J. Sci. & Devel., Vol. 11, No. 5: 729-734

Tạp chí Khoa học và Phát triển 2013, tập 11, số 5: 729-734
www.hua.edu.vn

MỘT THUẬT TOÁN TÌM TẬP RÚT GỌN THUỘC TÍNH SỬ DỤNG MA TRẬN PHÂN BIỆT ĐƯỢC

Ngọc Minh Châu, Nguyễn Xuân Thảo

Khoa Công nghệ Thông tin, Trường Đại học Nông nghiệp Hà Nội

Email: nmchau@hua.edu.vn/nxthao@hua.edu.vn

Ngày gửi lại bài: 25.04.2013 Ngày chấp nhận 20.09.2013

TÓM TẮT

Hệ thông tin giúp ích cho chúng ta lưu trữ và xử lý thông tin. Tuy nhiên, vì lý do nào đó có thể do cập nhật, thông tin lưu trữ có các thuộc tính dư thừa gây khó khăn cho việc khai phá tri thức. Do đó việc rút gọn thuộc tính là yêu cầu cần thiết trong khai phá tri thức. Có nhiều kiểu rút gọn tri thức và luật quyết định đã được đề xuất trong khai phá dữ liệu. Trong bài báo này chúng tôi đưa ra một thuật toán tìm tập rút gọn trên một bảng quyết định dựa trên ma trận phân biệt được.

Từ khóa: Hệ thông tin, ma trận phân biệt được, tập rút gọn.

An Algorithm to Find the Attibute Reduction by Using Discernibility Matrix

ABSTRACT

The information systems help users backup and process information. However, some reasons of apdating backup information having redundant attributes make it difficult for exploring knowledge. Thus, the attribute reductions are essential requirements for mining knowledge. There exist several types of attribute reduction and decision rules that have been proposed in data mining. The present paper described a heuristic algorithm to find the reduction on decision table based on the discernibility matrix.

Keywords: Information systems, discernibility matrix, attribute reduction.

1. ĐẶT VẤN ĐỀ

Lý thuyết tập thô được Pawlak đề xuất vào đầu những năm 1980 được xem như một cách tiếp cận mới để phát hiện tri thức và nó đã tạo tạo ra một cơ sở vững chắc cho việc phát triển các ứng dụng của lĩnh vực khai phá dữ liệu (Pawlak, 1982, 1991, 1998). Sự ra đời của lý thuyết tập thô đã thu hút sự quan tâm của nhiều nhà nghiên cứu lý thuyết và ứng dụng. Lý thuyết này được xem là cơ sở quan trọng trong lĩnh vực trí tuệ nhân tạo và các khoa học về nhận thức như học máy, hệ hỗ trợ quyết định, hệ chuyên gia, đặc biệt là lĩnh vực khai phá dữ liệu (Pawlak, 1982; 1991; 1998 và Klemettinen et al., 1994). Dữ liệu trong thực tế thường không đầy đủ, dư thừa hoặc không chính xác và đa dạng, gây ảnh hưởng không tốt trong quá trình phát hiện tri thức từ dữ liệu. Mỗi phần tử dữ liệu thu nhận từ thực tế (mỗi bộ hay bản ghi) thường chứa rất nhiều thuộc tính, trong đó có nhiều thuộc tính không cần thiết trong việc phân lớp các đối tượng. Khái niệm tập rút gọn (hay tập thuộc tính rút gọn) (Pawlak, 1982; 1991; 1998) là một khái niệm cơ bản, hỗ trợ việc phân tích dữ liệu dựa trên cơ sở lý thuyết tập thô. Một tập rút gọn là một tập thuộc tính tối tiểu, duy trì được thông tin hay tri thức của dữ liệu như đối với tập thuộc tính ban đầu. Việc tìm tập rút gọn cho đến nay vẫn được nhiều nhà nghiên cứu quan tâm như Hoa N.S. and N. H. Son (1996), Qian (2009), Xu and Yang (2009), Zhang and Gao (2012).

