Tạp chí Tin học và Điều khiển học, T.23, S.3 (2007), 284–295

PHÁT HIỆN CÁC PHỤ THUỘC HÀM XẤP XỈ THEO CÁCH TIẾP CẬN TẬP THÔ

TRẦN DUY ANH

Trường Cao Đẳng Sư PhạmThừa Thiên Huế

Abstract. Functional dependencies (FDs) play very an important role in the design of relational databases. Recently, researchers [1, 4, 5] presented a generalization of functional dependencies based on rough set theory, called approximate functional dependencies (AFDs).

In this article, firstly, We research on approximate functional dependency based on rough set theory. After that base on FD-Mine [9] which is an algorithm for finding functional dependencies, We construct an algorithm for mining approximate functional dependencies from databases, called AFD-Mine.

Tóm tắt. Các phụ thuộc hàm đóng một vai trò rất quan trọng trong thiết kế các hệ cơ sở dữ liệu quan hệ. Gần đây, các nhà nghiên cứu $[1, 4, 5]$ đã đưa ra một sự mở rộng của những phụ thuộc hàm dựa trên lý thuyết tập thô, được gọi là các phụ thuộc hàm xấp xỉ.

Trong bài báo này, đầu tiên, chúng tôi nghiên cứu phụ thuộc hàm xấp xỉ trên cơ sở lý thuyết tập thô. Sau đó, xây dựng thuật toán phát hiện các phụ thuộc hàm xấp xỉ trong cơ sở dữ liệu, gọi là AFD-Mine, thuật toán này dựa trên FD-Mine [9], một thuật toán phát hiện các phụ thuộc hàm.

1. MỞ ĐẦU

Khái niệm phụ thuộc hàm được đưa ra bởi Codd, nó đã đóng một vai trò rất quan trọng trong lý thuyết cơ sở dữ liệu (CSDL) quan hệ. Các phụ thuộc hàm rất hữu ích trong việc phân tích và thiết kế cơ sở dữ liệu quan hệ như xác định khóa, xác định các dạng chuẩn, các vấn đề về nhất quán dữ liệu... Tuy nhiên, trong thực tế, do có một số giá trị dữ liệu không chính xác hoặc một số ngoại lệ nào đó làm cho các phụ thuộc hàm không thỏa. Sự phụ thuộc tuyệt đối này dường như quá nghiêm ngặt khi ta hình dung tới một quan hệ có hàng nghìn bộ, trong khi đó, chỉ có khoảng vài bộ vi phạm phụ thuộc hàm. Điều này làm mất tính chất phụ thuộc vốn có giữa các thuộc tính. Vì vậy, các nhà nghiên cứu đã mở rộng khái niệm phụ thuộc hàm thành phụ thuộc hàm xấp xỉ (Approximate Functional Dependency), các phụ thuộc này cho phép có một số lượng lỗi nhất định của các bộ dữ liệu đối với phụ thuộc hàm.

Các phụ thuộc hàm xấp xỉ không những giúp chúng ta thấy được mối quan hệ tiềm ẩn giữa các thuộc tính mà còn giúp ta thuận tiện hơn trong việc phân tích dữ liệu và đánh giá thông tin.

Gần đây, việc phát hiện các phụ thuộc hàm xấp xỉ trong cơ sở dữ liệu là một vấn đề nghiên cứu thú vị, có ý nghĩa và cũng là một trong những mục tiêu của phát hiện tri thức (Knowledge Discovery).

Trong bài báo này, đầu tiên, chúng tôi đưa ra một mối liên hệ giữa độ phụ thuộc và độ

đo lỗi của phụ thuộc hàm và xem xét độ đo lỗi trên các phân hoạch thu gọn. Sau đó, đề xuất một số tính chất và luật cắt tỉa mới để xây dựng thuật toán phát hiện các phụ thuộc hàm xấp xỉ trong cơ sở dữ liệu, gọi là AFD-Mine, thuật toán này dựa trên FD-Mine [9], một thuật toán phát hiện các phụ thuộc hàm. Cuối cùng, mối liên hệ giữa Tane (một thuật toán hiệu quả để phát hiện các phụ thuộc hàm và phụ thuộc hàm xấp xỉ) và AFD-Mine cũng như một số nhận xét có ý nghĩa về độ đo lỗi, các tính chất và các luật cắt tỉa cũng được đưa ra.

