

FILTER-WRAPPER INCREMENTAL ALGORITHM FOR ATTRIBUTE REDUCTION IN INCOMPLETE DECISION TABLES WHEN OBJECT SET AND ATTRIBUTE SET CHANGE VALUE

Nguyen Anh Tuan^{1*}, Nguyen Long Giang², Vu Duc Thi³

¹Vinh Phuc College, ²Institute of Information Technology - VAST

³Institute of Information Technology - VNU

ARTICLE INFO	ABSTRACT
Received: 22/6/2021	In the development trend of big data, decision tables are often incomplete, increasingly large in size and always changing and updating. The construction of incremental algorithms efficiency according to the filter - wrapper approach to minimize the number attribute of reduct, thereby improving the efficiency of classification and machine learning models is a very important research issue. In this paper, we propose two distance based filter-wrapper incremental algorithms: the IFWA_U_Obj algorithm in case the object set change value and the IFWA_U_Attr algorithm in case attribute set change value. Experimental results show that proposed filter - wrapper incremental algorithm decreases significantly the number of attributes in the reduct and improves classification accuracy compared to filter incremental algorithms reported.
Revised: 12/8/2021	
Published: 18/8/2021	
KEYWORDS	
Tolerance Rough Set	
Incomplete Decision Tables	
Attribute Reduction	
Reduct	
Incremental Algorithm	
Filter-Wrapper	

THUẬT TOÁN GIA TĂNG LỘC - ĐÓNG GÓI TÌM TẬP RÚT GỌN TRONG BẢNG QUYẾT ĐỊNH KHÔNG ĐẦY ĐỦ KHI TẬP ĐỐI TƯỢNG VÀ TẬP THUỘC TÍNH THAY ĐỔI GIÁ TRỊ

Nguyễn Anh Tuấn^{1*}, Nguyễn Long Giang², Vũ Đức Thi³

¹Trường Cao đẳng Vĩnh Phúc, ²Viện Công nghệ thông tin - Viện Hàn lâm Khoa học và Công nghệ Việt Nam

³Viện Công nghệ thông tin - Đại học Quốc gia Hà Nội

THÔNG TIN BÀI BÁO	TÓM TẮT
Ngày nhận bài: 22/6/2021	Trong xu thế phát triển của dữ liệu lớn, các bảng quyết định thường không đầy đủ, ngày càng có kích thước lớn và luôn thay đổi, cập nhật. Việc xây dựng các thuật toán gia tăng hiệu quả theo phương pháp tiếp cận lộc - đóng gói nhằm giảm thiểu số thuộc tính tập rút gọn, từ đó nâng cao hiệu quả các mô hình phân lớp, học máy là vấn đề nghiên cứu rất cần thiết. Trong bài báo này, chúng tôi đề xuất hai thuật toán gia tăng lộc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi sử dụng khoảng cách: thuật toán IFWA_U_Obj trong trường hợp tập đối tượng thay đổi giá trị và thuật toán IFWA_U_Attr trong trường hợp tập thuộc tính thay đổi giá trị. Kết quả thực nghiệm trên các tập dữ liệu mẫu cho thấy, các thuật toán gia tăng lộc - đóng gói đề xuất hiệu quả hơn về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp so với các thuật toán lộc đã công bố.
Ngày hoàn thiện: 12/8/2021	
Ngày đăng: 18/8/2021	
TỪ KHÓA	
Lý thuyết tập thô	
Bảng quyết định không đầy đủ	
Rút gọn thuộc tính	
Tập rút gọn	
Thuật toán gia tăng	
Lộc - Đóng gói	