Có hai quan niệm rút gọn thuộc tính, quan niệm thứ nhất là rút gọn thuộc tính dựa trên ma trận phân biệt được (xem Hoa N.S. and N.H.Son (1996), Qian (2009), Xu and Yang

(2009), Zhang and Gao (2012)), quan niệm kia là rút gọn thuộc tính dựa trên vùng dương (Hoa N.S. and N.H.Son, 1996,...). May mắn là hai quan niệm đã được chứng minh là tương đương (Xu and Yang, 2009). Hơn nữa, việc tìm rút gọn theo quan niệm thứ nhất còn được chỉ ra là độ phức tạp tính toán trong nhiều trường hợp còn thấp hơn theo quan niệm còn lại. Cùng với việc xây dựng các ma trận khả phân để thuận lợi cho việc tìm rút gọn thì việc tìm, cải tiến thuật toán tìm rút gọn thuộc tính dựa trên các ma trận khả phân đã biết cũng được quan tâm. Trong bài báo này, chúng tôi đề xuất một thuật toán tìm tập rút gọn dựa trên ma trận phân biệt được đưa ra bởi Pawlak (1998).

Các phần tiếp theo của bài báo, phần 2 được giới thiệu lại các khái niệm cơ bản về hệ thông tin và rút gọn tri thức, phần 3 đưa ra thuật toán tìm tập rút gọn dựa trên ma trận phân biệt được.

2. CÁC KHÁI NIỆM

2.1. Hệ thông tin

Hệ thông tin là một cặp (U,A), với $U=\{u_1,u_2,...,u_n\}$ là tập hữu hạn khác rỗng (được gọi là tập vũ trụ các đối tượng) và A là tập hữu hạn khác rỗng các thuộc tính. Với mọi $a \in A$ ta kí hiệu V_a là tập giá trị thuộc tính a. Mặt khác, nếu $u \in U$ và $a \in A$, ta sẽ kí hiệu $a(u) \in V_a$ là giá trị thuộc tính a của đối tượng u.

2.2. Quan hệ tương đương

Mối quan hệ nhị phân $R \subseteq X \times X$ được gọi là quan hệ tương đương nếu thỏa mãn:

+ Tính phản xạ: xRx

+ Tính đối xứng: nếu xRy thì yRx.

+ Tính bắc cầu: nếu xRy và yRx thì xRz.

Lớp tương đương theo quan hệ tương đương R của phần tử $x \in X$ là các phần tử $y \in X$ sao cho xRy.

2.3. Quan hệ không phân biệt được

Cho hệ thông tin $\Sigma = (U, A)$, với $B \subseteq A$. Quan hệ không phân biệt được đối với B hay $B-$ không phân biệt được, ký hiệu là IND(B), được định nghĩa như sau:

$$IND(B) = \{(x, x') \in U^2 : \forall a \in B, a(x) = a(x')\}$$

Đây là một quan hệ tương đương. Lớp tương đương của quan hệ *B* - không phân biệt được ký hiệu là $[x]_B$. Như vậy, ta có $IND(B)$ là một quan hệ tương đương trên *U*.

2.4. Tập rút gọn và tập lõi

Dựa trên quan hệ bất khả phân có thể giảm kích thước dữ liệu bằng cách trích ra các lớp tương đương từ không gian tương đương. Từ đó chỉ cần 1 phần tử thuộc lớp tương đương đã thể hiện được cho toàn lớp tương đương (theo 1 số thuộc tính nào đó, chưa chắc đã hữu hiệu cho việc phân biệt các đối tượng trên toàn bộ cơ sở dữ liệu).

Một cách tiếp cận khác để giảm kích thước dữ liệu: chỉ giữ lại các thuộc tính cần thiết, loại đi những thuộc tính không làm ảnh hưởng đến quan hệ bất khả phân. Những tập thuộc tính như vậy gọi là tập thuộc tính thu gọn.

Cho hệ thông tin $\Sigma=(U,A), B\subseteq A$ *và* $a\in B$. Ta nhắc lại các định nghĩa về tập rút gọn và lõi (Pawlak, 1998).

