Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00075

MỘT SỐ VẤN ĐỀ VỀ KHAI PHÁ ĐỒ THỊ CON THƯỜNG XUYÊN ĐÓNG

Hoàng Minh Quang[1], Vũ Đức Thi[2], Vũ Thị Lan Anh[1]

[1]Viện Công nghệ thông tin, Viện Hàn lâm Khoa học và Công nghệ Việt Nam.

[2]Viện Công nghệ thông tin, Đại học Quốc gia Hà Nội.

[1]hoangquang@ioit.ac.vn, [2]vdthi@vnu.edu.vn, [1]vlanh@ioit.ac.vn

***TÓM TẮT**: Phân loại hay phân lớp dữ liệu là một phương pháp quan trọng trong khai phá dữ liệu. Có rất nhiều phương pháp phân loại dữ liệu, trong số đó phân loại dữ liệu bằng cây quyết định là một phương pháp phổ biến và hiệu quả. Các thuật toán như ID3 hay C4.5 được ứng dụng trong nhiều ứng dụng để phân loại dữ liệu. Trong bài báo này chúng tôi đề xuất một phương pháp sinh cây quyết định từ bảng quyết định nhất quán không dựa trên hàm thông tin Entropy như các thuật toán ID3 hay C4.5 đang thực hiện mà vẫn mang lại hiệu quả tương đương thậm chí còn tốt hơn.*

***Từ khóa**: phân loại dữ liệu, cây quyết định, rút gọn thuộc tính, rút gọn đối tượng, thuật toán xây dựng cây quyết định.*

I. GIỚI THIỆU

Bảng thông tin [4] biểu diễn dữ liệu đầu vào, thu thập được từ bất kỳ miền nào, chẳng hạn như y học, tài chính hoặc quân sự. Các hàng của một bảng quyết định được gọi là các đối tượng. Các thuộc tính, các cột của bảng quyết định, của các đối tượng được gán giá trị theo một số biến. Có hai loại biến phân biệt: thuộc tính (còn được gọi là thuộc tính điều kiện), quyết định (còn được gọi là thuộc tính quyết định). Thông thường các hệ thống chỉ đòi hỏi các quyết định đơn. Ví dụ nếu bảng thông tin mô tả một bệnh viện, các đối tượng là các bệnh nhân; các thuộc tính là triệu chứng và xét nghiệm; quyết định là bệnh tật. Bảng quyết định là một bảng thông tin và bảng quyết định nhất quán là bảng quyết định khi và chỉ khi tất cả các giá trị của các thuộc tính điều kiện bằng nhau thì các giá trị của các thuộc tính quyết định cũng phải bằng nhau.

Trong lĩnh vực học máy, cây quyết định là một kiểu mô hình dự báo, là một ánh xạ từ các quan sát về một sự vật hiện tượng tới các kết luận về giá trị mục tiêu của sự vật hiện tượng. Mỗi nút trong ứng với một biến, đường nối từ giữa nút trong và nút con của nó thể hiện giá trị cụ thể của biến. Nút lá đại diện cho giá trị dự đoán của biến mục tiêu. Kỹ thuật học máy dùng cây quyết định được gọi là học bằng cây quyết định hay ngắn gọn là cây quyết định. Trong khai phá dữ liệu, cây quyết định mô tả một cấu trúc cây mà các lá đại diện cho các giá trị được phân loại thông qua đường đi là sự kết hợp các thuộc tính điều kiện dẫn đến phân loại đó. Cây quyết định được ứng dụng trong nhiều lĩnh vực như trí tuệ nhân tạo, khoa học nhận dạng đặc biệt trong lĩnh vực học máy, sự thu nhận kiến thức, phân tích quyết định, khám phá tri thức từ các cơ sở dữ liệu, các hệ chuyên gia, các hệ thống hỗ trợ quyết định, lập luận suy dẫn và nhận dạng mẫu.

Trong các ứng dụng sử dụng cây quyết định, tuỳ từng mục đích mà có cách xây dựng cây quyết định khác nhau. Có một số ứng dụng cần xây dựng cây quyết định có không gian lưu trữ tối thiểu, có ứng dụng yêu cầu xây dựng cây quyết định trong thời gian tối thiểu hoặc có ứng dụng yêu cầu đưa ra số nhánh tối thiểu hoặc có ứng dụng yêu cầu đưa ra các đường đi trong cây là tối thiểu... Để đáp ứng được các yêu cầu trong mọi trường hợp thì cần phải tìm xây dựng ra tất cả các cây quyết định và từ đó chọn ra cây quyết định thoả mãn tiêu chí đề ra. Do đó vấn đề tìm cây quyết định tối ưu thuộc lớp bài toán NP-Hard. Một số công trình sử dụng phương pháp tham lam để xây dựng cây quyết định chẳng hạn như ID3 hay C4.5, các thuật toán này sử dụng hàm thông tin Entropy để xác định các thuộc tính được đưa vào cây quyết định trong từng bước của thuật toán đệ quy. Đến nay, ID3 và C4.5 được sử dụng trong hầu hết các ứng dụng xây dựng cây quyết định.