Một số khái niệm cơ bản của CSDL quan hệ, có thể tham khảo trong [2, 3, 8, 9].

2. MỘT SỐ KHÁI NIỆM CƠ BẢN CỦA LÝ THUYẾT TẬP THÔ

Định nghĩa 1. [2, 8] (Quan hệ không phân biệt được) Cho $r(R)$, khi đó, với bất kỳ $X \subseteq R$, tồn tại một quan hệ không phân biệt được $I(X)$ trên r được định nghĩa như sau:

$$\forall t, u \in r,\ (t, u) \in I(X) \Leftrightarrow t[X] = u[X].$$

Định nghĩa 2. [2, 9] (Lớp tương đương và phân hoạch) Quan hệ $I(X)$ sẽ phân hoạch r thành các lớp tương đương. Lớp tương đương của bộ $t \in r$ ứng với tập $X \subseteq R$, ký hiệu $[t]_X$, được định nghĩa như sau:

$$[t]_X = \{u \in r | t[A] = u[A],\ \forall A \in X\},\ [t]_X \neq \emptyset.$$

Khi đó, $\pi_X = \{[t]_X | t \in r\}$ là một phân hoạch của r ứng với X. Lực lượng của π_X, ký hiệu $|\pi_X|$, là số lớp tương đương của π_X.

Cho $U \in \pi_X$, khi đó, ta quan niệm rằng, U thỏa phụ thuộc hàm $X \to Y$, ký hiệu là $U \models X \to Y$ nếu với mọi $t, u \in U$ sao cho $t[X] = u[X]$, thì $t[Y] = u[Y]$.

Định nghĩa 3. [4] (Phân hoạch thu gọn) Phân hoạch thu gọn của π_X, ký hiệu $\hat{\pi}_X$, nếu $\hat{\pi}_X = \{U \in \pi_X \big| |U| > 1\}$.

Để giảm độ phức tạp tính toán khi làm việc với các phân hoạch, ta dùng các phân hoạch thu gọn thay cho các phân hoạch.

Định nghĩa 4. [1, 6] (Không gian dương) Không gian dương của tập thuộc tính X ứng với tập thuộc tính Y được định nghĩa như sau:

$$POS(X, Y) = \bigcup \{U \in \pi_X | U \subseteq V \ \text{ và } \ V \in \pi_Y\} = \bigcup \{U \in \pi_X | U \models X \to Y\}.$$

Định nghĩa 5. [4] (Làm mịn phân hoạch) Cho hai tập thuộc tính $X, Y \subseteq R$, phân hoạch π_X mịn hơn π_Y nếu với bất kỳ lớp tương đương $U \in \pi_X$, thì tồn tại một lớp tương đương $V \in \pi_Y$ sao cho $U \subseteq V$.

Bổ đề 1. [4] *$X \to A$ đúng khi và chỉ khi $|\pi_X| = |\pi_{X \cup \{A\}}|$.*

3. PHỤ THUỘC HÀM XẤP XỈ TRÊN CƠ SỞ LÝ THUYẾT TẬP THÔ

Định nghĩa 6. [1] (Độ phụ thuộc) Tập thuộc tính Y phụ thuộc vào tập thuộc tính X với mức độ $k \in [0, 1]$, ký hiệu là $X \to^k Y$, trong đó k được xác định như sau:

$$k = \frac{|POS(X, Y)|}{|r|}.$$

Độ phụ thuộc k rất thuận tiện trong việc xem xét hệ tiên đề Armstrong và một số phép toán đại số quan hệ đối với phụ thuộc hàm xấp xỉ trong [1, 6]. Tuy nhiên để xây dựng thuật

toán phát hiện các phụ thuộc hàm xấp xỉ, Kivinen và Mannila [5] đã đưa ra một độ đo để tính toán lỗi của một phụ thuộc hàm, gọi là độ đo g_3.