Định nghĩa 1:

i) Thuộc tính a là không quan trọng trong B nếu $\text{IND(B)} = \text{IND}(\text{B} \mid \{\text{a}\})$, ngược lại a được gọi là quan trọng trong B.

ii) Tập thuộc tính B là độc lập nếu tất cả các thuộc tính của nó là quan trọng.

iii) Tập con B' của B là một tập rút gọn (ký hiệu $\text{B}' = \text{R(B)}$) của B nếu B' là độc lập và $\text{IND}(\text{B}') = \text{IND(B)}$.

Định nghĩa 2: Cho B là tập con của A, lõi của tập B ký hiệu là: Core(B), là giao của tất cả các tập rút gọn của B.

$$Core(B) = \bigcap_{R(B)\in \mathrm{Re}\,d(B)} R(B)$$

Trong đó, $\mathrm{Re}\,d(B)$ là tập tất cả các rút gọn của B.

Việc tìm tập thuộc tính rút gọn có thể dựa vào ma trận phân biệt được.

2.5. Ma trận phân biệt được

Định nghĩa 3: Xét hệ thông tin $\Sigma = (U, A)$ giả sử $B \subseteq A$. Ta định nghĩa ma trận B - phân biệt được, kí hiệu *M(B),* là ma trận đối xứng cấp $n \times n$ mà mỗi phần tử của nó được xác định như sau:

$$c_{ij} = \{a \in B : a(x_i) \neq a(x_j)\}, \quad i, j = \overline{1, n}.$$

Định lí về việc xác định lõi của tập các thuộc tính B được Pawlak (1998) nêu ra, nhưng không chứng minh. Ở đây, chúng tôi nhắc lại kết quả này và chứng minh chi tiết trong định lí sau:

Định lý 1: $Core(B) = \{b \in B : \exists c_{ij} = \{b\}\}$

Chứng minh:

Nếu $b \in \{b \in B : \exists c_{ij} = \{b\}\}$ thì các lớp tương đương $[x_i]$, $[x_j]$ chỉ được phân biệt với nhau bởi thuộc tính {b}, suy ra $b \in Core(B)$.

Do đó $\{b \in B : \exists c_{ij} = \{b\}\} \subset Core(B)$. (1)

Ngược lại, nếu $b \in Core(B)$ thì $b \in \bigcap_{R(B) \in Red(B)} R(B)$. Vì vậy $\{b\} \subseteq \bigcap_{R(B) \in Red(B)} R(B)$

$\Rightarrow \{b\} \subseteq R(B)$, từ đó $[x]_{R(B)} \subset [x]_{\{b\}}$.

Vì $b \in Core(B)$ nên $\exists y, z \in U : [y]_{\{b\}} \neq [z]_{\{b\}}$. Do đó $\exists c_{12} \in M(B) : b \in c_{12}$.

Hơn nữa $c_{12} = \{b\}$ vì nếu giả sử ngược lại $\exists a \in B : a \neq b$ sao cho $[y]_{\{a\}} \neq [z]_{\{a\}}$. Khi đó phải tồn tại $t \notin [y]_{\{b\}} \cup [z]_{\{b\}} : \left[\begin{array}{l} [t]_{\{a\}} = [y]_{\{a\}} \\ [t]_{\{a\}} = [z]_{\{a\}} \end{array}\right.$, thậy vậy, nếu không tồn tại t như vậy thì $\forall x \notin [y]_{\{b\}} \cup [z]_{\{b\}}$ ta đều có $\left[\begin{array}{l} [x]_{\{a\}} \neq [y]_{\{a\}} \\ [x]_{\{a\}} \neq [z]_{\{a\}} \end{array}\right.$ tức là thuộc tính *b* không quan trọng trong B và có thay thế được bởi thuộc tính *a* ở trên, điều này mâu thuẫn với $b \in Core(B)$.

