Về một thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ

Phạm Minh Ngọc Hà[1], Nguyễn Long Giang[2], Nguyễn Văn Thiện[3], Nguyễn Bá Quảng[4]
[1]Học viện Tài chính, Hà Nội
[2]Viện Công nghệ Thông tin, Viện Khoa học và Công nghệ Việt Nam
[3]Trường Đại học Công nghiệp Hà Nội
[4]Trường Đại học Kiến trúc Hà Nội

Tác giả liên hệ: Nguyễn Long Giang, nlgiang@ioit.ac.vn
Ngày nhận bài: 26/04/2019, ngày sửa chữa: 08/06/2019, ngày duyệt đăng: 09/06/2019
Xem sớm trực tuyến: 24/06/2019, định danh DOI: 10.32913/mic-ict-research-vn.v2019.n1.855
Biên tập lĩnh vực điều phối phản biện và quyết định nhận đăng: PGS.TS. Lê Hoàng Sơn

Tóm tắt: Mô hình tập thô dung sai là công cụ hiệu quả giải quyết bài toán rút gọn thuộc tính trên bảng quyết định không đầy đủ. Trong mấy năm gần đây, các nhà nghiên cứu đã đề xuất một số thuật toán gia tăng tìm tập rút gọn theo tiếp cận tập thô dung sai nhằm giảm thiểu thời gian thực hiện. Tuy nhiên, các thuật toán đề xuất đều theo hướng tiếp cận lọc truyền thống, nghĩa là bước kiểm tra độ chính xác phân lớp độc lập với bước tìm tập rút gọn. Do đó, tập rút gọn tìm được chưa tối ưu cả về số lượng thuộc tính và độ chính xác phân lớp. Trong bài báo này, chúng tôi đề xuất thuật toán gia tăng IDS_IFW_AO tìm tập rút gọn theo tiếp cận lai ghép lọc – đóng gói sử dụng độ đo khoảng cách. Kết quả thử nghiệm trên các tập dữ liệu mẫu cho thấy, thuật toán lai IDS_IFW_AO hiệu quả hơn thuật toán lọc IARM-I về độ chính xác phân lớp và số thuộc tính tập rút gọn.

Từ khóa: *Tập thô dung sai, khoảng cách, thuật toán gia tăng, bảng quyết định không đầy đủ, rút gọn thuộc tính, tập rút gọn.*

Title: **An Incremental Algorithm for Finding Reducts of Incomplete Decision Tables**

Abstract: Tolerance rough set models provide an effective tool for attribute reduction in incomplete decision tables. In recent years, some incremental algorithms have been proposed to find reducts of dynamic incomplete decision tables in order to reduce the computation time. However, they are classical filter algorithms, in which the classification accuracy of decision tables is computed after obtaining the reducts. Therefore, the obtained reducts of these algorithms are not optimal in terms of reduct cardinality and classification accuracy. In this paper, we propose the incremental filter–wrapper algorithm to find a reduct of an incomplete decision table in case of adding multiple objects. The experimental results on some sample datasets show that the proposed filter–wrapper algorithm is more effective than IARM-I, a state-of-the-art filter algorithm, in terms of classification accuracy and reduct cardinality.

Keywords: *Tolerance rough set, distance, incremental algorithm, incomplete decision table, attribute reduction, reduct.*

I. GIỚI THIỆU

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. Trong thực tế, các bảng quyết định thường thiếu giá trị trên miền giá trị thuộc tính, lúc đó bảng quyết định được gọi là không đầy đủ. Để giải quyết bài toán rút gọn thuộc tính và trích lọc luật trên bảng quyết định không đầy đủ, 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. Dựa trên mô hình tập thô dung sai, các nhà nghiên cứu đã đề xuất các phương pháp rút gọn thuộc tính theo tiếp cận lọc sử dụng các độ đo khác nhau, điển hình là phương pháp rút gọn thuộc tính sử dụng độ đo khoảng cách [3, 4].

Trong các bài toán thực tế, bảng quyết định không đầy đủ thường có kích thước lớn và luôn thay đổi, cập nhật. Việc áp dụng các thuật toán tìm tập rút gọn theo tiếp cận tập thô dung sai gặp nhiều thách thức. Trường hợp các bảng quyết định thay đổi, các thuật toán này tính lại tập rút gọn trên toàn bộ bảng quyết định sau khi thay đổi nên chi phí về thời gian tính toán tăng lên đáng kể. Trường hợp bảng quyết định có kích có 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. Theo tiếp cận tập thô truyền thống, các nghiên

cứu liên quan đến các phương pháp gia tăng tìm tập rút gọn trong bảng quyết định thay đổi khá sôi động và tập trung vào các trường hợp: bổ sung/loại bỏ tập đối tượng [5–10], bổ sung/loại bỏ tập thuộc tính [11, 12], tập đối tượng hay tập thuộc tính thay đổi giá trị [13–15].

Trong mấy năm gần đây, một số công trình nghiên cứu đã đề xuất các thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ theo tiếp cận tập thô dùng sai [16–22]. Zhang và các cộng sự [16] xây dựng thuật toán gia tăng tìm tập rút gọn sử dụng hàm quyết định suy rộng trong trường hợp bổ sung một đối tượng. Shu và các cộng sự [17, 18] xây dựng cơ chế cập nhật miền dương trong trường hợp bổ sung và loại bỏ tập đối tượng, trên cơ sở đó đề xuất các thuật toán gia tăng tìm tập rút gọn. Yu và các cộng sự [19] xây dựng các công thức gia tăng tính toán entropy trong trường hợp bổ sung, loại bỏ tập đối tượng và đề xuất thuật toán gia tăng tìm tập rút gọn. Shu và các cộng sự [20] xây dựng cơ chế cập nhật miền dương trong trường hợp bổ sung, loại bỏ tập thuộc tính, trên cơ sở đó đề xuất các thuật toán gia tăng tìm tập rút gọn. Shu và các cộng sự [21] xây dựng cơ chế cập nhật miền dương trong trường hợp tập đối tượng thay đổi tập giá trị, trên cơ sở đó đề xuất các thuật toán gia tăng tìm tập rút gọn. Xie và các cộng sự [22] xây dựng độ đo không nhất quán và đề xuất các thuật toán gia tăng tìm tập rút gọn sử dụng độ không nhất quán trong trường hợp tập đối tượng, tập thuộc tính thay đổi giá trị.