Định nghĩa 7. [5] (Độ đo lỗi) Cho quan hệ $r(R)$, khi đó, độ đo lỗi g_3 của một phụ thuộc hàm $X \to A$ được xác định như sau.

$$g_3(X \to A, r) = 1 - \frac{\max\{|s| \big| s \subseteq r,\ s \models X \to A\}}{|r|}.$$

Từ đó, $X \to A$ đúng trên r ứng với một ngưỡng lỗi $\varepsilon \in [0, 1]$ khi và chỉ khi $g_3(X \to A, r) \leqslant \varepsilon$.

Từ Định nghĩa 7 và Bổ đề 1, Huhtala et al [4] đã xây dựng một công thức để tính độ đo lỗi dựa trên các phân hoạch như sau: Bất kỳ U thuộc π_X là hợp của một hoặc nhiều lớp tương đương $V_1, V_2, ...$ của $\pi_{X\cup\{A\}}$, nghĩa là, với mỗi $U \in \pi_X$ và $V_1, V_2, ..., V_i \in \pi_{X\cup\{A\}}$ sao cho $V_1, V_2, ..., V_i \subset U$ thì ta có $U = \bigcup\limits_i V_i$. Khi đó, để $X \to A$ là một phụ thuộc hàm đúng, thì với mỗi $U \in \pi_X$, ta phải loại bỏ những bộ ứng với các lớp tương đương $V_1, V_2, ..., V_i \in \pi_{X\cup\{A\}}$ sao cho $V_1, V_2, ..., V_i \subset U$, ngoại trừ các bộ ứng với lớp tương đương $V = \max\limits_i(V_1, V_2, ..., V_i)$, nghĩa là, ta loại bỏ những bộ ứng với các lớp tương đương $\{V_1, V_2, ..., V_i\}\backslash V$. Khi đó, độ đo lỗi của $X \to A$ là ([4])

$$g_3(X \to A) = \frac{|r| - \sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subseteq U\}}{|r|}.$$

Sau đây là một tính chất mới được đề xuất, thể hiện mối liên hệ giữa độ phụ thuộc và độ đo lỗi của một phụ thuộc hàm.

Tính chất 1. (Liên hệ giữa g_3 và k) *Cho độ phụ thuộc* $k = \dfrac{POS(X, A)}{|r|}$ *và độ đo lỗi* $g_3(X \to A)$ *của phụ thuộc hàm* $X \to A$. *Khi đó, ta có:*

$$g_3(X \to A) = 1 - k - \frac{\sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subset U\}}{|r|}.$$

Chứng minh. Ta có

$$\begin{aligned}
g_3(X \to A) &= \frac{|r| - \sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subseteq U\}}{|r|} \\
&= \frac{|r| - \sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subset U\} - \sum\limits_{U\in\pi_X} \{|V| \big| V \in \pi_{X\cup\{A\}},\ V = U\}}{|r|} \\
&= \frac{|r| - \sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subset U\} - \big|\bigcup\{U \in \pi_X \big| U \models X \to A\}\big|}{|r|} \\
&= \frac{|r| - \sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subset U\} - |POS(X, A)|}{|r|} \\
&= 1 - \frac{|POS(X, A)|}{|r|} - \frac{\sum\limits_{U\in\pi_X} \max\{|V| \big| V \in \pi_{X\cup\{A\}},\ V \subset U\}}{|r|}.
\end{aligned}$$

■

Tính chất 1 giúp ta thuận tiện trong việc phân tích, xây dựng thuật toán phát hiện các phụ thuộc hàm xấp xỉ sau này.

Nhận xét 1. Công thức tính độ đo lỗi $g_3(X \to A)$ của phụ thuộc hàm $X \to A$ từ các phân hoạch không còn đúng đối với các phân hoạch thu gọn.