Do đó, ta phải có $c_{12} = \{b\}$, tức là $Core(B) \subset \{b \in B : \exists c_{ij} = \{b\}\}$. (2)

Từ (1) và (2) suy ra :

$$Core(B) = \{b \in B : \exists c_{ij} = \{b\}\}.$$

Chú ý: Nếu B không có phần tử quan trọng nào thì Core(B) = φ.

3. THUẬT TOÁN

3.1. Thuật toán tìm lõi *Core(B)*

Input: Hệ thông tin

$\Sigma = (U, A)(*a_1, a_2, ..., a_m*), B \subseteq A$.

Output: M(B) and Core(B)

1. $Core(B) = \varphi; c_{ij} := \varphi; Index := \emptyset;$
   $t_{ij} = 0; sign(i, j) = -1;\ i, j = 1, ..., n$

2. For i:= 2 to n do

 For j:=1 to i - 1 do

  For k:= 1 to m do

 If $a_k(x_j) \neq a_k(x_i)$ then $c_{ij} := c_{ij} \cup \{a_k\}$, $t_{ij} := t_{ij} + 1$.

  else $c_{ij} := c_{ij}$; $t_{ij} = t_{ij}$.

  If $t_{ij} = 1$ then Core(B) := Core(B) $\cup$ c_{ij} and Index := Index $\cup$\{k\}; Sign(i, j):=1;

Chú ý:

Nếu $c_{ij} = \phi$ trong M(B) thì $[x_i]_B = [x_j]_B$.

Hơn nữa chúng ta xét $B \subset A$ và đặt $D(B) = \sum(|[x_i]_B| - |[x_i]_A|)$. Ta có kết quả sau:

Định lý 2: Nếu B là độc lập trong A và D(B) = 0 thì B = R(A).

Chứng minh: Giả sử rằng $B \subset A$, khi đó $\forall x \in U$ ta có $[x]_A \subset [x]_B$ và $|[x]_A| \geq |[x]_B|$.

Nếu $D(B) = \sum_{i=1}^{n}(|[x]_B| - |[x]_A|) = 0$ thì $|[x]_B| = |[x]_A|$.

Do đó $[x]_B = [x]_A, \forall x \in U$ hay $IND(B) = IND(A)$. Hơn nữa, vì B là độc lập trong A nên B là rút gọn của A, tức là $B = R(A)$.

3.2. Thuật toán tìm một rút gọn R(A) của A

Input: Hệ thông tin $\Sigma = (U, A)(*a_1, a_2, ..., a_m*)$,

Output: R(A).

Sử dụng thuật toán trong mục 3.1 tìm được *Core(A).*

1. Đặt $B := Core(A); T := \phi$ (* If $a_k \in B$ then $k \in Index$ *)

Nếu $B \neq \phi$ thì chuyển sang bước 2, nếu không thì chuyển sang bước 3.

2. Nếu $D(B) = 0$ thì chuyển sang bước 4, nếu không sang bước 3.

3. For k:= 1 to m do

If $k \notin Index$ then

For i:= 2 to n do

For j:=1 to i - 1 do

If Sign (i,j) = -1 then

If $a_k \in c_{ij}$ and $c_{ij} \cap B = \varnothing$ then $B := B \cup \{a_k\}$ and Sign (i,j) = 1; và quay lại bước 2.

4. *R(A):= B*

Trong thuật toán này chúng ta có thể dùng $B \cap c_{ij} \neq \phi, \forall c_{ij} \neq \phi$.

Điều đó có nghĩa là có thể bỏ qua điều kiện $D(B) = 0$ và cũng thu được tập rút gọn *R(A)* của *A*.

Có thể thấy rằng trong thuật toán này, ở bước 3 ta thêm các thuộc tính vào tập B, thuộc tính được thêm vào bảo đảm tính độc lập của B, trong bước 2 ta kiểm tra được *B = R(A)*. Do đó, có khẳng định sau như là sự bảo đảm sự đúng đắn của thuật toán.

Định lý 3: Tập kết quả *R(A)* trong thuật toán trên là một trong những tập rút gọn của A. Hơn nữa, nếu $Core(B) \cap c_{ij} \neq \phi, \forall c_{ij} \neq \phi$ thì *Core(A)* là rút gọn duy nhất của A.