DOI: <https://doi.org/10.34238/tnu-jst.4684>

* Corresponding author. Email: tuanna573@gmail.com

1. Giới thiệu

Bài toán tìm tập rút gọn trên bảng quyết định không đầy đủ thay đổi ngày càng trở nên quan trọng, các nhà nghiên cứu đã đề xuất nhiều thuật toán gia tăng để giảm thời gian thực thi. Chẳng hạn như lý thuyết tập thô do Pawlak [1] đề xuất được xem là công cụ hiệu quả giải quyết bài toán rút gọn thuộc tính trên bảng quyết định đầy đủ, đã và đang thu hút sự quan tâm của các nhà nghiên cứu trong suốt bốn thập kỷ qua. Trong thực tế, các bảng quyết định thường thiếu giá trị trên miền giá trị của tập thuộc tính, gọi là *bảng quyết định không đầy đủ*. Để giải quyết bài toán không qua bước tiền xử lý giá trị thiếu, Kryszkiewicz [2] mở rộng quan hệ tương đương trong lý thuyết tập thô truyền thống thành quan hệ dung sai và xây dựng mô hình tập thô dung sai. Với dữ liệu cố định, các tác giả trong [3] đã xây dựng công thức tính khoảng cách, từ đó đề xuất thuật toán IDS_F_DAR tìm tập rút gọn sử dụng khoảng cách. Thuật toán này theo tiếp cận lọc truyền thống, tập rút gọn chưa được tối ưu. Để khắc phục nhược điểm này, các tác giả trong [4] đã đề xuất thuật toán IDS_FW_DAR theo hướng tiếp cận lai ghép lọc - đóng gói. Trường hợp bảng quyết định thay đổi và có kích thước lớn, việc thực hiện thuật toán trên toàn bộ bảng quyết định sẽ gặp khó khăn về thời gian thực hiện. Do đó, các nhà nghiên cứu đề xuất hướng tiếp cận tính toán gia tăng tìm tập rút gọn. Các thuật toán gia tăng có khả năng giảm thiểu thời gian thực hiện và có khả năng thực hiện trên các bảng quyết định không đầy đủ kích thước lớn bằng giải pháp chia nhỏ bảng quyết định.

Trong mấy năm gần đây, một số thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ đã được đề xuất bởi các nhóm nghiên cứu với các trường hợp: bổ sung và loại bỏ tập đối tượng [5]-[9], bổ sung và loại bỏ tập thuộc tính [10], tập đối tượng và tập thuộc tính thay đổi giá trị [11], [12]. Các tác giả trong [11] xây dựng công thức cập nhật miền dương trong trường hợp tập đối tượng thay đổi giá trị, trên cơ sở đó đề xuất thuật toán gia tăng FSMV cập nhật tập rút gọn. Các tác giả trong [12] xây dựng công thức cập nhật độ đo không nhất quán trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị, trên cơ sở đó đề xuất hai thuật toán: thuật toán Object-R cập nhật tập rút gọn trong trường hợp tập đối tượng thay đổi giá trị và Attribute-R trong trường hợp tập thuộc tính thay đổi giá trị. Tuy nhiên, các thuật toán đề xuất nêu trên đều theo hướng tiếp cận lọc truyền thống. Do đó, bài báo này nghiên cứu, đề xuất các thuật toán gia tăng tìm tập rút gọn theo hướng tiếp cận lọc - đóng gói sử dụng khoảng cách trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị nhằm giảm thiểu số lượng thuộc tính tập rút gọn, từ đó nâng cao hiệu quả của mô hình phân lớp. Kết quả thực nghiệm trên các tập dữ liệu mẫu cho thấy, thuật toán gia tăng lọc - đóng gói đề xuất hiệu quả hơn về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp so với các thuật toán lọc đã công bố.

Cấu trúc bài báo như sau: Phần 1: Giới thiệu; Phần 2: Phương pháp nghiên cứu; Phần 3: Kết quả và bàn luận; Phần 4: Kết luận.

2. Phương pháp nghiên cứu

2.1. Khái niệm cơ bản

Bảng quyết định là một cặp $DS = (U, C \cup \{d\})$ trong đó $U \neq \emptyset$ là tập hữu hạn các đối tượng; $C \neq \emptyset$ là tập hữu hạn các thuộc tính điều kiện; d là thuộc tính quyết định. Mỗi thuộc tính $a \in C$ xác định một ánh xạ: $a: U \rightarrow V_a$ với V_a là tập giá trị của thuộc tính $a \in C$. Nếu V_a chứa *giá trị thiếu* thì DS được gọi là *bảng quyết định không đầy đủ*, được biểu diễn bởi $IDS = (U, C \cup \{d\})$ với $'*' \notin V_d$, trong đó giá trị thiếu được biểu diễn là $'*'$.

Xét $IDS = (U, C \cup \{d\})$, với mỗi tập con thuộc tính $P \subseteq C$, ta định nghĩa một quan hệ nhị phân trên U như sau:

$SIM(P) = \{(u, v) \in U \times U \mid \forall a \in P, a(u) = a(v) \vee a(u) = * \vee a(v) = *\}$ với $a(u)$ là giá trị thuộc tính a tại đối tượng u . $SIM(P)$ được gọi là *quan hệ dung sai* (tolerance relation) trên U vì chúng có tính phản xạ, đối xứng nhưng không có tính bắc cầu. Dễ thấy, $SIM(P) = \bigcap_{a \in P} SIM(\{a\})$. Với $u \in U$, $S_P(u) = \{v \in U \mid (u, v) \in SIM(P)\}$ được gọi là một *lớp dung sai* của đối tượng u . $S_P(u)$ là tập các đối tượng không phân biệt được với u trên quan hệ dung sai $SIM(P)$.