Thật vậy, cho $r(R),\ R = \{A, B, C\}$ như sau:

Bảng 1. Một quan hệ r trên tập thuộc tính $\{A, B, C\}$

Bộ	A	B	C
1	0	1	2
2	0	1	3
3	0	1	4
4	0	2	3
5	1	2	5
6	1	4	2
7	2	5	6

Ta có, $\pi_A = \{\{1,2,3,4\},\{5,6\},\{7\}\} \Rightarrow \hat{\pi}_A = \{\{1,2,3,4\},\{5,6\}\}$.

$\pi_{AB} = \{\{1,2,3\},\{4\},\{5\},\{6\},\{7\}\} \Rightarrow \hat{\pi}_{AB} = \{\{1,2,3\}\}$.

Khi đó

$$g_3(A \to B) = \frac{|r| - \sum\limits_{U \in \pi_A} \max\{|V| \big| V \in \pi_{AB},\ V \subseteq U\}}{|r|} = \frac{2}{7}.$$

Tuy nhiên

$$\frac{|r| - \sum\limits_{U \in \hat{\pi}_A} \max\{|V| \big| V \in \hat{\pi}_{AB},\ V \subseteq U\}}{|r|} = \frac{4}{7} \neq \frac{2}{7}.$$

Từ đó, ta xây dựng một công thức mới để tính độ đo lỗi của phụ thuộc hàm $X \to A$ từ các phân hoạch thu gọn như sau. Với mỗi U thuộc $\hat{\pi}_X$ nếu:

• $\exists V_1, V_2, ..., V_i : \forall j = 1, ..., i, V_j \in \hat{\pi}_{X \cup \{A\}}, V_j \subseteq U$ thì ta loại $(|U| - \max\{|V| \big| V \in \pi_{X \cup \{A\}},\ V \subset U\})$ bộ.

• $\nexists V_i : V_i \subset U$ thì ta loại $(|U| - 1)$ bộ.

Vậy, độ đo lỗi $g_3(X \to A)$ từ các phân hoạch thu gọn được tính như sau:

$g_3(X \to A)$

$$= \frac{\sum\limits_{U \in \hat{\pi}_X} (|U| - \max\{|V| \big| V \in \hat{\pi}_{X \cup \{A\}},\ V \subset U\}) + \sum\limits_{U \in \hat{\pi}_X} \{(|U| \big| \nexists V \in \hat{\pi}_{X \cup \{A\}}, V \subset U) - 1\}}{|r|}.$$

■

4. PHÁT HIỆN CÁC PHỤ THUỘC HÀM XẤP XỈ TRÊN CƠ SỞ TẬP THÔ

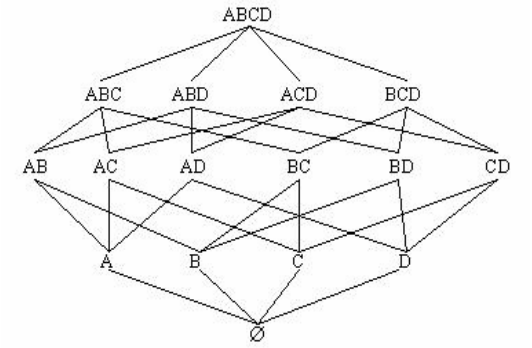
4.1. Phát biểu bài toán

Cho quan hệ $r(R), R = \{A_1, ..., A_n\}$. Vấn đề đặt ra là tìm các phụ thuộc hàm và phụ thuộc hàm xấp xỉ đầy đủ, không tầm thường trong quan hệ r ứng với một ngưỡng lỗi

ε nào đó, nghĩa là, với một ngưỡng lỗi $\varepsilon \in [0,1]$ cho trước, ta tìm các phụ thuộc dạng $X \to A,\ X \subseteq R,\ A \in R,\ A \notin X$ sao cho $g_3(X \to A) \leqslant \varepsilon$.