Trong bài báo này, chúng tôi đề xuất một thuật toán xây dựng cây quyết định từ bảng quyết định nhất quán theo phương pháp tham lam không sử dụng hàm Entropy như các thuật toán ID3 hay C4.5 xây dựng các thuật toán không dư thừa của bảng quyết định nhất quán theo cả chiều ngang và chiều dọc của bảng quyết định nhất quán để giảm số thuộc tính cũng như đối tượng về mức nhỏ nhất. Từ đó, chúng tôi xây dựng một thuật toán tìm một rút gọn thuộc tính và sử dụng tập lõi rút gọn cùng với rút gọn thuộc tính để xây dựng cây quyết định rất nhanh chóng và hiệu quả. Các thuật toán được xây dựng đều được thực hiện trong thời gian đa thức. Do được rút gọn cả về chiều ngang và chiều dọc của bảng quyết định nhất quán nên cây quyết định được chúng tôi xây dựng đạt tiêu chí không gian lưu trữ nhỏ, thời gian tính toán cũng đáp ứng được cho cả các bảng quyết định nhất quán cỡ lớn.

II. MỘT SỐ ĐỊNH NGHĨA

Phần này sẽ trình bày một số khái niệm cơ bản của lý thuyết cơ sở dữ liệu quan hệ [1, 2, 6] và lý thuyết tập thô [3, 4, 5]. Xem một bảng quyết định trong lý thuyết tập thô giống như một bảng quan hệ trong cơ sở dữ liệu quan hệ vì các hàng và cột của chúng có chung một số khái niệm về đối tượng và thuộc tính. Dựa trên các kết quả trong lý thuyết cơ sở dữ liệu quan hệ [6], chúng tôi áp dụng các kết quả này vào bảng quyết định nhất quán [7]. Một số định nghĩa và sự kết hợp của hai lý thuyết được sử dụng sẽ được trình bày dưới đây:

Một *hệ thông tin* S là một bộ bốn có thứ tự $S=(U,A,V,f)$ mà U là một tập hữu hạn không rỗng *các đối tượng*, được gọi là *tập vũ trụ*; A là một tập hữu hạn không rỗng *các thuộc tính;* $V=\bigcup_{a\in A}V_a$ là *miền giá trị* của các thuộc tính a; $f:U\times A\to V$ là một hàm toàn thể, mà $f(x,a)\in V_a$ với mọi $a\in A$ và $x\in U$ được gọi là *hàm thông tin*. Hàm $f_x:A\to V$ mà $f_x(a)=f(x,a)$ với mọi $a\in A$ và $x\in U$ sẽ được gọi là *thông tin về* x trong S. Ký hiệu $a(x)=f_x(a)$. Nếu $B=\{b_1,b_2,...,b_k\}\subseteq A$ là tập con các thuộc tính, thì tập $b_i(x)$ được ký hiệu như $B(x)$. Theo đó, nếu x,y là hai đối tượng trong U, thì $B(x)=B(y)$ nếu và chỉ nếu $b_i(x)=b_i(y),\forall i=1,...,k$.

Bảng quyết định là hệ thông tin $S=(U,A,V,f)$, mà $A=C\cup D$ và $C\cap D=\varnothing$. Không mất tính tổng quát, giả sử D chỉ chứa một thuộc tính quyết định d. Theo đó, từ đây xem bảng quyết định $DS=(U,C\cup\{d\},V,f)$, mà $\{d\}\notin C$

Cho $R=\{a_1,...,a_n\}$ là một tập hữu hạn các thuộc tính và cho $D(a_i)$ là tập tất cả các giá trị của các thuộc tính a_i, *quan hệ* r trên R là tập các bộ $\{h_1,...,h_m\}$ mà $h_j:R\to\bigcup_{a_i\in R}D(a_i),1\le j\le m$, là một hàm $h_j(a_i)\in D(a_i)$.

Cho $r=\{h_1,...,h_m\}$ là một quan hệ trên $R=\{a_1,...,a_n\}$. Bất kỳ cặp tập thuộc tính $A,B\subseteq R$ được gọi là *phụ thuộc hàm* trên R, và được ký hiệu $A\to B$ nếu và chỉ nếu:

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

Cho bảng quyết định $DS=(U,C\cup\{d\},V,f)$, $U=\{u_1,...,u_m\}$ là một quan hệ trên $C\cup\{d\}$.