Kết quả thực nghiệm cho thấy, các thuật toán gia tăng đã công bố giảm thiểu đáng kể thời gian thực hiện so với các thuật toán không gia tăng. Do đó, chúng có thể thực thi được trên các bảng quyết định thay đổi, cập nhật, có kích thước lớn. Tuy nhiên, các thuật toán đã công bố nêu trên đều theo hướng tiếp cận lọc (filter) truyền thống. Với cách tiếp cận này, tập rút gọn tìm được là tập thuộc tính tối thiểu bảo toàn độ đo được định nghĩa. Việc đánh giá độ chính xác phân lớp của mô hình được thực hiện sau khi tìm được tập rút gọn. Vì vậy, tập rút gọn tìm được của các thuật toán nêu trên chưa tối ưu cả về số lượng thuộc tính và độ chính xác phân lớp.

Trong bài báo này chúng tôi đề xuất thuật toán gia tăng tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp bổ sung tập đối tượng sử dụng độ đo khoảng cách trong [3], được viết tắt là IDS_IFW_AO. Thuật toán đề xuất IDS_IFW_AO theo hướng tiếp cận lọc – đóng gói, trong đó giai đoạn lọc tìm các ứng viên tập rút gọn mỗi khi bổ sung thuộc tính có độ quan trọng lớn nhất, giai đoạn đóng gói tìm tập rút gọn có độ chính xác phân lớp cao nhất.

Kết quả thử nghiệm trên các bộ số liệu mẫu cho thấy, thuật toán IDS_IFW_AO có độ chính xác phân lớp cao hơn thuật toán IARM-I [17]. Hơn nữa, số lượng thuộc tính tập rút gọn của IDS_IFW_AO nhỏ hơn IARM-I. Cấu trúc bài báo như sau. Phần II trình bày một số khái niệm cơ bản. Phần III xây dựng công thức gia tăng tính độ đo khoảng cách trong trường hợp bổ sung tập đối tượng. Phần IV trình bày thuật toán gia tăng tìm tập rút gọn khi bổ sung tập đối tượng. Phần V trình bày kết quả thử nghiệm các thuật toán. Cuối cùng là kết luận và định hướng nghiên cứu tiếp theo.

II. MỘT SỐ KHÁI NIỆM CƠ BẢN

Phần này trình bày một số khái niệm cơ bản về mô hình tập thô dung sai do Kryszkiewicz [2] đề xuất.

Bảng quyết định là một cặp $DS = \big(U, C \cup \{d\}\big)$ trong đó U là tập hữu hạn và khác rỗng của các đối tượng, C là tập hữu hạn và khác rỗng của các thuộc tính điều kiện, d là thuộc tính quyết định. Mỗi thuộc tính $a \in C$ xác định một ánh xạ $a : U \to V_a$ với V_a là tập giá trị của thuộc tính $a \in C$. Nếu V_a chứa giá trị thiếu (missing value) thì DS được gọi là bảng quyết định không đầy đủ, còn ngược lại là đầy đủ. Giá trị thiếu được biểu diễn là $*$. Khi đó, bảng quyết định không đầy đủ được biểu diễn bởi $IDS = \big(U, C \cup \{d\}\big)$ với $* \notin V_d$.

Xét bảng quyết định không đầy đủ $IDS = \big(U, C \cup \{d\}\big)$. Với mỗi tập con thuộc tính $P \subseteq C$, ta định nghĩa một quan hệ nhị phân trên U như sau: $SIM(P) = \big\{(u, v) \in U \times U\big\}$ sao cho với mọi $a \in P$, ta có $a(u) = a(v)$ hoặc $a(u) = *$ hoặc $a(v) = *$, với $a(u)$ là giá trị thuộc tính a tại đối tượng u.

Quan hệ $SIM(P)$ 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, $SIM(P) = \cap_{a \in P} SIM\big(\{a\}\big)$.

Với $u \in U$, $S_P(u) = \big\{v \in U \mid (u, v) \in SIM(P)\big\}$ 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 $SIM(P)$. Trường hợp đặc biệt, nếu $P = \varnothing$ thì $S_\varnothing(u) = U$.

Với $P \subseteq C$ và $X \subseteq U$, tập *P-xấp xỉ dưới* của X là $\underline{P}X = \big\{u \in U \mid S_P(u) \subseteq X\big\} = \big\{u \in X \mid S_P(u) \subseteq X\big\}$, tập *P-xấp xỉ trên* của X là $\overline{P}X = \big\{u \in U \mid S_P(u) \cap X \neq \varnothing\big\} = \bigcup \big\{S_P(u) \mid u \in U\big\}$, *B-miền biên* của X là tập $BN_P(X) = \overline{P}X - \underline{P}X$. Cặp $\langle \underline{P}X, \overline{P}X \rangle$ được gọi là mô hình tập thô dung sai. Với các tập xấp xỉ như vậy, ta gọi *P-miền dương* đối với D là tập $POS_P\big(\{d\}\big) = \bigcup_{X \in U/\{d\}} (\underline{P}X)$.

Xét bảng quyết định không đầy đủ $IDS = \big(U, C \cup \{d\}\big)$. Với $P \subseteq C$ và $u \in U$, $\partial_P(u) = \big\{d(v) \mid v \in S_P(u)\big\}$ được gọi là hàm quyết định suy rộng của IDS. Nếu $|\partial_C(u)| = 1$ với mọi $u \in U$ thì IDS là nhất quán, trái lại IDS là không nhất quán. Theo định nghĩa miền dương, IDS nhất quán khi và chỉ khi $POS_C\big(\{d\}\big) = U$, trái lại IDS là không nhất quán.

Định nghĩa 1: Cho trước bảng quyết định không đầy đủ $IDS = \big(U, C \cup \{d\}\big)$ với $U = \{u_1, u_2, \ldots, u_n\}$ và $P \subseteq C$.

Khi đó, ma trận dung sai của quan hệ $SIM(P)$, ký hiệu là $M(P) = [p_{i,j}]_{n\times n}$, được định nghĩa như sau:

$$M(P) = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix},$$

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

Với việc biểu diễn quan hệ dung sai $SIM(P)$ bằng ma trận dung sai $M(P)$, ta có $u_i \in U$ với mọi $i \in \{1, \ldots, n\}$, $S_P(u_i) = \{u_j \in U \mid p_{i,j} = 1\}$ và $|S_P(u_i)| = \sum_{j=1}^{n} p_{i,j}$.