4.2. Thuật toán Tane

Để tìm kiếm các phụ thuộc hàm và phụ thuộc hàm xấp xỉ đầy đủ, không tầm thường, thuật toán bắt đầu tìm kiếm từ tập một thuộc tính thông qua một dàn tìm kiếm như Hình 1. Khi thuật toán xử lý tập X, thuật toán kiểm tra các phụ thuộc dạng $X\backslash\{A\} \to A$, với $A \in X$. Ở dàn tìm kiếm, một cung giữa tập X và $X \cup \{A\}$ biểu diễn một phụ thuộc hàm không tầm thường dạng $X \to A$.



Hình 1. Một dàn tìm kiếm với 4 thuộc tính $\{A, B, C, D\}$

Thuật toán bắt đầu từ tập $L_1 = \{\{A\} | A \in R\}$ và tính L_2 từ L_1, L_3 từ $L_2, ...$, trong đó, L_ℓ là tập tất cả các tập thuộc tính có kích thước bằng ℓ. Với mỗi tập thuộc tính $X \in L_\ell$, thuật toán xây dựng một tập dự tuyển vế phải tối ưu $C^+(X)$. Tập $C^+(X)$ có thể được tính toán như là giao của các tập $C^+(X\backslash\{A\})$ với tất cả $A \in X$. Khi đó, phụ thuộc hàm $X\backslash\{A\} \to A, A \in X$ đầy đủ nếu $A \in C^+(X)$. Từ đó, để tìm các phụ thuộc hàm và phụ thuộc hàm xấp xỉ, thuật toán kiểm tra các phụ thuộc dạng $X\backslash\{A\} \to A$, trong đó $A \in X \cap C^+(X)$. Nếu $g_3(X\backslash\{A\} \to A) \leqslant \varepsilon$ thì thuật toán kết xuất phụ thuộc $X\backslash\{A\} \to A$ và loại bỏ A khỏi $C^+(X)$. Còn nếu $g_3(X\backslash\{A\} \to A) = 0$ thì thuật toán loại $R\backslash X$ khỏi $C^+(X)$. Việc loại bỏ này nhằm đảm bảo tính đầy đủ cho các phụ thuộc hàm ở mức tiếp theo. Khi tất cả các tập thuộc tính ở mức ℓ đã được xử lý, thì thuật toán sinh ra các tập thuộc tính ở mức $\ell + 1$ để tiếp tục phát hiện các phụ thuộc hàm và phụ thuộc hàm xấp xỉ.

Độ phức tạp của thuật toán Tane là $O((|r|.|R| + |R|^{2.5}).2^{|R|})$.

Chúng ta có thể tham khảo chi tiết thuật toán Tane trong [4].

4.3. Thuật toán AFD-Mine

Bây giờ, chúng tôi xây dựng thuật toán AFD-Mine để phát hiện các phụ thuộc hàm xấp xỉ. Thuật toán này dựa trên cơ sở thuật toán FD-Mine (phát hiện các phụ thuộc hàm) trong [9] và một số tính chất và luật cắt tỉa mới mà chúng tôi sẽ đề xuất dưới đây.

4.3.1. Chiến lược tìm kiếm

Thuật toán tìm kiếm theo từng mức của một dàn tìm kiếm (Hình 1) và những kết quả từ mức ℓ được sử dụng như là một tri thức để phát hiện các phụ thuộc hàm và phụ thuộc hàm xấp xỉ ở mức $\ell + 1$. Khi thuật toán xử lý tập X thì những phụ thuộc hàm và phụ thuộc hàm xấp xỉ dạng $X \to A, A \in R$ và $A \notin X$ được phát hiện. Trong quá trình tìm kiếm, thuật toán dùng một số luật cắt tỉa để thu hẹp được không gian tìm kiếm và đảm bảo các phụ thuộc tìm thấy là đầy đủ và không tầm thường.