Một bảng quyết định DS là *nhất quán* nếu và chỉ nếu phụ thuộc hàm $C\to\{d\}$ là đúng; nghĩa là cho bất kỳ $x,y\in U$ nếu $C(x)=C(y)$ thì $d(x)=d(y)$. Ngược lại, DS là không nhất quán.

Mọi tập con các thuộc tính $P\subseteq C\cup D$ định ra một *quan hệ bất khả phân biệt*

$IND(P)=\{(u,v)\in U\times U\mid\forall a\in P,f(u,a)=f(v,a)\}$

$IND(P)$ định ra một *phân hoạch* trên U được xác định bởi U/P.

Bất kỳ thành phần $[u]_P=\{v\in U\mid(u,v)\in IND(P)\}$ trong U/P được gọi là một *lớp tương đương*.

Pawlak [3] xác định xấp xỉ trên, xấp xỉ dưới và miền dương dựa trên lớp tương đương như sau:

B-xấp xỉ trên của X là tập $\bar{B}X=\{u\in U\mid[u]_B\cap X\neq\varnothing\}$,

B-xấp xỉ dưới của X là tập $\underline{B}X=\{u\in U\mid[u]_B\subseteq X\}$ với $B\subseteq C$, $X\subseteq U$,

B-vùng biên là tập $BN_B(X)=\bar{B}X\setminus\underline{B}X$,

B-miền dương của D là tập $POS_B(D)=\bigcup_{X\in U/D}(\underline{B}X)$

Một cặp $s=\langle R,F\rangle$, với R là một tập của các thuộc tính và F là một tập các phụ thuộc hàm trên R, được gọi là một *lược đồ quan hệ*. Cho bất kỳ $A\subseteq R$, tập $A^+=\{a:A\to\{a\}\in F^+\}$ được gọi là *đóng* của A trên s. Rõ ràng $A\to B\in F^+$ nếu và chỉ nếu $B\subseteq A^+$. Một cách tương tự, $A_r^+=\{a:A\to\{a\}\in F^+\}$ được gọi là đóng của A trên quan hệ r.

Cho r là một quan hệ, $s=\langle R,F\rangle$ là một lược đồ quan hệ và $A\subseteq R$. Thì A là một *khóa* của r (một khóa của s) nếu $A\to R$ ($A\to R\in F^+$). A là một *khóa tối tiểu* của r (s) nếu A là một khóa của r (s) và bất kỳ tập con thực sự của A không phải là khóa của r (s). Tập tất cả các khóa tối tiểu của r (s) được ký hiệu là K_r (K_s). Một họ $K\subseteq P(R)$ là một *hệ Sperner* trên R nếu cho bất kỳ $A,B\in K$ kéo theo $A\not\subset B$. Rõ ràng K_r (K_s) là các hệ Sperner.

Cho K là một hệ Sperner trên R cũng là tập tất cả các khóa tối tiểu của s. Định nghĩa tập *các phản khóa* của K, ký hiệu K^{-1}, như sau: $K^{-1}=\{A\subset R:(B\in K)\Rightarrow(B\not\subset A)\ and\ if\ (A\subset C)\Rightarrow(\exists B\in K)(B\subseteq C)\}$

Dễ thấy K^{-1} là tập các tập con của R, mà không chứa thành phần nào của K và là lớn nhất. Chúng là tập không khóa lớn nhất. Rõ ràng, K^{-1} cũng là một hệ Sperner.

Cho r là một quan hệ trên R. Ký hiệu $E_r = \{E_{ij} : 1 \le i \le j \le |r|\}$, mà $E_{ij} = \{a \in R : h_i(a) = h_j(a)\}$. Thì E_r được gọi là một *tập bằng nhau* của r. Cho $A_r \in R, A_r^+ = \cap E_{ij}$, nếu tồn tại $E_{ij} \in E_r : A \subseteq E_{ij}$, nếu không $A_r^+ = R$.

Cho $r = \{h_1, ..., h_m\}$ là một quan hệ trên R, E_r tập bằng nhau của r. Đặt

$M_r = \{E_{ij} \in E_r : \forall E_{st} \in E_r : E_{ij} \subseteq E_{st}, E_{ij} \ne E_{st}\}$ với $1 \le i < j \le m, 1 \le s < t \le m$.

M_r được gọi là *hệ bằng nhau cực đại* của r.

$M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$ được gọi là *hệ bằng nhau cực đại* của r đối với thuộc tính quyết định d của bảng quyết định nhất quán DS.