Với $P, Q \subseteq C$, và $u \in U$ ta có $S_{P\cup Q}(u) = S_P(u) \cap S_Q(u)$. Giả sử $M(P) = [p_{i,j}]_{n\times n}$ và $M(Q) = [q_{i,j}]_{n\times n}$ lần lượt là hai ma trận dung sai của $SIM(P)$ và $SIM(Q)$, khi đó ma trận dung sai trên tập thuộc tính $S = P \cup Q$ là $M(S) = M(P \cup Q) = [s_{i,j}]_{n\times n}$, với $s_{i,j} = p_{i,j}q_{i,j}$.

Xét bảng quyết định không đầy đủ $IDS = \big(U, C \cup \{d\}\big)$ với $U = \{u_1, u_2, \ldots, u_n\}$, $P \subseteq C$, và $X \subseteq U$. Giả sử tập đối tượng X được biểu diễn bằng một véc-tơ một chiều $X = (x_1, x_2, \ldots, x_n)$ trong đó $x_i = 1$ nếu $u_i \in X$ và $x_i = 0$ nếu $u_i \notin X$, khi đó $\underline{P}X = \big\{u_i \in U \mid p_{i,j} \le x_j, j = 1, \ldots, n\big\}$ và $\overline{P}X = \big\{u_i \in U \mid p_{i,j}x_j \ne \varnothing, j = 1, \ldots, n\big\}$.

III. XÂY DỰNG CÔNG THỨC TÍNH KHOẢNG CÁCH KHI BỔ SUNG TẬP ĐỐI TƯỢNG

Trong công trình [3], các tác giả đã xây dựng độ đo khoảng cách giữa các tập thuộc tính trong bảng quyết định không đầy đủ. Trong phần này, chúng tôi xây dựng công thức tính độ đo khoảng cách trong [3] với các trường hợp bổ sung một đối tượng và bổ sung tập đối tượng vào bảng quyết định không đầy đủ. Các công thức này là cơ sở để xây dựng thuật toán gia tăng tìm tập rút gọn được trình bày ở phần IV.

Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \ldots, u_n\}$. Khi đó, khoảng cách giữa hai tập thuộc tính C và $C \cup \{d\}$ được xác định như sau [2]:

$$D(C, C\cup\{d\}) = \frac{1}{n^2}\sum_{i=1}^{n}\Big(\big|S_C(u_i)\big| - \big|S_C(u_i) \cap S_{\{d\}}(u_i)\big|\Big). \quad (1)$$

Giả sử $M(C) = [c_{i,j}]_{n\times n}$, $M(\{d\}) = [d_{i,j}]_{n\times n}$ tương ứng là ma trận dung sai trên C và $\{d\}$. Đặt

$$\gamma(n,m) = \sum_{i=1}^{n}\sum_{j=1}^{m}\big(c_{i,j} - c_{i,j}d_{i,j}\big), \quad (2)$$

Khi đó, độ đo khoảng cách được tính bởi

$$\begin{aligned} D(C, C\cup\{d\}) &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(c_{i,j} - c_{i,j}d_{i,j}) \\ &= \frac{1}{n^2}\gamma(n,n). \end{aligned} \quad (3)$$

1. Công thức tính khoảng cách khi bổ sung một đối tượng

Mệnh đề 1: Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \ldots, u_n\}$. Giả sử đối tượng u được bổ sung vào U. Đặt $M_{U\cup\{u\}}(C) = [c_{i,j}]_{(n+1)\times(n+1)}$ và $M_{U\cup\{u\}}(\{d\}) = [d_{i,j}]_{(n+1)\times(n+1)}$ tương ứng là ma trận dung sai trên C và $\{d\}$ với $S_C(u) = \big\{u_j \in U \mid c_{n+1,j} = 1\big\}$. Khi đó, công thức tính khoảng cách là

$$\begin{aligned} D_{U\cup\{u\}}(C, C\cup\{d\}) = \Big(\frac{n}{n+1}\Big)^2 D_U(C, C\cup\{d\}) + \\ \frac{2}{(n+1)^2}\sum_{i=1}^{n+1}\big(c_{n+1,i} - c_{n+1,i}d_{n+1,i}\big). \end{aligned} \quad (4)$$

Chứng minh: Đặt $k = \big|S_C(u)\big| - \big|S_C(u) \cap S_{\{d\}}(u)\big|$. Từ công thức (3) tính độ đo khoảng cách ta có

$$\begin{aligned} D_{U\cup\{u\}}(C, C\cup\{d\}) &= \frac{1}{(n+1)^2}\Big(\gamma(n, n+1) + k\Big) \\ &= \frac{1}{(n+1)^2}\Big(\gamma(n,n) + k + \\ &\qquad \sum_{i=1}^{n}\big(c_{i,n+1} - c_{i,n+1}d_{i,n+1}\big)\Big) \\ &= \frac{1}{(n+1)^2}\Big(\gamma(n,n) + 2k\Big). \end{aligned} \quad (5)$$

Mặt khác ta có $\gamma(n,n) = n^2 D_U\big(C, C\cup\{d\}\big)$ theo công thức (3). Từ đó ta có kết quả như công thức (4). □

2. Công thức tính khoảng cách khi bổ sung tập đối tượng

Trên cơ sở mệnh đề 1, chúng tôi xây dựng công thức tính khoảng cách trong trường hợp bổ sung tập đối tượng bởi mệnh đề 2 như sau:

Mệnh đề 2: Cho bảng quyết định không đầy đủ $IDS = (U, C \cup \{d\})$ với $U = \{u_1, u_2, \ldots, u_n\}$. Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_{n+1}, u_{n+2}, \ldots, u_{n+s}\}$ được bổ sung vào U với $s \ge 2$, đặt $M_{U\cup\Delta U}(C) = [c_{i,j}]_{(n+s)\times(n+s)}$ và $M_{U\cup\Delta U}(\{d\}) = [d_{i,j}]_{(n+s)\times(n+s)}$ tương ứng là ma trận dung sai trên C và $\{d\}$. Khi đó, công thức tính khoảng cách là

$$\begin{aligned} D_{U\cup\Delta U}(C, C\cup\{d\}) = \Big(\frac{n}{n+s}\Big)^2 D_U(C, C\cup\{d\}) + \\ \frac{2}{(n+s)^2}\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}(c_{i,j} - c_{i,j}d_{i,j}). \end{aligned} \quad (6)$$