4.3.2. Một số định nghĩa, tính chất, bổ đề sử dụng trong thuật toán

Trong [4, 7] đã đưa ra một bổ đề để tính tích của hai phân hoạch. Trên cơ sở đó, ta có một tính chất để tính tích của hai phân hoạch thu gọn như sau.

Tính chất 2. (Tích phân hoạch thu gọn) *Với bất kỳ $X, Y \subseteq R$, ta có $\hat{\pi}_{XY} = \hat{\pi}_X.\hat{\pi}_Y$, trong đó*

$$\hat{\pi}_X.\hat{\pi}_Y = \bigcup_{U\in\hat{\pi}_X} C(U), \text{ với } C(U) = \bigcup_{U\in\hat{\pi}_Y} \{(U \cap V) | U \cap V \neq \emptyset,\ |U \cap V| > 1\}.$$

Chứng minh:

- Với bất kỳ $Q \in \hat{\pi}_X.\hat{\pi}_Y$ sẽ $\exists U \in \hat{\pi}_X,\ V \in \hat{\pi}_Y$ sao cho $U \cap V = Q$. Khi đó, với mọi $t_1, t_2 \in Q$ suy ra $t_1, t_2 \in U$ và $t_1, t_2 \in V$. Do đó, $t_1[X] = t_2[X]$ và $t_1[Y] = t_2[Y]$, suy ra $t_1[XY] = t_2[XY]$. Từ $t_1[XY] = t_2[XY]$ suy ra $\exists S \in \hat{\pi}_{XY}$ sao cho $t_1, t_2 \in S$. Do đó, $Q \subset S$.

Ta có, $\forall Q \in \hat{\pi}_X.\hat{\pi}_Y,\ \exists S \in \pi_{XY}$ sao cho $Q \subset S$. Suy ra $\hat{\pi}_X.\hat{\pi}_Y$ mịn hơn $\hat{\pi}_{XY}$. (1)

- Mặt khác, với mọi $Q \in \hat{\pi}_{XY},\ t_1, t_2 \in Q$ thì $t_1[XY] = t_2[XY]$ suy ra $t_1[X] = t_2[X]$ và $t_1[Y] = t_2[Y]$. Do đó $\exists U \in \hat{\pi}_X,\ V \in \hat{\pi}_Y : t_1, t_2 \in U,\ t_1, t_2 \in V$ suy ra $t_1, t_2 \in U \cap V$.

Ta có, với bất kỳ $t_1, t_2 \in Q$ suy ra $t_1, t_2 \in U \cap V$, nên $Q \subset U \cap V \in \hat{\pi}_{XY}$.

Từ đó, suy ra $\hat{\pi}_{XY}$ mịn hơn $\hat{\pi}_X.\hat{\pi}_Y$. (2)

Từ (1), (2) suy ra $\hat{\pi}_X.\hat{\pi}_Y = \hat{\pi}_{XY}$ ■

Tính chất 3. (Hệ quả 2 trong [1]) *Nếu $X \to Y$ và $Y \to^k Z$ thì $X \to^{k'} Z$, với $k' \geqslant k$.*

Tính chất 4. (Hệ quả 3 trong [1]) *Nếu $X \to^k Y$ và $Y \to Z$ thì $X \to^{k'} Z$, với $k' \geqslant k$.*

Định nghĩa 8. [9] (Dự tuyển tương đương) Cho $r(R)$ và $X, Y \subseteq R$. Khi đó, nếu $X \to Y$ đúng và $Y \to X$ đúng, thì X và Y là hai dự tuyển tương đương, ký hiệu, $X \leftrightarrow Y$.

Từ các dự tuyển tương đương, ta có hai tính chất sau.