Cho $s = \langle R, F \rangle$ là một lược đồ quan hệ trên R và $a \in R$. Tập $K_a^s = \{A \subseteq R : A \to \{a\}, \nexists B : (B \to \{a\})(B \subset A)\}$ được gọi là một họ các tập tối tiểu của thuộc tính a trên s.

Tương tự, tập $K_a^r = \{A \subseteq R : A \to \{a\}, \nexists B \subseteq R : (B \to \{a\})(B \subset A)\}$ được gọi là một họ các tập tối tiểu của thuộc tính a trên r.

Nếu K là một hệ Sperner trên R cũng như họ các tập tối tiểu của thuộc tính a trên r (hoặc s); nghĩa là $K = K^r$ (hoặc $K = K^s$), thì $K^{-1} = \{K_a^r\}^{-1}$ (hoặc $K^{-1} = \{K_a^s\}^{-1}$) là họ các tập con cực đại của R mà không là họ các tập tối tiểu của thuộc tính a, được xác định:

$$\{K_a^r\}^{-1} = \{A \subseteq R : A \to \{a\} \notin R_r^+, A \subset B \Rightarrow B \to \{a\} \in F_r^+\}$$

$$\{K_a^s\}^{-1} = \{A \subseteq R : A \to \{a\} \notin R_r^+, A \subset B \Rightarrow B \to \{a\} \in F_r^+\}$$

Rõ ràng $R \notin K_a^s$, $R \notin K_a^r$, $\{a\} \in K_a^s$, $\{a\} \in K_a^r$ and K_a^s, K_a^r là các hệ Sperner trên R.

Cho $DS = (U, C \cup \{d\}, V, f)$ là một bảng quyết định. Nếu $B \subseteq C$ thỏa mãn:

1) $POS_B(D) = POS_C(D)$

2) $\forall b \in B, POS_{B-\{b\}}(D) \ne POS_C(D)$

thì B được gọi là *rút gọn* của C.

Nếu DS là một bảng quyết định nhất quán, B là một *rút gọn thuộc tính* của C nếu B thỏa mãn $B \to \{d\}$ và $\forall B' \subset B, B' \not\to \{d\}$. Cho $RED(C)$ là tập tất cả các rút tọn của C. Từ định nghĩa *rút gọn* trên và công thức K_a^r trong định nghĩa *khóa tối tiểu* ta có $RED(C) = K_d^r - \{d\}$ với K_d^r là họ tất cả các tập tối tiểu của thuộc tính $\{d\}$ trên $r = \langle U, C \cup \{d\} \rangle$.

Cho $U = \{u_1, u_2, ..., u_m\}$ là tập vũ trụ trên một bảng quyết định DS. *Ma trận phân biệt* được xác định bởi:

$m_{ij} = \{a \in C : (a(u_i) \ne a(u_j)) \wedge (d(u_i) \ne d(u_j)), d \in D, \forall i, j = 1, 2, ..., m\}$ với m_{ij} là tập tất cả các thuộc tính mà phân loại các đối tượng u_i và u_j vào trong các lớp quyết định khác nhau trong phân hoạch U / D.

Tập tất cả các thuộc tính mà tham gia vào mọi rút gọn của C được gọi là *tập lõi* của C ký hiệu $CORE(C)$. Vì vậy $CORE(C) = \cap RED(C)$ với $RED(C)$ là tập tất cả các rút gọn của C. Tập lõi là tập chứa tất cả các thành phần đơn (tập chỉ có một phần tử) của ma trận phân biệt. Do vậy, $CORE(C) = \{a \in C : (\exists i, j) m_{ij} = \{a\}\}$

III. THUẬT TOÁN XÂY DỰNG CÂY QUYẾT ĐỊNH

Trong phần này, chúng tôi xây dựng một số thuật toán thực hiện trong thời gian đa thức để tìm rút gọn đối tượng và rút gọn thuộc tính và áp dụng thuật toán xây dựng cây quyết định trên rút gọn cả đối tượng và thuộc tính.

Cho một bảng quyết định nhất quán $DS = (U, C \cup \{d\}, V, f)$ với $U = \{u_1, ..., u_m\}$ trên tập thuộc tính $R = C \cup \{d\}$, từ định nghĩa *rút gọn* dựa trên miền dương của Pawlak thì $RED(C) = K_d^r - \{d\}$, nếu ký hiệu $REAT(C)$ tập tất cả các thuộc tính rút gọn của C thì:

$$REAT(C) = \bigcup_{A \in RED(C)} A = \left(\bigcup_{A \in K_d^r} A \right) - \{d\}$$

Thuật toán 1: Tìm tập chứa tất cả các thuộc tính rút gọn của C

Đầu vào: $DS = (U, C \cup \{d\}, V, f)$, $POS_C(\{d\}) = U$, $C = \{c_1, ..., c_n\}$, $U = \{u_1, ..., u_m\}$
Đầu ra : $REAT(C)$