Chứng minh: Ký hiệu $D_1, D_2, \ldots, D_s$ tương ứng là khoảng cách giữa C và $C \cup \{d\}$. Khi thêm lần lượt các đối tượng $u_{n+1}, u_{n+2}, \ldots, u_{n+s}$ vào U, và D_0 là khoảng cách giữa C và $C \cup \{d\}$ trên tập đối tượng ban đầu U. Khi bổ

sung đối tượng u_{n+1} vào U, theo công thức (4) của mệnh đề 1 ta có

$$D_1 = \left(\frac{n}{n+1}\right)^2 D_0 + \frac{2}{(n+1)^2}\sum_{j=1}^{n+1}\left(c_{n+1,j} - c_{n+1,j}d_{n+1,j}\right).$$

Khi bổ sung đối tượng u_{n+2} vào U, ta có

$$\begin{aligned} D_2 &= \left(\frac{n+1}{n+2}\right)^2 D_1 + \frac{2}{(n+2)^2}\sum_{j=1}^{n+2}\left(c_{n+2,j} - c_{n+2,j}d_{n+2,j}\right) \\ &= \left(\frac{n}{n+2}\right)^2 D_0 + \frac{2}{(n+2)^2}\sum_{j=1}^{n+1}\left(c_{n+1,j} - c_{n+1,j}d_{n+1,j}\right) + \\ &\quad \frac{2}{(n+2)^2}\sum_{j=1}^{n+2}\left(c_{n+2,j} - c_{n+2,j}d_{n+2,j}\right). \end{aligned}$$

Tính toán tương tự, khi bổ sung đối tượng u_{n+s} vào U, ta có

$$D_s = \left(\frac{n}{n+s}\right)^2 D_0 + \frac{2}{(n+s)^2}A_s,$$

trong đó

$$A_s = \sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{i,j} - c_{i,j}d_{i,j}\right).$$

Từ đó ta có

$$D_s = \left(\frac{n}{n+s}\right)^2 D_0 + \frac{2}{(n+s)^2}\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{i,j} - c_{i,j}d_{i,j}\right)$$

hay

$$\begin{aligned} D_{U\cup\Delta U}\left(C, C\cup\{d\}\right) &= \left(\frac{n}{n+s}\right)^2 D_U\left(C, C\cup\{d\}\right) + \\ &\quad \frac{2}{(n+s)^2}\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}\left(c_{i,j} - c_{i,j}d_{i,j}\right). \end{aligned}$$

□

IV. THUẬT TOÁN GIA TĂNG LỌC – ĐÓNG GÓI TÌM TẬP RÚT GỌN KHI BỔ SUNG TẬP ĐỐI TƯỢNG

Trong công trình [3], các tác giả đã đề xuất thuật toán heuristic tìm tập rút gọn của bảng quyết định không đầy đủ sử dụng độ đo khoảng cách theo hướng tiếp cận lọc. Với cách tiếp cận này, tập rút gọn tìm được là tập thuộc tính nhỏ nhất bảo toàn độ đo khoảng cách ban đầu, việc đánh giá độ chính xác phân lớp được thực hiện sau khi tìm được tìm rút gọn. Trong mục này, dựa trên công thức tính khoảng cách ở mệnh đề 2, chúng tôi xây dựng thuật toán gia tăng tìm tập rút gọn sử dụng độ đo khoảng cách. Thuật toán đề xuất theo hướng tiếp cận lai ghép lọc – đóng gói, trong đó giai đoạn lọc tìm các ứng viên cho tập rút gọn mỗi khi bổ sung thuộc tính có độ quan trọng lớn nhất; giai đoạn đóng gói tìm tập rút gọn có độ chính xác phân lớp cao nhất. Trước hết, chúng tôi trình bày định nghĩa tập rút gọn và độ quan trọng của thuộc tính dựa trên khoảng cách.

Định nghĩa 2 ([3]): Cho bảng quyết định không đầy đủ $IDS = (U, C\cup\{d\})$ với $B \subseteq C$. Nếu

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

thì B là một tập rút gọn của C dựa trên khoảng cách.

Định nghĩa 3 ([3]): Cho bảng quyết định không đầy đủ $IDS = (U, C\cup\{d\})$ với $B \subset C$ và $b \in C - B$. Độ quan trọng của thuộc tính b đối với B được định nghĩa bởi

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

Độ quan trọng $SIG_B(b)$ đặc trưng cho chất lượng phân lớp của thuộc tính b đối với thuộc tính quyết định $\{d\}$ và được sử dụng làm tiêu chuẩn lựa chọn thuộc tính cho thuật toán tìm tập rút gọn.

Mệnh đề 3: Cho bảng quyết định không đầy đủ $IDS = (U, C\cup\{d\})$ với $U = \{u_1, u_2, \ldots, u_n\}$, $B \subseteq C$ là tập rút gọn dựa trên khoảng cách. Giả sử tập đối tượng gồm s phần tử $\Delta U = \{u_{n+1}, u_{n+2}, \ldots, u_{n+s}\}$ được bổ sung vào U với $s \geq 1$. Khi đó, nếu $S_B(u_{n+i}) \subseteq S_{\{d\}}, (u_{n+i})$, $i = 1, 2, \ldots, s$, thì B là tập rút gọn của $IDS_1 = (U\cup\Delta U, C\cup\{d\})$.

Chứng minh: Giả sử $M_{U\cup\Delta U}(C) = [c_{i,j}]_{(n+s)\times(n+s)}$ và $M_{U\cup\Delta U}(B) = [b_{i,j}]_{(n+s)\times(n+s)}$ lần lượt là ma trận tương đương mờ trên C và B của IDS_1.

Nếu $S_B(u_{n+i}) \subseteq S_{\{d\}}(u_{n+i})$, với mọi $i \in \{1, 2, \ldots, s\}$, thì $S_C(x_{n+i}) \subseteq S_B(x_{n+i}) \subseteq S_{\{d\}}(x_{n+i})$. Khi đó ta có các kết quả sau:

1) Với mọi $i \in \{n+1, n+2, \ldots, n+s\}$ và $j \in \{1, 2, \ldots, i\}$, từ $S_B(u_i) \subseteq S_{\{d\}}(u_i)$ suy ra $b_{i,j} \leq d_{i,j}$, hay $b_{i,j} - b_{i,j}d_{i,j} = b_{i,j} - b_{i,j} = 0$. Từ đó ta có $\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}(b_{i,j} - b_{i,j}d_{i,j}) = 0$. Theo mệnh đề 2, ta có

$$D_{U\cup\Delta U}(B, B\cup\{d\}) = \left(\frac{n}{n+s}\right)^2 D_U(B, B\cup\{d\}). \quad (7)$$

