Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XI về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 09-10/8/2018
DOI: 10.15625/vap.2018.00075

VỀ MỘT VẤN ĐỀ THUẬT TOÁN LIÊN QUAN ĐẾN TẬP RÚT GỌN TRONG BẢNG QUYẾT ĐỊNH NHẤT QUÁN

Vũ Đức Thi

Đại học Quốc gia Hà Nội

Email: vdthi@vnu.edu.vn

TÓM TẮT: *Việc nghiên cứu về các tập rút gọn nói chung và các tập rút gọn trong bảng quyết định nhất quán nói riêng được nhiều nhà khoa học thực hiện. Đối với bảng quyết định nhất quán ta đã có một thuật toán có độ phức tạp thời gian tính đa thức tìm một tập rút gọn bất kỳ. Đồng thời việc tìm các thuộc tính dư thừa (thuộc tính không tham gia một tập rút gọn nào) cũng được thực hiện bởi một thuật toán thời gian tính đa thức. Tuy vậy, việc tìm tất cả các tập rút gọn trong bảng quyết định nhất quán là bài toán có độ phức tạp thời gian tính hàm mũ.*

Trong bài báo này, tác giả đưa ra khái niệm tập tựa rút gọn (tập thuộc tính chứa một tập rút gọn nào đó) trong bảng quyết định nhất quán. Tác giả trình bày một bài toán NP- đầy đủ liên quan đến lực lượng của các tập tựa rút gọn. Trên cơ sở kết quả này tác giả chỉ ra rằng việc tìm tập rút gọn có lực lượng bé nhất không thể thực hiện được bằng một thuật toán có thời gian tính đa thức. Có nghĩa là cho đến nay, việc tìm tập này là không khả thi trên hệ thống máy tính.

Keywords:

I. CÁC KHÁI NIỆM CƠ BẢN

Trong các bài toán thực tế, bảng quyết định thường chứa các đối tượng không nhất quán (là các đối tượng bằng nhau trên tập thuộc tính điều kiện nhưng khác nhau trên tập thuộc tính quyết định), gọi là bảng quyết định không nhất quán. Tuy nhiên, tùy thuộc vào lớp bài toán cần giải quyết mà ta có thể chuyển bảng quyết định không nhất quán về bảng quyết định nhất quán qua bước tiền xử lý số liệu bằng cách loại bỏ các đối tượng không nhất quán.

Có thể thấy rằng, trong một bảng quyết định DS bất kỳ, nếu ta không cho phép có hai hàng giá trị giống nhau, thì việc kiểm tra DS có là bảng quyết định nhất quán hay không có thể thực hiện bằng một thuật toán có độ phức tạp tính toán đa thức với kích cỡ của bảng này.

Việc nghiên cứu các tập rút gọn trên bảng quyết định nhất quán liên hệ khá chặt chẽ với lí thuyết cơ sở dữ liệu quan hệ. Trong phần này, chúng tôi đưa ra một vài khái niệm cơ bản cần dùng trong lí thuyết cơ sở dữ liệu quan hệ và lí thuyết tập thô. Các khái niệm này đã được trình bày chi tiết trong [2, 4, 5].

Định nghĩa 1.1. Cho $R=\{a_1,...,a_n\}$ là tập hữu hạn, khác rỗng các thuộc tính, mỗi thuộc tính a_i có miền giá trị là $D(a_i)$. Quan hệ *r* trên *R* là tập các bộ $\{h_1,...,h_m\}$ với $h_j: R\to \bigcup_{a_i\in R} D(a_i), 1\le j\le m$ là một hàm sao cho $h_j(a_i)\in D(a_i)$.

Cho $r=\{h_1,...,h_m\}$ là một quan hệ trên tập thuộc tính $R=\{a_1,...,a_n\}$. Phụ thuộc hàm (*PTH*) trên *R* là một dãy ký tự có dạng $A\to B$ với *A, B* $\subseteq$ *R. PTH* $A\to B$ thỏa mãn quan hệ *r* trên *R* nếu:

$$(\forall h_i,h_j\in r)\Big((\forall a\in A)\big(h_i(a)=h_j(a)\big)\Rightarrow(\forall b\in B)\big(h_i(b)=h_j(b)\big)\Big).$$

Đặt $F_r=\{(A,B): A,B\subseteq R, A\to B\}$ là họ đầy đủ các *PTH* thỏa mãn quan hệ *r*. Ký hiệu $P(R)$ là tập các tập con của *R*. Cho $F\subseteq$ *P(R)xP(R)* . Ta nói rằng *F* là một họ *f* trên *R* nếu với mọi $A,B,C,D\subseteq R$