Định nghĩa: Cho $IDS = (U, C \cup D)$ với $U = \{u_1, u_2, \dots, u_n\}$ và $P \subseteq C$. Khi đó, ma trận dung sai của quan hệ $SIM(P)$, ký hiệu $M(P) = [p_{ij}]_{n \times n}$, được định nghĩa:

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

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

Với việc biểu diễn quan hệ dung sai $SIM(P)$ bằng ma trận dung sai $M(P)$, ta có mọi $u_i \in U$, $S_P(u_i) = \{u_j \in U \mid p_{ij} = 1\}$ và $|S_P(u_i)| = \sum_{j=1}^n p_{ij}$. Với $P, Q \subseteq C, u \in U$, ta có $S_{P \cup Q}(u) = S_P(u) \cap S_Q(u)$. Giả sử $M(P) = [p_{ij}]_{n \times n}$, $M(Q) = [q_{ij}]_{n \times n}$ là hai ma trận dung sai của $SIM(P)$, $SIM(Q)$, khi đó ma trận dung sai trên tập thuộc tính $S = P \cup Q$ là:

$$M(S) = M(P \cup Q) = [s_{ij}]_{n \times n} \text{ với } s_{ij} = p_{ij} \cdot q_{ij}$$

2.2. Phương pháp gia tăng rút gọn thuộc tính khi tập đối tượng, tập thuộc tính thay đổi giá trị

Trong phần này, chúng tôi xây dựng công thức gia tăng tính khoảng cách và đề xuất hai thuật toán hiệu quả tìm tập rút gọn trong trường hợp tập đối tượng và tập thuộc tính thay đổi giá trị.

2.2.1. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi tập đối tượng thay đổi giá trị

Mệnh đề 1 [3]. Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$ và $M(C) = [c_{ij}]_{n \times n}$, $M(\{d\}) = [d_{ij}]_{n \times n}$ tương ứng là ma trận dung sai trên C và d . Khi đó, khoảng cách giữa hai tập thuộc tính C và $C \cup \{d\}$ được xác định như sau: $D(C, C \cup \{d\}) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (c_{ij} - c_{ij} \cdot d_{ij})$ (2)

Mệnh đề 2. Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$. Không mất tính tổng quát, giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị thành $\Delta U' = \{u'_k, u'_{k+1}, \dots, u'_{k+s-1}\}$. Với $M_U(C) = [c_{ij}]_{n \times n}$ và $M_U(\{d\}) = [d_{ij}]_{n \times n}$ tương ứng là ma trận dung sai trên C và $\{d\}$, khi đó các phần tử $c_{i,k}, \dots, c_{i,k+s-1}$ bị thay đổi giá trị thành $c'_{i,k}, \dots, c'_{i,k+s-1}$ với $i = k..(k+s-1)$. Giả sử $D_U(C, C \cup \{d\})$ là khoảng cách sau khi cập nhật tập đối tượng ΔU và $D_U(C, C \cup \{d\})$ là công thức khoảng cách trước khi cập nhật. Khi đó, công thức tính gia tăng khoảng cách như sau:

$$D_{U'}(C, C \cup \{d\}) = D_U(C, C \cup \{d\}) + \frac{2}{n^2} \sum_{i=k}^{k+s-1} \sum_{j=1}^n (c'_{i,j} - c_{i,j}) (1 - d_{i,j}) \quad (3)$$

Từ mệnh đề 2, xây dựng mệnh đề 3 như sau:

Mệnh đề 3. Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$ và $R \subseteq C$ là tập rút gọn dựa trên khoảng cách. Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị thành $\Delta U' = \{u'_k, u'_{k+1}, \dots, u'_{k+s-1}\}$ và U' là tập đối tượng sau khi thay đổi giá trị. Với $M_U(C) = [c_{ij}]_{n \times n}$ và $M_U(\{d\}) = [d_{ij}]_{n \times n}$

trương ứng là ma trận dung sai trên C , giả sử các phần tử $c_{ik}, \dots, c_{i,k+s-1}$ bị thay đổi giá trị thành $c'_{ik}, \dots, c'_{i,k+s-1}$ với $i = k..(k+s-1)$. Khi đó ta có: Nếu $d_{ij} = 1$ hoặc $c'_{ij} = c_{ij}$ với mọi $k \leq i \leq k+s-1, 1 \leq j \leq n$ thì R là tập rút gọn của $IDS' = (U', C \cup \{d\})$.