2) Tương tự, với mọi $i \in \{n+1, \ldots, n+s\}$ và $j \in \{1, 2, \ldots, i\}$, từ $S_C(u_i) \subseteq S_{\{d\}}(u_i)$ suy ra $c_{i,j} \leq d_{i,j}$, hay $c_{i,j} - c_{i,j}d_{i,j} = c_{i,j} - c_{i,j} = 0$. Từ đó ta có $\sum_{i=n+1}^{n+s}\sum_{j=1}^{i}(c_{i,j} - c_{i,j}d_{i,j}) = 0$. Theo mệnh đề 2, ta có

$$D_{U\cup\Delta U}(C, C\cup\{d\}) = \left(\frac{n}{n+s}\right)^2 D_U(C, C\cup\{d\}). \quad (8)$$

Mặt khác, do B là tập rút gọn của IDS nên ta có $D_U(B, B\cup\{d\}) = D_U(C, C\cup\{d\})$. Từ (7) và (8) suy ra $D_{U\cup\Delta U}(B, B\cup\{d\}) = D_{U\cup\Delta U}(C, C\cup\{d\})$. Hơn nữa, với mọi $b \in B$, $D_U\left(B-\{b\}, (B-\{b\})\cup\{d\}\right) \neq D_U\left(C, C\cup\{d\}\right)$. Từ (7) và (8) suy ra, với mọi $b \in B$, $D_{U\cup\Delta U}\left(B-\{b\}, (B-\{b\})\cup\{d\}\right) \neq D_{U\cup\Delta U}\left(C, C\cup\{d\}\right)$. Do đó, B là tập rút gọn của $IDS_1 = \left(U\cup\Delta U, C\cup\{d\}\right)$. □

Thuật toán 1: Thuật toán IDS_IFW_AO

Input:

- Bảng quyết định không đầy đủ $IDS = \left(U, C \cup \{d\}\right)$ với $U = \{u_1, u_2, \ldots, u_n\}$;
- Tập rút gọn $B \subseteq C$;
- Các ma trận dung sai $M_U(B) = \left[b_{i,j}\right]_{n\times n}$, $M_U(C) = \left[c_{i,j}\right]_{n\times n}$, $M_U(\{d\}) = \left[d_{i,j}\right]_{n\times n}$;
- Tập đối tượng bổ sung $\Delta U = \{u_{n+1}, u_{n+2}, \ldots, u_{n+s}\}$.

Output:

- Tập rút gọn B_{best} của $IDS_1 = \left(U \cup \Delta U, C \cup \{d\}\right)$.

Begin

Bước 1: Khởi tạo.

1: $T = \varnothing$. *// Chứa các ứng viên tập rút gọn tốt nhất.*

2: Tính các ma trận dung sai trên tập đối tượng $U \cup \Delta U$:

$$M_{U\cup\Delta U}(B) = \left[b_{i,j}\right]_{(n+s)\times(n+s)},$$
$$M_{U\cup\Delta U}(\{d\}) = \left[d_{i,j}\right]_{(n+s)\times(n+s)}.$$

Bước 2: Kiểm tra tập đối tượng bổ sung.

3: Đặt $X = \Delta U$.

4: **For** $i = 1$ **to** s **do**

5: **If** $S_B\left(u_{n+i}\right) \subseteq S_{\{d\}}\left(u_{n+i}\right)$ **then** $X := X - \{u_{n+i}\}$.

6: **End**

7: **If** $X = \varnothing$ **then Return** B.

8: Đặt $\Delta U = X$ và $s = |\Delta U|$. *// Gán lại tập đối tượng.*

Bước 3: Thực hiện thuật toán tìm tập rút gọn.

9: Tính các khoảng cách ban đầu: $D_U\left(B, B \cup \{d\}\right)$ và $D_U\left(C, C \cup \{d\}\right)$.

10: Tính các khoảng cách: $D_{U\cup\Delta U}\left(B, B \cup \{d\}\right)$ và $D_{U\cup\Delta U}\left(C, C \cup \{d\}\right)$ theo (6).

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

11: **While** $D_{U\cup\Delta U}\left(B, B \cup \{d\}\right) \neq D_{U\cup\Delta U}\left(C, C \cup \{d\}\right)$ **do**

12: **For each** $a \in C - B$ **do**

13: Tính $D_{U\cup\Delta U}\left(B \cup \{a\}, B \cup \{a\} \cup \{d\}\right)$ theo (6).

14: Tính

$$SIG_B(a) = D_{U\cup\Delta U}\left(B, B \cup \{d\}\right) - D_{U\cup\Delta U}\left(B \cup \{a\}, B \cup \{a\} \cup \{d\}\right).$$

15: **End**

16: Chọn $a \in C - B$ sao cho

$$SIG_B(a_m) = \max_{a\in C-B}\{SIG_B(a)\}.$$

17: $B = B \cup \{a_m\}$.

18: $T = T \cup B$.

19: **End**

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

20: Đặt $t = |T|$.

// T là tập hợp thuộc tính được chọn, nghĩa là $T = \left\{B \cup \left\{a_{i_1}\right\}, B \cup \left\{a_{i_1}, a_{i_2}\right\}, \ldots, B \cup \left\{a_{i_1}, a_{i_2}, \ldots, a_{i_t}\right\}\right\}$, và t là số phần tử của T.

21: Đặt $T_1 = B \cup \left\{a_{i_1}\right\}$, $T_2 = B \cup \left\{a_{i_1}, a_{i_2}\right\}$, ..., và $T_t = B \cup \left\{a_{i_1}, a_{i_2}, \ldots, a_{i_t}\right\}$.

22: **For** $i = 1$ **to** t

23: Tính độ chính xác phân lớp trên T_i bằng một bộ phân lớp sử dụng phương pháp 10-fold.

24: **End**

25: $B_{\text{best}} = T_{jo}$ với T_{jo} có độ chính xác phân lớp lớn nhất.

26: **Return** B_{best}.

Dựa trên mệnh đề 3, thuật toán gia tăng lọc – đóng gói tìm tập rút gọn trong bảng quyết định không đầy đủ sử dụng khoảng cách khi bổ sung tập đối tượng ΔU được mô tả như trong thuật toán 1.

Giả sử $|C|$, $|U|$, $|\Delta U|$ tương ứng là số thuộc tính điều kiện, số đối tượng và số đối tượng bổ sung thêm. Ở câu lệnh 2, độ phức tạp tính ma trận dung sai $M_{U\cup\Delta U}(B)$ khi biết $M_U(B)$ là $O\left(|\Delta U| * (|U| + |\Delta U|)\right)$. Độ phức tạp của vòng lặp **For** ở câu lệnh số 4 là $O\left(|\Delta U| * (|U| + |\Delta U|)\right)$. Trong trường hợp tốt nhất, thuật toán kết thúc ở câu lệnh 7 (tập rút gọn không thay đổi). Khi đó, độ phức tạp thuật toán IDS_IFW_AO là $O\left(|\Delta U| * (|U| + |\Delta U|)\right)$.