(1) $(A,A)\in F.$

(2) $(A,B)\in F,(B,C)\in F\Rightarrow(A,C)\in F.$

(3) $(A,B)\in F, A\subseteq C, D\subseteq B\Rightarrow(C,D)\in F.$

(4) $(A,B)\in F,(C,D)\in F\Rightarrow(A\cup C,B\cup D)\in F.$

Rõ ràng là F_r là một họ *f* trên *R*. Nếu *F* là một họ *f* trên *R* thì có một quan hệ *r* trên *R* sao cho $F_r = F$. Ký hiệu F^+ là tập tất cả các *PTH* được dẫn xuất từ *F* bằng việc áp dụng các quy tắc $(1)-(4)$.

Sơ đồ quan hệ *(SĐQH) s* là một cặp $< R, F >$ với *R* là tập thuộc tính và *F* là tập các phụ thuộc hàm trên *R*. Ký hiệu $A^+ = \{a : A \to \{a\} \in F^+\}$, A^+ được gọi là bao đóng của *A* trên *s*. Dễ thấy $A \to B \in F^+$ khi và chỉ khi $B \subseteq A^+$. Tương tự ký hiệu $A_r^+ = \{a : A \to \{a\} \in F^+\}$, $A_r{}^+$ được gọi là bao đóng của *A* trên quan hệ *r*.

Gọi $\mathcal{K} \subseteq P(R)$ là một hệ Sperner trên *R* nếu với mọi $A, B \in \mathcal{K}$ kéo theo $A \not\subset B$. Ở đây P(R) là tập các tập con của R. Với tập $\mathcal{K}$ là một hệ Sperner trên *R*, ta định nghĩa tập $\mathcal{K}^{-1}$ như sau:

$$\mathcal{K}^{-1} = \{A \subset R : (B \in \mathcal{K}) \Rightarrow (B \not\subset A)\}$$

và nếu $(A \subset C) \Rightarrow (\exists B \in \mathcal{K})(B \subseteq C)$.

Dễ thấy $\mathcal{K}^{-1}$ cũng là một hệ Sperner trên *R*. Nếu $\mathcal{K}$ là một hệ Sperner trên *R* đóng vai trò là tập các khóa tối thiểu của quan hệ *r* (hoặc *SĐQH s*) thì $\mathcal{K}^{-1}$ là họ tất cả các tập không phải khóa lớn nhất của *r* (hoặc của *s*), gọi là tập các phản khóa.

Cho *r* là một quan hệ trên *R* và $a \in R$. Đặt $\mathcal{K}_a^r = \{A \subseteq R : A \to \{a\}, \nexists B : (B \to \{a\})(B \subset A)\}$. Khi đó, $\mathcal{K}_a^r$ được gọi là họ các tập tối thiểu của thuộc tính *a* trên *r*.

Định nghĩa 1.2. Hệ thông tin là một bộ bốn S = (U, A, V, f) trong đó *U* là tập hữu hạn, khác rỗng các đối tượng; *A* là tập hữu hạn, khác rỗng các thuộc tính; $V = \bigcup_{a \in A} V_a$ với V_a là tập giá trị của thuộc tính $a \in A$; $f : U \times A \to V_a$ là hàm thông tin, $\forall a \in A, u \in U \quad f(u,a) \in V_a$.

Với mọi $u \in U$, $a \in A$, ta ký hiệu giá trị thuộc tính *a* tại đối tượng *u* là $a(u)$ thay vì $f(u,a)$. Nếu $B = \{b_1, b_2, ..., b_k\} \subseteq A$ là một tập con các thuộc tính thì ta ký hiệu bộ các giá trị $b_i(u)$ bởi $B(u)$. Như vậy, nếu *u* và *v* là hai đối tượng, thì ta viết $B(u) = B(v)$ nếu $b_i(u) = b_i(v)$ với mọi $i = 1, ..., k$.

Định nghĩa 1.3. Bảng quyết định là một hệ thông tin *S = (U, A, V, f)* với A= C $\cup$ D và $C \cap D = \varnothing$. Bảng quyết định S được gọi là nhất quán nếu *D* phụ thuộc hàm vào *C*, tức là với mọi $u, v \in U$, $C(u) = C(v)$ kéo theo $D(u) = D(v)$. Ngược lại thì gọi là không nhất quán hay mâu thuẫn. C được gọi là tập thuộc tính điều kiện và D là tập thuộc tính quyết định

Thông thường D = {d} chứa một thuộc tính

Định nghĩa 1.4. Cho bảng quyết định nhất quán $DS = (U, C \cup D, V, f)$ và tập thuộc tính. P ⊆ C được gọi là tập rút gọn nếu:

- Với mọi cặp đối tượng u, v thì P(u) = P(v) kéo theo D(u) = D(v);
- Với mọi E là tập con thực sự của P thì tồn tại cặp u, v để E(u) = E(v) không kéo theo D(u) = D(v).