Xét một số ví dụ minh họa:

Ví dụ 1: Cho hệ thông tin

U	a	b	c	d	e	f
x1	0	0	1	0	1	0
x2	1	1	1	1	1	0
x3	1	0	1	1	0	0
x4	0	0	1	0	0	1
x5	0	0	1	0	1	1
x6	0	0	1	0	1	1
x7	1	0	1	0	1	1
x8	1	0	1	0	1	1

* Ma trận khả phân

	x1	x2	x3	x4	x5	x6	x7	x8
x1	ϕ							
x2	{a,b,d}	ϕ						
x3	{a,d,e}	{b,e}	ϕ					
x4	{e}	{a,b,d,e,f}	{a,d,f}	ϕ				
x5	{f}	{a,b,d,f}	{a,d,e,f}	{e}	ϕ			
x6	{f}	{a,b,d,f}	{a,d,e,f}	{e}	ϕ	ϕ		
x7	{a,f}	{b,d,f}	{d,e,f}	{a,e}	{a}	{a}	ϕ	
x8	{a,f}	{b,d,f}	{d,e,f}	{a,e}	{a}	{a}	ϕ	ϕ

$$U / IND(A) = \{[x1],[x2],[x3],[x4],[x5,x6],[x7,x8]\}$$

Dễ dàng nhận thấy $Core(A) = \{a, e, f\}$ và $Core(A) \cap c_{ij} \neq \phi, \forall c_{ij} \neq \phi$, vì thế $Core(A)$ là rút gọn duy nhất của A.

Ví dụ 2: Cho hệ thông tin

U	a1	a2	a3	a4	a5
x1	0	1	1	1	0
x2	0	2	1	1	0
x3	1	3	0	1	1
x4	1	2	1	1	1
x5	0	3	1	0	1

* Ma trận khả phân

	x1	x2	x3	x4	x5
x1	ϕ				
x2	{a2}	ϕ			
x3	{a1,a2,a3,a5}	{a1,a2,a3,a5}	ϕ		
x4	{a1,a2,a5}	{a1,a5}	{a2,a3}	ϕ	
x5	{a2,a4}	{a1,a5}	{a1,a3,a4}	{a1,a2,a4}	ϕ

Trong ví dụ này chúng ta có:

$U / IND(A) = \{[x1],[x2],[x3],[x4],[x5]\}$

Sign(i,j):= -1, $\forall i, j = 1,...,5$.

$Core(A) = \{a2\} = c_{21}$; *Index:={2}; Sign(2,1):=1.*

Ta tìm các rút gọn theo các bước sau:

Bước 0: $B = Core(A) = \{a2\}$; Index:={2};

Bước 1: $D(B) = 2 > 0$. Chuyển sang bước 2.

Bước 2:

$1 \notin \text{Index}; \text{sign}(4,2) = -1; \text{a}1 \in c_{42},$

$c_{42} = \{\text{a}1, \text{a}5\}, c_{42} \cap B = \varnothing, \to B := \{\text{a}1, \text{a}2\}; .\text{Sign}(4,2) := 1.$

Đến đây ta có: $D(B) = 0$, và $R_1(A) = \{a1, a2\}$.

Ví dụ 3. Xét một bảng thông tin về việc theo dõi các bệnh nhân bị bệnh cúm (Flu) (Pawlak, 1998).

Patient	Headache (a1)	Muscle _pain(a2)	Temperature(a3)	Flu
u1	No	Yes	High	Yes
u2	Yes	No	High	Yes
u3	Yes	Yes	Very high	Yes
u4	No	Yes	Normal	No
u5	Yes	No	High	No
u6	No	Yes	Very high	Yes

Ở đây ta quan tâm đến các thuộc tính điều kiện A= {Headache, Muscle-pain, Temprature} và ta tìm tập rút gọn trên tập các thuộc tính này.