Ngược lại, xét vòng lặp **While** từ câu lệnh 11 đến 19, để tính $SIG_B(a)$ ta phải tính $D_{U+\Delta U}\left(B \cup \{a\}, B \cup \{a\} \cup \{d\}\right)$ vì $D_{U\cup\Delta U}\left(B, B \cup \{d\}\right)$ đã được tính ở bước trước. Độ phức tạp tính gia tăng $D_{U+\Delta U}\left(B \cup \{a\}, B \cup \{a\} \cup \{d\}\right)$ là $O\left(|\Delta U| * (|U| + |\Delta U|)\right)$. Do đó, độ phức tạp của vòng lặp **While** là $O\left(\left(|C| - |B|\right)^2 |\Delta U| * (|U| + |\Delta U|)\right)$ và độ phức tạp của giai đoạn lọc trong trường hợp xấu nhất là $O\left(\left(|C| - |B|\right)^2 |\Delta U| * (|U| + |\Delta U|)\right)$. Giả sử độ phức tạp của bộ phân lớp là $O(T)$, khi đó độ phức tạp của giai đoạn đóng gói là $O\left((|C| - |B|) * T\right)$. Vì vậy, độ phức tạp của thuật toán IDS_IFW_AO là $O\left(\left(|C| - |B|\right)^2 * |\Delta U| * (|U| + |\Delta U|)\right) + O\left((|C| - |B|) * T\right)$.

Nếu thực hiện thuật toán không gia tăng lọc – đóng gói trực tiếp trên bảng quyết định có số đối tượng $U \cup \Delta U$, khi đó độ phức tạp là $O\left(|C|^2 * (|U| + |\Delta U|)^2\right) + O\left(|C| * T\right)$. Do đó, thuật toán gia tăng IDS_IFW_AO giảm thiểu đáng kể độ phức tạp thời gian thực hiện, đặc biệt trong trường hợp $|U|$ lớn hoặc $|B|$ lớn.

V. THỬ NGHIỆM

Mục tiêu của thử nghiệm là đánh giá tính hiệu quả của thuật toán gia tăng lọc – đóng gói IDS_IFW_AO với thuật toán gia tăng lọc IARM-I [17] về số lượng thuộc tính tập

Bảng I
BỘ DỮ LIỆU THỬ NGHIỆM THUẬT TOÁN IDS_IFW_AO

Tập dữ liệu	**Số đối tượng**	**Ban đầu**	**Gia tăng**	**Số thuộc tính**	**Số lớp QĐ**
Audiology	226	111	115	69	24
Soyblarge	307	152	155	35	2
CV. Record	435	215	220	16	2
Arrhyth	452	227	225	279	16
Anneal	798	398	400	38	6
Advers.	3279	1639	1640	1558	2

rút gọn và độ chính xác của mô hình phân lớp. IARM-I là thuật toán gia tăng lọc tìm tập rút gọn của bảng quyết định không đầy đủ trong trường hợp bổ sung tập đối tượng sử dụng miền dương. Việc thử nghiệm được thực hiện trên 6 tập dữ liệu mẫu lấy từ kho dữ liệu UCI [23] được mô tả ở Bảng I, đây là các tập dữ liệu thiếu giá trị (missing value) trên miền giá trị thuộc tính điều kiện. Mỗi tập dữ liệu được chia thành hai phần xấp xỉ bằng nhau: tập dữ liệu ban đầu (cột 3 Bảng I) ký hiệu là U_0, và tập dữ liệu gia tăng (cột 4 Bảng I). Tập dữ liệu gia tăng được chia thành 5 phần bằng nhau, ký hiệu lần lượt là U_1, U_2, U_3, U_4 và U_5.

Để tiến hành thử nghiệm hai thuật toán IDS_IFW_AO và IARM-I, trước hết chúng tôi thực hiện hai thuật toán trên tập dữ liệu ban đầu (coi tập dữ liệu ban đầu là tập gia tăng). Tiếp theo, thực hiện hai thuật toán khi lần lượt bổ sung từ phần thứ nhất đến phần thứ năm của tập dữ liệu gia tăng. Chúng tôi sử dụng bộ phân lớp C4.5 để tính độ chính xác phân lớp của hai thuật toán và phương pháp kiểm tra chéo 10-fold. Công cụ thực hiện thử nghiệm là Matlab R2016a. Môi trường thử nghiệm là máy tính PC với cấu hình Intel(R) Core(TM) 2 i3 − 2120 CPU, 3,3 GHz và 4 GB bộ nhớ.

Bảng II trình bày kết quả so sánh về số lượng thuộc tính tập rút gọn và độ chính xác phân lớp của hai thuật toán IDS_IFW_AO và IARM-I. Ký hiệu U là tập đối tượng đưa vào để thử nghiệm, N là số đối tượng đưa vào, S là tổng số đối tượng đưa vào, $|R|$ là số thuộc tính tập rút gọn, Acc là độ chính xác. Kết quả Bảng II cho thấy, độ chính xác phân lớp của IDS_IFW_AO cao hơn IARM-I trên hầu hết các tập dữ liệu. Hơn nữa, số thuộc tính tập rút gọn của IDS_IFW_AO nhỏ hơn khá nhiều so với số thuộc tính tập rút gọn của IARM-I, đặc biệt trên tập rút gọn có số thuộc tính lớn như Advertisements. Do đó, tính khái quát hóa của tập luật phân lớp trên tập rút gọn của IDS_IFW_AO tốt hơn so với IARM-I.

Bảng III trình bày kết quả so sánh thời gian thực hiện hai thuật toán IDS_IFW_AO và IARM-I (tính bằng giây). Trong Bảng III, ký hiệu t là thời gian thực hiện, ts là tổng thời gian thực hiện. Kết quả Bảng III cho thấy, thời gian thực hiện của IDS_IFW_AO lớn hơn IARM-I trên tất cả các tập dữ liệu. Nguyên nhân là IDS_IFW_AO mất thêm chi phí thời gian thực hiện bộ phân lớp trong giai đoạn đóng gói.