1 Xem xét quan hệ $r = \{u_1, ..., u_m\}$ trên tập thuộc tính $R = C \cup \{d\}$.
2 Bước 1: Tính $E_r = \{A_1, ..., A_t\}$;
3 Bước 2: Tính $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.;
4 Bước 3: Xây dựng $N = R - \bigcap_{B \in M_d} B$;
5 Bước 4: Đặt $REAT(C) = N - \{d\}$

Thuật toán 1 tìm tất cả các *thuộc tính rút gọn* của bảng quyết định nhất quán DS. Thuộc tính rút gọn là thuộc tính có tham gia vào một rút gọn nào đó trong $RED(C)$. Thuật toán 1 này có độ phức tạp thời gian đa thức.

Thuật toán 2: Tìm tập các phản khóa

Đầu vào: Cho $K = \{B_1, ..., B_m\}$ là một hệ Sperner trên U
Đầu ra : K^{-1}

1 Bước 1: Đặt $K^{-1} = \{U - \{a\} : a \in B_1\}$. Rõ ràng $K_1 = \{B_1\}^{-1}$.;
2 Bước $q+1$: $(q < m)$ Giả sử $K_q = F_q \cup \{X_1, ..., X_{t_q}\}$, với X_1, ..., X_{t_q} là thành phần của K chứa B_{q+1} và $F_q = \{A \in K_q : B_{q+1} \not\subseteq A\}$. Với mọi i $(i = 1, ..., t_q)$, xây dựng tập phản khóa của $\{B_{q+1}\}$ trên X_i tương tự như với K_1, là tập cực đại của X_i không chứa B_{q+1}. Ký hiệu A_1^i, ... , $A_{r_i}^i$. Cho

$$K_{q+1} = F_q \cup \{A_p^i : A \in F_q \Rightarrow A_p^i \not\subset A, 1 \le i \le t_q, 1 \le p \le r_i\}.$$

Đặt $K^{-1} = K_m$.

Thuật toán 2 tìm tập chứa tất cả các phản khóa của quan hệ trên DS. Thuật toán này sẽ được ứng dụng trong tìm tất cả các rút gọn của bảng quyết định nhất quán DS.

Thuật toán 3: Tìm một khóa tối tiểu từ tập các phản khóa

Đầu vào: Cho K, H là các hệ Sperner và $C = \{c_1, ..., c_n\} \subseteq U$ mà $H^{-1} = K$ và $\exists B \in K : B \subseteq C$
Đầu ra : $D \in H$

1 Bước 1: Đặt $A(0) = C$;
2 Bước $i+1$: Đặt

$$A(i+1) = \begin{cases} A(i) - \{c_{i+1}\}, & \text{nếu } \forall B \in K : A(i) - \{c_{i+1}\} \not\subseteq B \\ A(i), & \text{ngược lại} \end{cases}$$

Đặt $D = A(n)$.

Thuật toán 3 tìm một khóa tối tiểu từ tập các phản khóa. Trong một vòng lặp để tìm tất cả các khóa tối tiểu của một quan hệ hay một lược đồ quan hệ, khi chưa tìm được tập chứa tất cả các khóa tối tiểu thì mỗi bước sẽ tìm một số khóa tối tiểu từ tập phản khóa của bước trước nó.

Thuật toán 4: Tìm một tập khóa tối tiểu từ tập các phản khóa

Đầu vào: Cho $K = \{B_1, ..., B_k\}$ là một hệ Sperner trên U
Đầu ra : H mà $H^{-1} = K$

1 Bước 1: Theo thuật toán 3 tính A_1, đặt $K(1) = A_1$;
2 Bước $i+1$: Nếu tồn tại một $B \in K_i^{-1}$ sao cho $B \not\subseteq B_j$ $(\forall j : 1 \le j \le k)$, thì theo thuật toán 3 tính A_{i+1}, với $A_{i+1} \in H$, $A_{i+1} \subseteq B$. Đặt $K(i+1) = K(i) \cup A_{i+1}$. Trong trường hợp ngược lại, đặt $H = K(i)$

Thuật toán 4 tìm tập khóa tối tiểu từ tập các phản khóa bằng cách sử dụng một vòng lặp. Tại mỗi bước dựa vào kết quả quả thuật toán 2 để tìm tập phản khóa của khóa tối tiểu ở bước trước sau đó sử dụng thuật toán 3 để tìm tập khóa tối tiểu từ tập phản khóa được sinh ra khi dùng thuật toán 2. Vòng lặp cứ thế tiếp diễn cho đến khi tập phản khóa bằng với tập phản khóa ban đầu. Thuật toán 4 có độ phức tạp thời gian thuộc lớp NP-Complete.