Ta có ma trận phân biệt được:

	u1	u2	u3	u4	u5	u6
u1	ϕ					
u2	{a1,a2}	ϕ				
u3	{a1,a3}	{a2,a3}	ϕ			
u4	{a3}	{a1,a2,a3}	{a1,a3}	ϕ		
u5	{a1,a2}	ϕ	{a2,a3}	{a1,a2,a3}	ϕ	
u6	{a3}	{a1,a2,a3}	{a1,a2}	{a3}	{a1,a2,a3}	ϕ

Trong ví dụ này chúng ta có:

$U / IND(A) = \{[u1], [u2, u5], [u3], [u4], [u6]\}$

Sign(i,j):= -1, $\forall i, j = 1,\ldots,6.$.

$Core(A) = \{a3\} = c_{41} = c_{61} = c_{64}$;

Index :={3},

Sign(4,1) = Sign(6,1) = Sign (6,4) = 1.

$c_{21} = \{a1, a2\} \Rightarrow c_{21} \cap Core(A) = \phi$

và $c_{51} = \{a1, a2\} \Rightarrow c_{51} \cap Core(A) = \phi$

$c_{63} = \{a1, a2\} \Rightarrow c_{63} \cap Core(A) = \phi$

Ta tìm các rút gọn theo các bước sau:

Bước 0: $B = Core(A) = \{a3\}; Index = \{3\}$.

Bước 1: $D(B) = 6 > 0$ thì chuyển sang bước 2.

Bước 2:

$1 \notin \text{Index}, \text{a}1 \in c_{21}, c_{21} \cap B = \varnothing,$

$\to B := \{\text{a}1, \text{a}3\}; \text{Sign}(2,1) := 1.$

Đến đây ta có: $D(B) = 0$, và $R_1(A) = \{a1, a3\}$.

Chú ý: Nếu bước 2 ta chọn

$2 \notin \text{Index}, \text{a}2 \in c_{21}, c_{21} \cap B = \varnothing,$

$\to B := \{\text{a}2, \text{a}3\}; \text{Sign}(2,1) := 1.$

Ta cũng có $D(B) = 0$, và $R_2(A) = \{a2, a3\}$. Trong trường hợp này có hai tập rút gọn $R_1(A) = \{a1, a3\}$ và $R_2(A) = \{a2, a3\}$.

4. KẾT LUẬN

Thuật toán tìm lõi và tìm tập rút gọn của các thuộc tính của hệ thông tin dựa trên ma trận phân biệt được đã được xây dựng. Việc này cho phép chúng ta loại bỏ các thuộc tính dư thừa trong cơ sở dữ liệu và hữu ích cho việc khai phá dữ liệu. Độ phức tạp của thuật toán là $O(|U|^2 \times |A|)$.

TÀI LIỆU THAM KHẢO

Hoa N. S. and N. H. Son (1996). Some efficient algorithms for rough set methods. Proceedings of International Conference on Information processing Management of Uncertainty in Knowledge – based Systems, pp. 1451 – 1456.

Klemettinen M., H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo (1994). Finding interesting rules from large sets of discovered association rules, Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, p. 401– 407.

Pawlak Z. (1982). Rough sets, International Journal of computer and information sciences 11(5): 341-356.

Pawlak Z. (1991). Rough sets- Theoretical aspects of reasoning about data, Kluwer academic publishers, Dordrecht – Netherlands.

Pawlak Z. (1998). Rough sets elements, Institute of theoretical and applied informattics, Polish Academy of Sciences.

Qian J. (2009). Applying Indiscernibility Attribute to Attribute Reduction Based on Discernibility

Matrix, Environmental Science and Information Application Technology. ESIAT 2009 International Conference on, 2: 397 - 400.

Xu Z. and B. Yang (2009). An Efficient Algorithm for Pawlak Reduction Based on Simplified Discernibility Matrix, Fuzzy Information and Engineering: Advances in Soft Computing, 54: 610-619.

Zhang L. and S. Gao (2012). Attribute reduction with discernibility matrix approaches, 24th Chinese control and decision conference, p. 2070 - 2072.