Bảng II
SỐ LƯỢNG THUỘC TÍNH TẬP RÚT GỌN VÀ ĐỘ CHÍNH XÁC CỦA IDS_IFW_AO VÀ IARM-I

Tập dữ liệu	U	N	S	**IDS_IFW_AO**		**IARM-I**	
				$\|R\|$	Acc	$\|R\|$	Acc
Audiology	U_0	111	111	5	76, 18	8	74, 29
	U_1	23	134	5	76, 18	9	75, 12
	U_2	23	157	6	81, 26	12	78, 26
	U_3	23	180	6	81, 26	12	78, 26
	U_4	23	203	7	78, 84	14	78, 17
	U_5	23	226	7	78, 84	15	76, 64
Soyblarge	U_0	152	152	5	96, 12	7	95, 46
	U_1	31	183	5	96, 12	7	95, 46
	U_2	31	214	6	96, 72	9	95, 04
	U_3	31	245	7	95, 18	9	95, 04
	U_4	31	276	7	95, 18	10	94, 19
	U_5	31	307	8	94, 58	11	94, 28
CV. Record	U_0	215	215	4	92, 48	9	91, 17
	U_1	44	259	5	92, 76	10	91, 45
	U_2	44	303	7	94, 48	14	92, 28
	U_3	44	347	7	94, 48	14	92, 28
	U_4	44	391	9	94, 12	16	92, 06
	U_5	44	435	9	94, 12	17	92, 88
Arrhyth	U_0	227	227	6	70, 08	14	69, 16
	U_1	45	272	7	72, 45	17	72, 05
	U_2	45	317	7	72, 45	17	72, 05
	U_3	45	362	8	74, 18	21	73, 23
	U_4	45	407	8	74, 18	21	73, 23
	U_5	45	452	9	76, 04	24	73, 08
Anneal	U_0	398	398	4	84, 18	8	84, 06
	U_1	80	478	5	89, 06	8	84, 06
	U_2	80	558	5	89, 06	8	84, 06
	U_3	80	638	6	91, 28	9	88, 48
	U_4	80	718	6	91, 28	9	88, 48
	U_5	80	798	6	91, 28	10	90, 06
Advers.	U_0	1639	1639	12	93, 01	23	92, 16
	U_1	328	1967	14	91, 18	28	90, 48
	U_2	328	2295	14	91, 18	28	90, 48
	U_3	328	2623	17	91, 65	32	91, 17
	U_4	328	2951	18	92, 82	36	92, 06
	U_5	328	3279	19	92, 90	45	92, 46

VI. KẾT LUẬN

Các thuật toán gia tăng tìm tập rút gọn trong bảng quyết định không đầy đủ đã đề xuất đều theo hướng tiếp cận lọc truyền thống. Do đó, tập rút gọn thu được chưa tối ưu về số lượng thuộc tính và độ chính xác phân lớp. Trong bài báo này, chúng tôi xây dựng công thức tính khoảng cách trong công trình [3] với trường hợp bổ sung tập đối tượng. Sử dụng công thức tính khoảng cách đề xuất, chúng tôi xây dựng thuật toán gia tăng lọc – đóng gói IDS_IFW_AO tìm

Bảng III
SỐ LƯỢNG THUỘC TÍNH TẬP RÚT GỌN VÀ ĐỘ CHÍNH XÁC CỦA IDS_IFW_AO VÀ IARM-I

Tập dữ liệu	U	N	S	IDS_IFW_AO		IARM-I	
				t	ts	t	ts
Audiology	U_0	111	111	6, 08	6, 08	5, 82	5, 82
	U_1	23	134	0, 61	6, 69	0, 51	6, 33
	U_2	23	157	0, 35	7, 04	0, 26	6, 59
	U_3	23	180	0, 64	7, 68	0, 42	7, 01
	U_4	23	203	0, 34	8, 02	0, 28	7, 29
	U_5	23	226	0, 44	8, 46	0, 35	7, 64
Soyblarge	U_0	152	152	3, 04	3, 04	2, 86	2, 86
	U_1	31	183	0, 64	3, 68	0, 42	3, 28
	U_2	31	214	0, 34	4, 02	0, 22	3, 52
	U_3	31	245	0, 73	4, 75	0, 54	4, 06
	U_4	31	276	0, 43	5, 18	0, 34	4, 40
	U_5	31	307	0, 68	5, 86	0, 40	4, 80
CV. Record	U_0	215	215	5, 86	5, 86	5, 03	5, 03
	U_1	44	259	0, 56	6, 42	0, 39	5, 42
	U_2	44	303	0, 61	7, 03	0, 46	5, 88
	U_3	44	347	0, 53	7, 56	0, 37	6, 25
	U_4	44	391	0, 47	8, 03	0, 31	6, 56
	U_5	44	435	0, 55	8, 58	0, 32	6, 88
Arrhyth	U_0	227	227	35, 48	35, 48	28, 27	28, 72
	U_1	45	272	1, 58	37, 06	1, 42	30, 14
	U_2	45	317	3, 12	40, 18	2, 26	32, 40
	U_3	45	362	2, 50	42, 68	2, 03	34, 43
	U_4	45	407	1, 36	44, 04	1, 15	35, 58
	U_5	45	452	2, 14	46, 18	1, 84	37, 42
Anneal	U_0	398	398	7, 48	7, 48	6, 05	6, 05
	U_1	80	478	0, 58	8, 06	0, 38	6, 43
	U_2	80	558	0, 81	8, 95	0, 63	7, 06
	U_3	80	638	0, 53	9, 48	0, 34	7, 40
	U_4	80	718	0, 77	10, 25	0, 56	7, 96
	U_5	80	798	0, 80	11, 05	0, 59	8, 55
Advers.	U_0	1639	1639	96, 74	96, 74	82, 05	82, 05
	U_1	328	1967	5, 69	102, 43	4, 84	86, 89
	U_2	328	2295	6, 13	108, 56	5, 18	92, 07
	U_3	328	2623	5, 70	114, 26	4, 26	96, 33
	U_4	328	2951	3, 86	118, 12	2, 54	98, 87
	U_5	328	3279	4, 74	122, 86	2, 98	101, 85

tập rút gọn nhằm giảm thiểu số thuộc tính tập rút gọn và nâng cao độ chính xác phân lớp. Kết quả thử nghiệm trên 06 tập dữ liệu mẫu cho thấy, độ chính xác phân lớp của thuật toán gia tăng lọc – đóng gói IDS_IFW_AO cao hơn thuật toán gia tăng lọc IARM-I. Hơn nữa, số thuộc tính tập rút gọn của IDS_IFW_AO nhỏ hơn khá nhiều so với IARM-I. Do đó, tính khái quát hóa của tập luật phân lớp trên tập rút gọn của IDS_IFW_AO tốt hơn IARM-I. Định hướng nghiên cứu tiếp theo là xây dựng các thuật toán gia tăng lọc – đóng gói tìm tập rút gọn trong trường hợp bổ sung, loại bỏ tập thuộc tính.