Trong mục này, bài báo đề xuất thuật toán gia tăng tìm tập rút gọn theo tiếp cận lọc - đóng gói. Thuật toán bao gồm hai giai đoạn: *Giai đoạn lọc*: Tìm các ứng viên cho tập rút gọn. *Giai đoạn đóng gói*: Tìm tập rút gọn có độ chính xác phân lớp lớn nhất. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi tập đối tượng thay đổi giá trị được mô tả như sau:

Thuật toán FWIA_U_Obj (Filter-Wrapper Incremental Algorithm for Attribute Reduction in Incomplete Decision Tables when Update Objects).

Đầu vào: Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$

- Tập rút gọn $R \subseteq C$.

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

- Tập đối tượng gồm s phần tử $AU = \{u_k, u_{k+1}, \dots, u_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị thành $AU' = \{u'_k, u'_{k+1}, \dots, u'_{k+s-1}\}$ U' là tập đối tượng sau khi thay đổi giá trị.

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

Bước 1: Khởi tạo và kiểm tra

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

2. Tính các ma trận $M_{U'}(R)$, $M_{U'}(C)$, $M_{U'}(\{d\})$

3. If $d_{ij} = 1$ or $c'_{ij} = c_{ij}$ for any $k \leq i \leq k+s-1, 1 \leq j \leq n$ then Return R ;

Bước 2: Tìm tập rút gọn

4. Tính độ đo khoảng cách $D_U(R, R \cup \{d\}), D_U(C, C \cup \{d\})$

5. Tính độ đo khoảng cách $D_{U'}(R, R \cup \{d\}), D_{U'}(C, C \cup \{d\})$ sử dụng công thức gia tăng trong mệnh đề 2;

//Loại bỏ các thuộc tính dư thừa trong R

6. For each $a \in R$

7. If $D_{U'}(R - \{a\}, (R - \{a\}) \cup \{d\}) = D_{U'}(C, C \cup \{d\})$ then $R := R - \{a\}$;

//Giai đoạn lọc

// Bổ sung các thuộc tính còn lại vào R

8. Repeat

9. For each $r \in C - R$

10. Tính $SIG_R(r)$;

11. Chọn $r_m \in C - R$ sao cho $SIG_R(r_m) = \max_{r \in C - R} \{SIG_R(r)\}$;

12. $R := R \cup \{r_m\}$;

13. $T := T \cup R$;

14. Until $D_{U'}(R, R \cup \{d\}) = D_{U'}(C, C \cup \{d\})$

// Giai đoạn đóng gói

15. Đặt $t := |T|$; // $T = \{R \cup \{r_{i_1}\}, R \cup \{r_{i_1}, r_{i_2}\}, \dots, R \cup \{r_{i_1}, r_{i_2}, \dots, r_{i_t}\}\}$

16. Đặt $T_1 = \{R \cup \{r_{i_1}\}, T_2 = R \cup \{r_{i_1}, r_{i_2}\}, \dots, T_t = R \cup \{r_{i_1}, r_{i_2}, \dots, r_{i_t}\}\}$

17. For $i = 1$ to t

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

19. $R_{best} = T_{i_b}$ với T_{i_b} có độ chính xác phân lớp cao nhất.

20. Return R_{best} .

2.2.2. Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi tập thuộc tính thay đổi giá trị

Phần này xây dựng công thức gia tăng tính khoảng cách trong trường hợp tập thuộc tính thay đổi giá trị bởi mệnh đề 4 dưới đây.

Mệnh đề 4. Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$. Giả sử tập s thuộc tính $\Delta C = \{c_k, c_{k+1}, \dots, c_{k+s-1}\}$ với $1 \leq k \leq n, s \geq 1$ bị thay đổi giá trị. Giả sử $M^{old}(\Delta C) = [c_{ij}^{old}]_{n \times n}$, $M^{new}(\Delta C) = [c_{ij}^{new}]_{n \times n}$ tương ứng là ma trận dung sai của tập thuộc tính ΔC trước và sau khi thay đổi giá trị và $M(A) = [a_{ij}]_{n \times n}$, $M(\{d\}) = [d_{ij}]_{n \times n}$ tương ứng là ma trận dung sai trên là ma trận dung sai của tập thuộc tính còn lại không thay đổi giá trị $A = C - \Delta C$ và $\{d\}$. Giả sử $D(C, C \cup \{d\})$, $D'(C, C \cup \{d\})$ tương ứng là khoảng cách trước khi và sau khi tập thuộc tính ΔC thay đổi giá trị. Khi đó, công thức tính gia tăng khoảng cách như sau:

$$D'(C, C \cup \{d\}) = D(C, C \cup \{d\}) + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n a_{ij} \cdot (c_{ij}^{new} - c_{ij}^{old}) \cdot (1 - d_{ij})$$