Một rút gọn đối tượng của bảng quyết định nhất quán $DS = (U, C \cup \{d\}, V, f)$ là một bảng quyết định nhất quán $DS' = (U', C \cup \{d\}, V, f)$, với $RED(C) = RED_U(C)$ và:

1) $U' \subseteq U$,

2) $RED_U(C) = RED_{U'}(C)$,

3) $RED_U(C) \neq RED_{U'-\{u\}}(C), \forall u \in U'$.

Thuật toán 5: Tìm một rút gọn đối tượng của bảng quyết định nhất quán

Đầu vào: $DS = (U, C \cup \{d\}, V, f)$
Đầu ra : $DS' = (U', C \cup \{d\}, V, f)$

1 Bước 1: Tính $E_r = \{A_1, ..., A_t\}$;
2 Bước 2: Tính $M_d^U = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.;
3 Bước 3: Đặt $T(0) = U = \{u_1, ..., u_m\}$;
4 Bước 4: Đặt

$$T(i+1) = \begin{cases} T(i) - u_{i+1}, & \text{nếu } M_d^{T(i)-u_{i+1}} = M_d^U \\ T(i), & \text{ngược lại} \end{cases}$$

Đặt $U' = T(m)$.

Thuật toán 5 tìm một *rút gọn đối tượng* của bảng quyết định nhất quán. Rút gọn đối tượng là rút gọn các hàng của bảng quyết định nhất quán nếu đối tượng hàng không góp phần tham gia vào quá trình rút gọn thuộc tính của bảng quyết định nhất quán. Kết quả này làm giảm đáng kể các hàng dư thừa, chỉ giữ lại các hàng có ý nghĩa trong việc tìm tất cả các rút gọn thuộc tính. Thuật toán 5 có độ phức tạp thời gian đa thức.

Thuật toán 6: Tìm một rút gọn thuộc tính của bảng quyết định nhất quán

Đầu vào: $DS = (U, C \cup \{d\}, V, f)$
Đầu ra : $D \in RED(C)$

1 Bước 1: Tính $E_r = \{A_1, ..., A_t\}$;
2 Bước 2: Tính $M_d = \{A \in E_r : d \notin A, \nexists B \in E_r : d \notin B, A \subset B\}$.;
3 Bước 3: Đặt $H(0) = C = \{c_1, ..., c_n\}$;
4 Bước 4: Đặt

$$H(i+1) = \begin{cases} H(i) - c_{i+1}, & \text{nếu } \nexists B \in M_d : H(i) - c_{i+1} \subseteq B \\ H(i), & \text{ngược lại} \end{cases}$$

Đặt $D = H(n)$.

Thuật toán 6 tìm một rút gọn thuộc tính của bảng quyết định nhất quán. Hầu hết các phương pháp rút gọn thuộc tính đều tìm tất cả các rút gọn do đó có độ phức tạp thời gian thuộc lớp NP-Hard. Thuật toán 6 chỉ tìm một rút gọn thuộc tính và có độ phức tạp thời gian đa thức.

Thuật toán 7: Xây dựng cây quyết định

Đầu vào: $DS = (U, C \cup \{d\}, V, f)$
Đầu ra : Decision tree DT

1 Bước 1: Tính $CORE(C)$;
2 Bước 2: Xây dựng bảng không dư thừa thuộc tính bằng thuật toán 1;
3 Bước 3: Xây dựng bảng không dư thừa đối tượng bằng thuật toán 5;
4 Bước 4: Xây dựng bảng thu gọn $NS = (U', C' \cup \{d.\}, V, f)$ bằng thuật toán 6;
5 Bước 5: Với mỗi thuộc tính trong tập lõi $CORE(C)$, chọn thuộc tính có nhiều giá trị nhất đưa vào cây quyết định cho đến khi hết tất cả các thuộc tính trong tập lõi $CORE(C)$ thì tiếp tục chọn các thuộc tính trong tập $H = C' - CORE(C)$ với C' là tập rút gọn tìm được bằng thuật toán 6.

Thuật toán 7 là thuật toán xây dựng cây quyết định từ bảng quyết định nhất quán có độ phức tạp thời gian đa thức.

Ví dụ

Cho bảng quyết định nhất quán $DS = (U, C \cup \{d\}, V, f)$ với $U = \{u_1, ..., u_{14}\}$ ($\{1, ..., 14\}$), d là thuộc tính quyết định "Play Golf"

$C = \{Outlook, Grass, Temperature, Humidity, Windy, NumberHoles\}$ ($\{o, g, t, h, w, n\}$ hay $\{ogthwn\}$),

$R = C \cup \{d\} = \{ogthwnd\}$.