TÀI LIỆU THAM KHẢO

[1] Z. Pawlak, *Rough sets: Theoretical aspects of reasoning about data*. London: Kluwer Academic Publisher, 1991.

[2] M. Kryszkiewicz, "Rough set approach to incomplete information systems," *Information sciences*, vol. 112, no. 1-4, pp. 39–49, 1998.

[3] L. G. Nguyen and H. S. Nguyen, "Metric based attribute reduction in incomplete decision tables," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Springer, 2013, pp. 99–110.

[4] J. Demetrovics, V. D. Thi, and N. L. Giang, "A distance-based method for attribute reduction in incomplete decision systems," *Serdica Journal of Computing*, vol. 7, no. 4, pp. 355–374, 2013.

[5] F. Ma, M. Ding, T. Zhang, and J. Cao, "Compressed binary discernibility matrix based incremental attribute reduction algorithm for group dynamic data," *Neurocomputing*, vol. 344, pp. 20–27, 2019.

[6] W. Wei, P. Song, J. Liang, and X. Wu, "Accelerating incremental attribute reduction algorithm by compacting a decision table," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 9, pp. 2355–2373, 2019.

[7] J. Demetrovics, V. D. Thi, and N. L. Giang, "Metric based attribute reduction in dynamic desicion tables," in *Annales Univ. Sci. Budapest., Sect. Comp.*, vol. 42. ELTE, 2014, pp. 157–172.

[8] N. T. L. Huong and N. L. Giang, "Incremental algorithms based on metric for finding reduct in dynamic decision tables," *Journal on Research and Development on Information and Communications Technology*, vol. E-3, no. 9(13), pp. 26–39, 2016.

[9] W. Shu, W. Qian, and Y. Xie, "Incremental approaches for feature selection from dynamic data with the variation of multiple objects," *Knowledge-Based Systems*, vol. 163, pp. 320–331, 2019.

[10] L. Wang, X. Yang, Y. Chen, L. Liu, S. An, and P. Zhuo, "Dynamic composite decision-theoretic rough set under the change of attributes," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 355–370, 2018.

[11] D. Janos, N. T. L. Huong, V. D. Thi, and N. L. Giang, "Metric based attribute reduction method in dynamic decision tables," *Cybernetics and Information Technologies*, vol. 16, no. 2, pp. 3–15, 2016.

[12] G. Lang, Q. Li, M. Cai, T. Yang, and Q. Xiao, "Incremental approaches to knowledge reduction based on characteristic matrices," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 203–222, 2017.

[13] Y. Chuanjian, G. Hao, L. Longshu, and D. Jian, "A unified incremental reduction with the variations of the object for decision tables," *Soft Computing*, vol. 23, no. 15, pp. 6407–6427, 2019.

[14] W. Wei, X. Wu, J. Liang, J. Cui, and Y. Sun, "Discernibility matrix based incremental attribute reduction for dynamic data," *Knowledge-Based Systems*, vol. 140, pp. 142–157, 2018.

[15] Y. Jing, T. Li, J. Huang, H. Chen, and S.-J. Horng, "A group incremental reduction algorithm with varying data values," *International Journal of Intelligent Systems*, vol. 32, no. 9, pp. 900–925, 2017.

[16] D. Zhang, R. Li, X. Tang, and Y. Zhao, "An incremental reduct algorithm based on generalized decision for incomplete decision tables," in *2008 3rd International Conference on Intelligent System and Knowledge Engineering*, vol. 1. IEEE, 2008, pp. 340–344.

[17] W. Shu and W. Qian, "An incremental approach to attribute reduction from dynamic incomplete decision systems in

rough set theory," *Data & Knowledge Engineering*, vol. 100, pp. 116–132, 2015.
[18] W. Shu and H. Shen, "A rough-set based incremental approach for updating attribute reduction under dynamic incomplete decision systems," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2013, pp. 1–7.
[19] J. Yu, L. Sang, and H. Dong, "Based on attribute order for dynamic attribute reduction in the incomplete information system," in *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, 2018, pp. 2475–2478.
[20] W. 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.
[21] ——, "Incremental feature selection based on rough set in dynamic incomplete data," *Pattern Recognition*, vol. 47, no. 12, pp. 3890–3906, 2014.
[22] 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.
[23] "The UCI machine learning repository." [Online]. Available: http://archive.ics.uci.edu/ml/datasets.html

Phạm Minh Ngọc Hà tốt nghiệp Trường Đại học Tổng hợp Hà Nội năm 1994, Thạc sĩ ngành Khoa học Máy tính tại Học viện Kỹ thuật Quân sự năm 2006. Hiện đang công tác tại Học viện tài chính. Hướng nghiên cứu bao gồm: cơ sở dữ liệu, khai phá dữ liệu, lý thuyết tập thô và ứng dụng.

Nguyễn Long Giang tốt nghiệp Trường Đại học Bách khoa Hà Nội năm 1997, Thạc sĩ tại Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội năm 2003, Tiến sĩ tại Viện Công nghệ Thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam năm 2012. Được phong danh hiệu Phó giáo sư năm 2017 ngành Công nghệ Thông tin. Hướng nghiên cứu bao gồm: cơ sở dữ liệu, khai phá dữ liệu và học máy, lý thuyết tập thô và ứng dụng,

Nguyễn Văn Thiện tốt nghiệp Trường Đại học Bách khoa Hà Nội năm 1996, Thạc sỹ tại Trường Đại học Sư phạm Hà Nội năm 2000, Tiến sỹ tại Học viện Khoa học và Công nghệ, Viện Hàn lâm Khoa học và Công nghệ Việt Nam năm 2018. Hiện đang công tác tại Trường Đại học Công nghiệp Hà Nội. Hướng nghiên cứu bao gồm: hệ thống thông tin, cơ sở dữ liệu, khai phá dữ liệu.

Nguyễn Bá Quảng tốt nghiệp Cử nhân ngành Toán tại Trường Đại học Sư phạm Hà Nội 1 năm 1986, Thạc sĩ ngành Công nghệ Thông tin tại Trường Đại học Bách khoa Hà Nội năm 1999. Hướng nghiên cứu bao gồm: cơ sở dữ liệu, khai phá dữ liệu và học máy, lý thuyết tập thô và ứng dụng.