Thuật toán gia tăng lọc - đóng gói tìm tập rút gọn khi tập thuộc tính thay đổi giá trị được mô tả như sau:

Thuật toán FWIA_U_Attr (Filter-Wrapper Incremental Algorithm for Attribute Reduction in Incomplete Decision Tables when Update Attributes).

Đầu vào: 1) Cho $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \dots, u_n\}$, tập rút gọn $R \subseteq C$ các ma trận dung sai $M(C) = [c_{ij}]_{n \times n}$, $M(\{d\}) = [d_{ij}]_{n \times n}$, khoảng cách $D(C, C \cup \{d\})$;

2) Tập thuộc tính ΔC bị thay đổi giá trị, với $\Delta C \subseteq C$;

Đầu ra: Tập rút gọn R' của $IDS' = (U, C \cup \{d\})$ sau khi ΔC bị thay đổi giá trị.

Bước 1: Khởi tạo

1. $T := \emptyset$; // Chứa các ứng viên tập rút gọn
2. Đặt $A := C - \Delta C$;
3. Tính ma trận dung sai $M(A) = [a_{ij}]_{n \times n}$, $M^{new}(\Delta C) = [c_{ij}^{new}]_{n \times n}$, $M^{old}(\Delta C) = [c_{ij}^{old}]_{n \times n}$;
4. Tính khoảng cách $D'(R, R \cup \{d\})$, $D'(C, C \cup \{d\})$ bởi công thức gia tăng trong mệnh đề 4;
- // Loại bỏ các thuộc tính dư thừa trong R ;
5. For each $a \in R$
6. If $D'(R - \{a\}, (R - \{a\}) \cup \{d\}) = D'(C, C \cup \{d\})$ then $R := R - \{a\}$;

Bước 2: Thực hiện thuật toán tìm tập rút gọn

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

7. While $D'(R, R \cup \{d\}) \neq D'(C, C \cup \{d\})$ do
8. Begin
9. For each $a \in C - R$ tính $SIG_R(a) = D'(R, R \cup \{d\}) - D'(R \cup \{a\}, R \cup \{a\} \cup \{d\})$
- Với $D'(R \cup \{a\}, R \cup \{a\} \cup \{d\})$ được tính bởi công thức gia tăng trong mệnh đề 4;
10. Chọn $a_m \in C - R$ sao cho $SIG_R(a_m) = \max_{a \in C - R} \{SIG_R(a)\}$;
11. $R := R \cup \{a_m\}$;
12. $T := T \cup R$;
13. End;
- // Giai đoạn đóng gói
14. Đặt $t := |T|$; // $T = \{R \cup \{r_1\}, R \cup \{r_2\}, \dots, R \cup \{r_t\}, r_1, r_2, \dots, r_t\}$
15. Đặt $T_1 = \{R \cup \{r_1\}, T_2 = R \cup \{r_1, r_2\}, \dots, T_t = R \cup \{r_1, r_2, \dots, r_t\}\}$
16. For $i = 1$ to t
17. Tính độ chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp kiểm tra chéo 10-fold;

18. $R_{best} = T_{i_b}$ với T_{i_b} có độ chính xác phân lớp cao nhất.

19. Return R_{best} .

3. Kết quả và bàn luận

Trong phần này, chúng tôi tiến hành thực nghiệm để đánh giá hiệu quả của thuật toán FWIA_U_Obj

3.1. Mục tiêu thực nghiệm

Đánh giá tính hiệu quả của thuật toán gia tăng lọc - đóng gói FWIA_U_Obj tìm tập rút gọn khi tập đối tượng thay đổi giá trị dựa trên các tiêu chí: *số lượng thuộc tính trong tập rút gọn, độ chính xác phân lớp và thời gian thực hiện*. Thuật toán FWIA_U_Obj được so sánh với hai thuật toán FSMV [11] và Object-R [12].