Tập rút gọn định nghĩa như trên còn gọi là tập rút gọn Pawlak. Ký hiệu $PRED(C)$ là họ tất cả các tập rút gọn của *C*.

Để phục vụ cho việc giải quyết một bài toán NP-đầy đủ, chúng tôi trình bày khái niệm sau đã có trong [1].

Định nghĩa 1.5. (Tập điểm phủ cạnh - vertex cover set): Cho trước đồ thị không định hướng G = <V,E>, với V là tập đỉnh và E là tập cung. Tập C ⊆ V là tập điểm phủ cạnh nếu ta có C ∩ {a_i, a_j} ≠ ∅ đối với mọi (a_i, a_j) ∈ E

Trước tiên, tác giả trình bày một kết quả cần thiết cho vấn đề này.

Định lí 1.1. *[4] Cho bảng quyết định nhất quán*

$$DS = \left(U, C \cup \{d\}, V, f\right) \text{ với } C = \{c_1, c_2, ..., c_n\},\ U = \{u_1, u_2, ..., u_m\}.$$

Xét quan hệ $r = \{u_1, u_2, ..., u_m\}$ *trên tập thuộc tính* $R = C \cup \{d\}$.

Đặt $\mathcal{E}_r = \{E_{ij} : 1 \le i < j \le m\}$ *với* $E_{ij} = \{a \in R : a(u_i) = a(u_j)\}$

Đặt $\mathcal{M}_d = \{A \in \mathcal{E}_r : d \notin A, \nexists B \in \mathcal{E}_r : d \notin B, A \subset B\}$.

Thì $\mathcal{M}_d = \left(\mathcal{K}_d^r\right)^{-1}$. Ở đây $\mathcal{K}_d^r$ *là họ các tập tối thiểu của thuộc tính* $\{d\}$ *trên quan hệ r.*

2. CÁC KẾT QUẢ

Định nghĩa 2.1. Cho trước *DS = (U, C ∪ {d}, V, f)*, tập B được gọi là tập tựa rút gọn của DS nếu tồn tại một tập rút gọn A của DS sao cho A ⊆ B.

Trước tiên, tác giả đưa ra kết quả sau.

Bổ đề 2.1. *Cho K là hệ Sperner trên C thì tồn tại một bảng quyết định nhất quán:*

$$DS = (U, C \cup \{d\}, V, f) \text{ để } K = (K_d^r)^{-1}$$

Chứng minh:

Giả sử K = { $A_1,\ldots,A_m$ }. Ta xây dựng bảng quyết định DS = (U, C$\cup${d}, V, f) như sau:

U = {u_0, $u_1,\ldots, u_m$} với mọi c $\in$ C : c(u_0) = 0 và d(u_0) = 0. Với mọi i, i = 1,…m và c là phần tử của C. Ta đặt c(u_i) = 0 nếu c $\in$ A_i. Ngược lại c(u_i) = i. Đặt d(u_i) = i. Ở đây R = C $\cup$ {d}.

Đặt $\mathcal{E}_r = \{E_{ij} : 1 \le i < j \le m\}$.

với $E_{ij} = \{a \in R : a(u_i) = a(u_j)\}$.

Đặt $\mathcal{M}_d = \{A \in \mathcal{E}_r : d \notin A, \nexists B \in \mathcal{E}_r : d \notin B, A \subset B\}$.

Có thể thấy M_d = { $A_1,\ldots,A_m$ }. Theo Định lí 1.1, ta có $M_d = (K_d^r)^{-1}$. Như vậy

$$K = (K_d^r)^{-1}.$$

Kết quả đã được chứng minh.

Định lý 2.1. Vấn đề sau là NP- đầy đủ

Cho trước một hệ Sperner K trên R = $\{a_1, a_2, \ldots, a_n\}$, và một số nguyên dương k (k ≤n). Việc xác định có tồn tại hay không một tập A ⊆ R sao cho $|A| \le k$ và mỗi B (B ∈ K) $A \not\subseteq B$.

Chứng minh:

Chọn ngẫu nhiên A sao cho $|A| \le k$ và xác định A không là tập con của mỗi tập B ∈ K. Dễ thấy việc xác định này có thời gian tính đa thức với n và m (Ở đây $|K| = m$). Do đó vấn đề trên thuộc NP.

