ma, J. M. Zain, T. Herawan, "A novel soft set approach in selecting clustering attribute," *Knowledge-Based Systems*, vol. 36, p. 139–149, 2012.
- [56] Y. Y. Yao, Y. Zhao, J. Wang, "On reduct construction algorithms, Rough Sets and Knowledge Technology," *First International Conference, RSKT 2006, Proceedings, LNAI 4062*, pp. 297-304, 2006.
- [57] J. McCaffrey, "Data Clustering Using Entropy Minimization.," <http://visualstudiomagazine.com/Articles/2013/02/01/Data-Clustering-Using-Entropy-Minimization.aspx?Page=2&p=1>, 2018.
- [58] N. T. T. Hien, H. V. Nam, "A k-Means-Like Algorithm for Clustering Categorical Data Using an Information Theoretic-Based Dissimilarity Measure," *Foundations of Information and Knowledge Systems: 9th International Symposium, FoIKS, Linz, Austr*, 2016.
- [59] J. Liang, K. S. Chin, C. Dang, R. C. M. Yam, "A new method for measuring uncertainty and fuzziness in rough set theory," *International Journal of General Systems*, vol. 31(4), pp. 331-342, 2002.
- [60] Y. Zhao, "R and Data Mining: Examples and Case Studies. <https://www.webpages.uidaho.edu/~stevel/517/RDataMining-book.pdf>," *Published by Elsevier*, December 2012.
- [61] M. J. Wierman, "Measuring uncertainty in rough set theory," *International Journal of General System*, 28(4-5), pp. 283-297, 1999.
- [62] J. Zhou, D. Miao, W. Pedrycz, H. Zhang, "Analysis of alternative objective functions for attribute reduction in complete decision tables," *Soft Comput* 15, p. 1601–1616, 2011.
- [63] Q. Song, J. Ni, G. Wang, "A fast clustering based feature subset selection algorithm for highdimensional data.," *IEEE Transactions on Knowledge and Data Engineering*, 25(1), p. 2013, 1–14.
- [64] K. Zhu, J. Yang, "A cluster-based sequential feature selection algorithm.," *In 2013 Ninth International Conference on Natural Computation (ICNC)*, p. 848–852, 2013.
- [65] C. Shannon , "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423, 1948.
- [66] N. Xie, M. Liu, Z. Li, G. Zhang, "New measures of uncertainty for an interval-valued information system," *Information Sciences*, vol. 470, pp. 156-174, 2019.
- [67] M. Zhang, J. T. Yao, "A rough sets based approach to feature selection," *IEEE Annual Meeting of the Fuzzy Information*, 2004.
- [68] M. S. Raza, U. Qamar, "Feature selection using rough set-based direct dependency calculation by avoiding the positive region," *International Journal of Approximate Reasoning* 92, p. 175–197, 2018.
- [69] K. Thangavel, A. Pethalakshmi, "Dimensionality reduction based on rough set theory: A review," *Applied Soft Computing* 9, p. 1–12, 2009.
- [70] Z. Abbas, A. Burney, "A survey of software packages used for rough set analysis," *Journal of Computer and Communications*, 4 (9), pp. 10-18, 2016.

- [71] S. L. M. Belaidan, L. Y. Yee, N. A. Abd Rahman, K. S. Harun, "Implementing k-means clustering algorithm in collaborative trip advisory and planning system," *Periodicals of Engineering and Natural Sciences (PEN)*, 7(2), pp. 723-740, 2019.
- [72] R. Kohavi, G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, 97(1-2), pp. 273-324, 1997.
- [73] A. Jakulin, "Machine Learning Based on Attribute Interactions," *PhD Dissertation*, 2005.
- [74] A. Sandro, A. Pessoa, S. Stephany, "An Innovative Approach for Attribute Reduction in Rough Set Theory," *Intelligent Information Management*, pp. 223-239, 2014.
- [75] T. P. Hong, P. C. Wang, Y. C. Lee, "Cybernetics and Systems: An International Journal," *An effective attribute clustering approach for feature selection and replacement.*, p. 657–669, 2009.
- [76] D. E. Goldberg, "Genetic algorithms in search, optimization & machine learning," *Boston, MA, USA: Addison Wesley*, 1989.
- [77] L. Kaufman, P. J. Rousseeuw, "Computing groups in data: an introduction to cluster analysis," *John Wiley & Sons, Toronto*, 1990.
- [78] I. K. Park, G. S. Choi, "Rough set approach for clustering categorical data using information-theoretic dependency measure," *Information Systems*, vol. 48, pp. 289-295, 2015.
- [79] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "On clustering validation techniques," *Journal of intelligent information systems*, vol. 17, pp. 107-145, 2001.
- [80] F. M. Reza, "An introduction to information theory," *Dover Publications, New York*, 1994.
- [81] Han, J., Pei, J., Tong, H, "Data Mining: Concepts and Techniques," *Morgan Kanufmann*, 2022.
- [82] S. Raheem, S. Al Shehabi, and A. M. Nassief, "MIGR: A Categorical Data Clustering Algorithm Based on Information Gain in Rough Set Theory," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vols. Vol. 30, No. 05, pp. 757-771, 2022.
- [83] T. Herawan, "Rough Set Approach for Categorical Data Clustering," *A thesis submitted in fulfillment of requirements for the award of the Doctor of Philosophy*, 2010.
- [84] T. Herawan, "Rough clustering for cancer data sets," *International Journal of Modern Physics: Conference Series*, vol. 09, pp. 240-258, 2012.