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

3.2. Số liệu và môi trường thực nghiệm

Chúng tôi tiến hành cài đặt cả 3 thuật toán: FWIA_U_Obj, FSMV và Object-R. Sau đó chạy 3 thuật toán trên cùng môi trường thực nghiệm đó là trên máy tính cá nhân PC: Bộ xử lý Intel®, Core™ i7-3770, 3,40 GHz, Windows 7 sử dụng Matlab. Dữ liệu thực nghiệm là: 06 bộ dữ liệu được lấy trong kho dữ liệu UCI

<<https://archive.ics.uci.edu/ml/datasets.php>>)

Dữ liệu thực nghiệm được mô tả ở bảng 1. Mỗi tập dữ liệu được chia ngẫu nhiên thành hai phần xấp xỉ bằng nhau: Tập dữ liệu không thay đổi được ký hiệu là O_{ori} và tập dữ liệu bị thay đổi được ký hiệu là O_{chan} . Tiếp theo, tập dữ liệu bị thay đổi O_{chan} được chia thành năm phần bằng nhau được ký hiệu lần lượt là O_1, O_2, O_3, O_4, O_5 . Với tập dữ liệu O_{chan} , chúng tôi thực hiện cập nhật ngẫu nhiên giá trị thuộc tính của các đối tượng bị thay đổi, bảo đảm nguyên tắc các giá trị bị thay đổi thuộc miền giá trị của thuộc tính ban đầu. Trong bảng 1, các cột $|O|, |O_{ori}|, |O_{chan}|, |A|, |k|$ được ký hiệu tương ứng là: Số đối tượng; Số đối tượng trong O_{ori} ; Số đối tượng trong O_{chan} ; Số thuộc tính điều kiện; Số lớp quyết định.

Bảng 1. Các bộ dữ liệu được sử dụng trong thực nghiệm khi tập đối tượng thay đổi giá trị

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

3.3. Kịch bản thực nghiệm

Trước hết, chúng tôi thực hiện thuật toán IDT_FW_DAR [4] để tìm tập rút gọn trên tập đối tượng ban đầu, làm đầu vào cho các thuật toán gia tăng. Tiếp theo, thực hiện cài đặt và chạy 03 thuật toán FWIA_U_Obj, FSMV và Object-R khi lần lượt đưa vào các tập đối tượng thay đổi giá trị O_1, O_2, O_3, O_4, O_5 . Sau đó, các giá trị số lượng thuộc tính tập rút gọn, độ chính xác phân lớp và thời gian thực hiện được ghi lại.

3.4. Đánh giá thuật toán FWIA_U_Obj trên hai tiêu chí: số lượng thuộc tính trong tập rút gọn và độ chính xác phân lớp

Bảng 2 trình bày kết quả về số thuộc tính trong tập rút gọn và độ chính xác phân lớp của các thuật toán FWIA_U_Obj, FSMV và Object-R. Trong đó, cột $|R|$ và Acc lần lượt là số thuộc tính trong tập

rút gọn và độ chính xác phân lớp. Dựa trên kết quả trong bảng 2 ta thấy rằng, độ chính xác phân lớp của thuật toán gia tăng lại ghép lọc - đóng gói FWIA_U_Obj cao hơn một chút so với FSMV và Object-R trên tất cả các tập dữ liệu và trên tất cả các bước lặp khi đưa lần lượt các tập đối tượng thay đổi giá trị O_1, O_2, O_3, O_4, O_5 . Hơn nữa, số lượng thuộc tính trong tập rút gọn thu được bởi FWIA_U_Obj nhỏ hơn nhiều so với FSMV và Object-R, đặc biệt là trong tập dữ liệu có nhiều thuộc tính như Ad.data. Do đó, mô hình phân lớp dựa trên tập rút gọn của thuật toán FWIA_U_Obj hiệu quả hơn mô hình phân lớp của thuật toán FSMV và thuật toán Object-R về chất lượng phân lớp và độ phức tạp của mô hình. Có thể thấy rằng, thuật toán Object-R hiệu quả hơn một chút so với thuật toán FSMV về cả độ chính xác của phân lớp và số lượng thuộc tính trong tập rút gọn.