$V_{Outlook} = \{Sunny, OverCast, Rain\}$,

$V_{Temperature} = \{High, Middle, Low\}$,

$V_{Humidity} = \{High, Middle\}$,

$V_{Grass} = \{Wet, Dry\}$,

$V_{Windy} = \{Weak, Strong\}$,

$V_{NumberHoles} = \{20,10\}$,

$V_d = \{No, Yes\}$, $V = V_{Outlook} \cup V_{Grass} \cup V_{Temperature} \cup V_{Humidity} \cup V_{Windy} \cup V_{NumberHoles} \cup V_d$,

và hàm $f : U \times C \cup \{d\} \to \bigcup_{a \in C} V_a$

Bảng 1. Bảng quyết định nhất quán gốc

STT	Outlook	Grass	Temperature	Humidity	Windy	NumberHoles	d
1	Sunny	Wet	High	High	Weak	10	No
2	Sunny	Dry	High	High	Strong	20	No
3	Overcast	Wet	High	High	Weak	10	Yes
4	Rain	Dry	Middle	High	Weak	10	Yes
5	Rain	Wet	Low	Middle	Weak	20	Yes
6	Rain	Wet	Low	Middle	Strong	20	No
7	Overcast	Dry	Middle	Middle	Strong	20	Yes
8	Sunny	Wet	Low	High	Weak	10	No
9	Sunny	Wet	Middle	Middle	Weak	10	Yes
10	Rain	Dry	Middle	Middle	Weak	20	Yes
11	Sunny	Dry	Middle	Middle	Strong	20	Yes
12	Overcast	Dry	Middle	High	Strong	10	Yes
13	Overcast	Dry	High	Middle	Weak	20	Yes
14	Rain	Dry	Middle	High	Strong	10	No

Áp dụng thuật toán 1 ta được bảng 2.

Bảng 2. Bảng quyết định nhất quán không dư thừa thuộc tính

STT	Outlook	Temperature	Humidity	Windy	d
1	Sunny	High	High	Weak	No
2	Sunny	High	High	Strong	No
3	Overcast	High	High	Weak	Yes
4	Rain	Middle	High	Weak	Yes
5	Rain	Low	Middle	Weak	Yes
6	Rain	Low	Middle	Strong	No
7	Overcast	Middle	Middle	Strong	Yes
8	Sunny	Low	High	Weak	No
9	Sunny	Middle	Middle	Weak	Yes
10	Rain	Middle	Middle	Weak	Yes
11	Sunny	Middle	Middle	Strong	Yes
12	Overcast	Middle	High	Strong	Yes
13	Overcast	High	Middle	Weak	Yes
14	Rain	Middle	High	Strong	No

Áp dụng thuật toán 5 ta được bảng 3.

Bảng 3. Bảng quyết định nhất quán không dư thừa đối tượng

STT	Outlook	Temperature	Humidity	Windy	d
5	Rain	Low	Middle	Weak	Yes
6	Rain	Low	Middle	Strong	No
8	Sunny	Low	High	Weak	No
9	Sunny	Middle	Middle	Weak	Yes
12	Overcast	Middle	High	Strong	Yes
14	Rain	Middle	High	Strong	No

Áp dụng thuật toán 6 ta được bảng 4.

Bảng 4. Bảng quyết định nhất quán thu gọn

STT	Outlook	Humidity	Windy	d
1	Rain	Middle	Weak	Yes
2	Rain	Middle	Strong	No
3	Sunny	High	Weak	No
4	Sunny	Middle	Weak	Yes
5	Overcast	High	Strong	Yes
6	Rain	High	Strong	No

Tìm $CORE(C)$ theo định nghĩa.

$$\begin{bmatrix} 0 & & & & & \\ w & 0 & & & & \\ oh & 0 & 0 & & & \\ 0 & ow & h & 0 & & \\ 0 & oh & ow & 0 & 0 & \\ hw & 0 & 0 & ohw & o & 0 \end{bmatrix}$$

Từ ma trận phân biệt trên ta nhận thấy chỉ có thuộc tính đơn là $\{w\}$ và $\{o\}$ do vậy $CORE(C) = \{ow\}$.

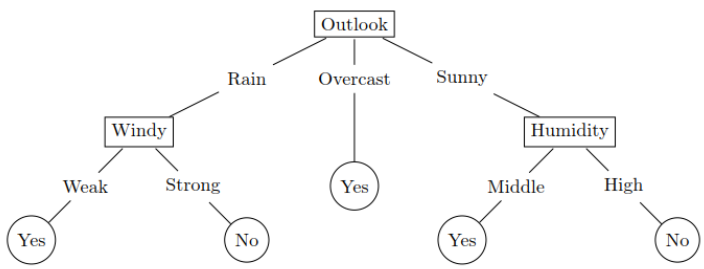
Thuộc tính $\{o\}$ có nhiều giá trị hơn so với thuộc tính $\{w\}$ nên ta lấy thuộc tính này làm gốc của cây quyết định DT.