Chúng ta chọn vấn đề sau [1] là NP - đầy đủ (vấn đề lực lượng của tập điểm phủ cạnh -vertex cover problem).

Cho số k nguyên dương và đồ thị không định hướng G = <V,E>, với V là tập đỉnh và E là tập cung, xác định có một tập điểm phủ cạnh có lực lượng không lớn hơn k.

Chúng ta chứng minh vấn đề này được chuyển về vấn đề của chúng ta bằng một phép biến đổi có thời gian đa thức.

Giả sử G=<V,E> là đồ thị không định hướng và k $\le$ |A|. Đặt R= V, và P = $\{R\backslash\{a_i, a_j\} : (a_i, a_j) \in E\}$. Dễ thấy P là một hệ Sperner trên R. Giả sử P={$B_1,\ldots,B_m$}.

Nếu $|A| \leq k$ và $A \not\subseteq B_i$, với i = 1,...,m, thì do định nghĩa của P ta có $A \cap \{a_i, a_j\} \neq \emptyset$ đối với mọi $(a_i, a_j) \in E$. Do đó A là một tập điểm phủ cạnh của G. Ngược lại A là một tập điểm phủ cạnh của G thì từ định nghĩa của P và định nghĩa của tập điểm phủ cạnh, ta có $A \not\subseteq B_i$, với mọi i = 1,...,m. Do đó $A \not\subseteq B_i$ (với mọi i = 1,...,m) khi và chỉ khi A là một tập điểm phủ cạnh của G.

Kết quả được chứng minh.

Trên cơ sở Bổ đề 2.1, chúng ta có thuật toán thời gian tính đa thức để tìm một bảng quyết định nhất quán từ một hệ Sperner cho trước K sao cho $K_d^{-1} = K$, cho nên với định lý trên chúng ta có kết quả sau.

Hệ quả 2.1. Vấn đề sau là NP - đầy đủ: Cho trước số nguyên dương k và một bảng quyết định nhất quán *DS = (U, C ∪ {d}, V, f)*. Việc xác định có tồn tại hay không một tập tựa rút gọn A của DS mà $|A| \leq k$.

Như chúng ta đã biết, nếu kí pháp lớp bài toán được nhận biết bởi máy Turing tiền định là P và lớp bài toán được nhận biết bởi máy Turing bất định là NP, thì bài toán NP = P hay không là bài toán chưa giải được. Tuy vậy, cho đến nay hầu hết các nhà khoa học đều cho rằng NP khác P.

Từ kết quả trên, chúng ta có kết quả sau.

Hệ quả 2.2. Cho trước bảng quyết định DS = (U, C ∪ {d}, V, f). Khi đó việc tìm tập rút gọn có lực lượng nhỏ nhất của DS không thể thực hiện được bằng một thuật toán có thời gian tính đa thức.

LỜI CÁM ƠN

Nghiên cứu này cảm ơn sự tài trợ của đề tài mã số 01/2018/KCM phối hợp thực hiện giữa Viện CNTT, ĐHQGHN với Học viện Kỹ thuật Mật mã.

TÀI LIỆU THAM KHẢO

[1] Aho A. V., Hofcroft J. E., Ullman J. D. The design and analysis of computer algorithms. Addison - Wesley, Reading, Mass., 1974.

[2] Demetrovics J. and Thi V. D. (1995). "Some remarks on generating Armstrong and inferring functional dependencies relation". *Acta Cybernetica 12*, pp. 167-180.

[3] Nguyen Long Giang, Vu Duc Thi (2011). "Some Problems Concerning Condition Attributes and Reducts in Decision Tables". *Proceeding of the Fifth National Symposium "Fundamental and Applied Information Technology Research" (FAIR)*, Bien Hoa, Dong Nai, pp. 142-152.

[4] Nguyễn Long Giang, Vũ Đức Thi (2011). "Thuật toán tìm tất cả các rút gọn trong bảng quyết định". *Tạp chí Tin học và Điều khiển học*, T.27, S.3, tr. 199-205.

[5] Pawlak Z. (1991). "*Rough sets: Theoretical Aspects of Reasoning About Data"*. Kluwer Academic Publishers.

ON THE COMPUTATIONAL PROBLEM RELATED TO REDUCT IN THE CONSISTENT DECISION TABLES

Vu Duc Thi

ABSTRACT: In this paper, we show the NP- complete problem in the consistent decision tables. This problem is related to reduct in the consistent decision tables. From this result, we show that up to now, there is no polynomial algorithm to find the minimal reduct.