Bảng 2. Số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của ba thuật toán FWIA_U_Obj, FSMV và Object-R

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		FSMV		Object-R	
			R	Acc	R	Acc	R	Acc
1	Audiology	O_1	11	78,12	18	78,06	17	77,92
		O_2	12	79,24	23	78,84	22	78,16
		O_3	9	75,46	12	72,46	13	73,45
		O_4	14	81,28	26	80,72	24	80,27
		O_5	13	80,74	19	79,25	18	79,92
2	Soybean-large	O_1	8	91,12	14	90,23	13	90,46
		O_2	9	92,54	16	91,17	16	91,17
		O_3	6	88,56	15	87,48	15	87,48
		O_4	11	89,23	22	89,24	21	90,15
		O_5	10	90,78	18	89,72	19	90,26
3	house-votes-84	O_1	5	92,36	10	92,05	9	91,84
		O_2	6	93,84	12	92,18	11	92,82
		O_3	8	94,15	12	93,46	12	93,46
		O_4	6	92,87	11	92,14	11	92,14
		O_5	5	91,72	9	90,58	10	91,05
4	Arrhythmia	O_1	22	72,18	41	71,24	38	71,69
		O_2	24	73,45	45	72,92	42	72,28
		O_3	18	71,26	51	71,02	46	70,89
		O_4	15	69,18	34	68,72	31	68,06
		O_5	21	74,82	46	73,86	43	74,15
5	Anneal	O_1	6	92,08	14	90,46	13	91,15
		O_2	8	93,16	18	92,95	18	92,54
		O_3	13	91,85	25	91,05	24	91,24
		O_4	9	89,28	17	88,48	17	88,95
		O_5	11	93,18	23	92,45	22	92,84
6	Ad	O_1	25	90,18	54	89,15	52	89,82
		O_2	32	91,23	61	90,68	55	90,23
		O_3	24	86,72	65	85,18	59	86,04
		O_4	36	92,55	54	91,45	51	91,11
		O_5	37	92,94	58	91,16	53	91,84

3.5. Đánh giá thời gian thực hiện của thuật toán FWIA_U_Obj

Thời gian thực hiện của thuật toán FWIA_U_Obj, FSMV và Object-R (tính theo giây) được trình bày như trong bảng 3.

Trên tất cả các tập dữ liệu trong bảng 3, thuật toán FWIA_U_Obj có thời gian thực hiện cao hơn thuật toán FSMV và thuật toán Object-R vì thuật toán FWIA_U_Obj cần nhiều thời gian hơn để chạy phân lớp trong giai đoạn đóng gói. Trong khi đó, thời gian thực hiện của thuật toán

Object-R cao hơn một chút thuật toán FSMV vì thời gian tính độ không nhất quán trong Object-R cao hơn thời gian tính miền dương trong FSMV.

Bảng 3. Thời gian thực hiện của ba thuật toán FWIA_U_Obj, FSMV và Object-R (tính bằng giây)

STT	Tập dữ liệu	Tập dữ liệu thay đổi giá trị	FWIA_U_Obj		FSMV		Object-R	
			Thời gian thực hiện	Tổng Thời gian thực hiện	Thời gian thực hiện	Tổng Thời gian thực hiện	Thời gian thực hiện	Tổng Thời gian thực hiện
1	Audiology	O_1	1,25	1,25	0,86	0,86	0,95	0,95
		O_2	1,38	2,63	0,92	1,78	1,02	1,97
		O_3	1,24	3,87	1,05	2,83	1,16	3,13
		O_4	1,64	5,51	1,16	3,99	1,25	4,38
		O_5	1,34	6,85	0,92	4,91	1,06	5,44
2	Soybean-large	O_1	0,86	0,86	0,54	0,54	0,59	0,59
		O_2	0,94	1,80	0,68	1,22	0,72	1,31
		O_3	1,06	2,86	0,75	1,97	0,82	2,13
		O_4	1,12	3,98	0,84	2,81	0,89	3,02
		O_5	0,85	4,83	0,68	3,49	0,75	3,77
3	house-votes-84	O_1	0,84	0,84	0,72	0,72	0,78	0,78
		O_2	0,63	1,47	0,52	1,24	0,59	1,37
		O_3	0,72	2,19	0,58	1,82	0,65	2,02
		O_4	0,68	2,87	0,49	2,31	0,52	2,54
		O_5	0,59	3,46	0,42	2,73	0,56	3,10
4	Arrhythmia	O_1	3,24	3,24	2,86	2,86	2,92	2,92
		O_2	3,65	6,89	2,95	5,81	3,05	5,97
		O_3	3,12	10,01	2,74	8,55	2,82	8,79
		O_4	2,96	12,97	2,25	10,80	2,34	11,13
		O_5	2,85	15,82	2,16	12,96	2,28	13,41
5	Anneal	O_1	0,98	0,98	0,65	0,65	0,72	0,72
		O_2	0,75	1,73	0,52	1,17	0,58	1,30
		O_3	0,86	2,59	0,68	1,85	0,76	2,06
		O_4	0,72	3,31	0,54	2,39	0,62	2,68
		O_5	0,78	4,09	0,57	2,96	0,65	3,33
6	Ad	O_1	7,35	7,35	5,46	5,46	5,82	5,82
		O_2	6,48	13,83	5,11	10,57	5,95	11,77
		O_3	7,84	21,67	6,08	16,65	6,24	18,01
		O_4	6,28	27,95	5,12	21,77	5,89	23,90
		O_5	5,72	33,22	4,86	26,63	5,17	29,07