Trong bảng rút gọn 'Bảng 4', thuộc tính "Outlook" chỉ có 1 dòng chứa giá trị "Overcast" nên nhánh "Overcast" sẽ đưa ra luôn lá quyết định là "Yes".

Giá trị "Sunny" của thuộc tính 'Outlook' chứa hai giá trị quyết định, ta kết hợp với thuộc tính 'Windy' thấy cả hai giá trị của thuộc tính này cùng là "Weak" nên ta sẽ bỏ qua thuộc tính 'Windy' và kết hợp tiếp với thuộc tính 'Humidity'. Thuộc tính 'Humidity' có hai giá trị khác nhau do vậy sẽ sinh ra một nút mới là nút 'Humidity'. Do thuộc tính 'Humidity' có hai giá trị khác nhau ứng với hai giá trị của thuộc tính quyết định nên từ nút 'Humidity' ứng với hai giá trị khác nhau sẽ sinh ra hai nhánh tương ứng với hai quyết định ở nút lá.

Giá trị "Rain" của thuộc tính nút gốc 'Outlook' chứa hai giá trị quyết định, ta kết hợp với thuộc tính 'Windy' do thuộc tính 'Humidity' đã được kết hợp ở bước trước nên bước này ta loại 'Humidity' chỉ còn lại 'Windy'. Tương ứng với hai giá trị của thuộc tính 'Windy' là hai giá trị quyết định do vậy ta đưa 'Windy' làm một nút và sinh ra hai nhánh tương ứng với hai quyết định.

Kết quả cuối cùng ta đã xây dựng được cây quyết định dựa vào các thuật toán từ 1 đến 7.



Hình 1. Cây quyết định sinh ra từ tập lõi {ow} và một rút gọn {ohw}

IV. KẾT LUẬN

Trong bài báo này chúng tôi đã nghiên cứu, đề xuất một thuật toán xây dựng cây quyết định mà sử dụng rất ít dung lượng lưu trữ bộ nhớ cũng như thực hiện nhanh chóng do các thuật toán thực hiện đều thuộc lớp độ phức tạp tính toán thời gian đa thức. Dựa trên thuật toán của chúng tôi có thể xây dựng cây quyết định từ các bảng có kích thước rất lớn mà vẫn đạt được hiệu quả cao. Thuật toán của chúng tôi cũng cho ra kết quả cây quyết định xấp xỉ tối ưu tương đương như các thuật toán ID3 hoặc C4.5 đang được sử dụng rất thịnh hành hiện nay.

LỜI CẢM ƠN

Bài báo này nhận được sự tài trợ của Đề tài nghiên cứu QG.15.41 - Đại học Quốc gia Hà Nội. Các tác giả xin cảm ơn Đại học Quốc gia Hà Nội.

TÀI LIỆU THAM KHẢO

[1] J. Demetrovics and V. D. Thi. Keys, antikeys and prime attributes. In Annales Univ. Sci. Budapest, Sect. Comp, volume 8, pages 35-52, 1987.

[2] J. Demetrovics and V. D. Thi. Algorithms for generating an armstrong relation and inferring functional dependencies in the relational datamodel. Computers & Mathematics with Applications, 26(4): 43-55, 1993.

[3] Z. Pawlak. Rough sets. International Journal of Computer & Information Sciences, 11(5):341-356, 1982.

[4] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, and W. Ziarko. Rough sets. Communications of the ACM, 38(11): 88-95, 1995.

[5] A. Skowron and C. Rauszer. The discernibility matrices and functions in information systems. In Intelligent Decision Support, pages 331-362. Springer, 1992.

[6] V. D. Thi. The minimal keys and antikeys. Acta Cybernetica, 7(4):361-371, 1986.

[7] V. D. Thi and N. L. Giang. A method to construct decision table from relation scheme. Cybernetics and Information Technologies, 11(3):32-41, 2011.

CONSTRUCT A DECISION TREE FROM A CONSISTENT DECISION TABLE

Hoang Minh Quang, Vu Duc Thi, Vu Thi Lan Anh

ABSTRACTS: *Data classification is an important method of data mining. There are many methods of data classification which decision trees is one of popular and effective methods of. The algorithms such as ID3 or C4.5 is used in many applications to classify data. In this paper we propose a method that is not based on the Entropy information function as ID3 or C4.5 and have the same effective or better than in implementation to generate a decision tree from a consistent decision table.*