4. Kết luận

Trong bài báo này, chúng tôi đã nghiên cứu đề xuất các thuật toán gia tăng lọc - đóng gói tìm tập rút gọn của bảng quyết định không đầy đủ thay đổi, sử dụng độ đo khoảng cách trong các tình huống tập đối tượng và tập thuộc tính thay đổi giá trị. Kết quả thực nghiệm cho thấy, các thuật toán đề xuất theo tiếp cận lọc - đóng gói giảm thiểu số lượng thuộc tính tập rút gọn và cải thiện độ chính xác của mô hình phân lớp so với các thuật toán gia tăng khác theo tiếp cận lọc đã công bố. Tuy nhiên, các thuật toán đề xuất có thời gian thực hiện cao hơn, đây là hạn chế của cách tiếp cận này. Trong thời gian tới, chúng tôi tiếp tục nghiên cứu, cải tiến các thuật toán gia tăng lọc - đóng gói đã công bố nhằm phù hợp với các lớp bài toán khác nhau trong thực tế, nhất là giảm thiểu thời gian thực hiện bằng giải pháp không chạy lặp lại các bộ phân lớp.

TÀI LIỆU THAM KHẢO/ REFERENCES

- [1] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, no. 5, pp. 341-356, 1982.
- [2] M. Kryszkiewicz, "Rough set approach to incomplete information systems," *Information Science*, vol. 112, pp. 39-49, 1998.
- [3] L. G. Nguyen and H. S. Nguyen, "Metric based attribute reduction in incomplete decision tables," *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing. Springer*, 2013, pp. 99-110.
- [4] A. T. Nguyen and L. G. Nguyen, "About a Distance Measure and Application for Finding Reduct in Incomplete Decision Tables," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 6294-6298, 2019.
- [5] D. Liu, T. Li, and J. Zhang, "A rough set-based incremental approach for learning knowledge in dynamic incomplete information systems," *International Journal of Approximate Reasoning*, vol. 55, no. 8, pp. 1764-1786, 2014.
- [6] W. H. Shu and W. B. Qian, "An incremental approach to attribute reduction from dynamic incomplete decision systems in rough set theory," *Data and Knowledge Engineering*, vol. 100, pp. 116-132, 2015.
- [7] J. Yu, L. Sang, and H. Dong, "Based on attribute order for dynamic attribute reduction in the incomplete information system," *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), IEEE*, 2018, pp. 2475-2478, doi: <https://doi.org/10.1007/s13042-020-01089-4>.
- [8] C. Zhang, J. Dai, and J. Chen, "Knowledge granularity based incremental attribute reduction for incomplete decision systems", *International Journal of Machine Learning and Cybernetics*, vol. 11, pp. 1141-1157, 2020. <https://doi.org/10.1007/s13042-020-01089-4>.
- [9] D. Zhang, R. Li, X. Tang, and Y. Zhao, "An incremental reduct algorithm based on generalized decision for incomplete decision tables," *2008 3rd International Conference on Intelligent System and Knowledge Engineering, IEEE*, vol. 1, pp. 340-344, 2008.
- [10] W. H. Shu and H. Shen, "Updating attribute reduction in incomplete decision systems with the variation of attribute set," *International Journal of Approximate Reasoning*, vol. 55, no. 3, pp. 867-884, 2014.
- [11] W. H. Shu and H. Shen, "Incremental feature selection based on rough set in dynamic incomplete data," *Pattern Recognition*, vol. 47, pp. 3890-3906, 2014.
- [12] X. Xie and X. Qin, "A novel incremental attribute reduction approach for dynamic incomplete decision systems," *International Journal of Approximate Reasoning*, vol. 93, pp. 443-462, 2018.