Tính chất 5. *Cho $r(R)$ và $X, Y, Z \subseteq R,\ X \leftrightarrow Y$. Khi đó, nếu $XW \to^k Z$ đúng, thì $YW \to^k Z$ đúng, với $k \in (0, 1]$.*

Chứng minh. Ta có $Y \to X$ đúng nên $YW \to XW$ đúng. Mà $XW \to^k Z$ đúng, suy ra $YW \to^{k'} Z$ đúng, với $k' \geqslant k$ và $k, k' \in (0, 1]$ (theo Tính chất 3). (3)

Ngược lại, nếu $X \to Y$ đúng và $YW \to^{k'} Z$ đúng, thì chứng minh tương tự như trên, ta được $XW \to^k Z$ đúng, với $k \geqslant k'$ và $k, k' \in (0, 1]$. (4)

Từ (3) và (4) suy ra Tính chất 5 được chứng minh. ■

Tính chất 6. *Cho $r(R)$ và $X, Y, Z \subseteq R,\ X \leftrightarrow Y$. Khi đó, nếu $WZ \to^k X$ đúng, thì $WZ \to^k Y$ đúng, với $k \in (0, 1]$.*

Chứng minh. Ta có $WZ \to^k X$ đúng. Mà $X \to Y$ đúng, suy ra $WZ \to^{k'} Y$ đúng, với $k' \geqslant k$ và $k, k' \in (0, 1]$ (theo Tính chất 4). (5)

Ngược lại, với $WZ \to^{k'} Y$ đúng và $Y \to X$ đúng, thì $WZ \to^k X$ đúng, với $k \geqslant k'$ ■(6)

Từ (5) và (6) suy ra Tính chất 6 được chứng minh. ■

Từ đó, nếu $X \leftrightarrow Y$, thì ta có thể loại bỏ Y khỏi tập các dự tuyển mà không ảnh hưởng đến việc phát hiện các phụ thuộc hàm và phụ thuộc hàm xấp xỉ.

Nhận xét 2. Trong trường hợp $k = 1$, thì Tính chất 5 chính là Bổ đề 3.1 và Tính Chất 6 là Bổ đề 3.2 trong [9].

Bổ đề 2. [9] *$X, Y, Z \subseteq R,\ Z = X \cap Y$. Khi đó, nếu $X^+ \backslash X \supseteq Y \backslash Z$ và $Y^+ \backslash Y \supseteq X \backslash Z$, thì $X \leftrightarrow Y$.*

Nhận xét 3. Để xây dựng luật cắt tỉa, trong thuật toán FD-Mine [9] phát biểu Tính chất 3.2 như sau: "*Cho* $X, Y \subseteq R$. *Khi đó:* $(X^+\backslash X) \cup (Y^+\backslash Y) \subseteq (XY)^+\backslash XY$".

Chúng tôi cho rằng tính chất này không đúng và đưa ra phản ví dụ như sau:

Phản ví dụ 1. Cho $r(A, B, C, D, E)$ và $F = \{A \to BD, C \to AE, DE \to B\}$.

Lấy $X = \{A\}, Y = \{C\}$. Khi đó: $X^+ = \{A, B, D\} \Rightarrow X^+\backslash X = \{B, D\}$,

$Y^+ = \{C, E, A, B, D\} \Rightarrow Y^+\backslash Y = \{A, E, B, D\}$.

Suy ra $(X^+\backslash X) \cup (Y^+\backslash Y) = \{A, E, B, D\}$.

Mặt khác $(XY)^+ = \{A, C, B, D, E\} \Rightarrow (XY)^+\backslash XY = \{B, D, E\}$.

Ta thấy rằng $(X^+\backslash X) \cup (Y^+\backslash Y) \not\subset (XY)^+\backslash XY$.

Nhận xét 4. Cho $X, Y \subseteq R$. Khi đó: $X^+\backslash XY \cup Y^+\backslash XY \subseteq (XY)^+\backslash XY$.

Tính chất 7. [3] (Siêu khoá) *Cho* $X \subseteq R$. *Khi đó, nếu* $X^+ = R$, *thì* X *là một siêu khoá.*

4.3.3. Thu hẹp không gian tìm kiếm

Lý thuyết phụ thuộc hàm cho chúng ta suy diễn một phụ thuộc hàm từ các phụ thuộc hàm khác. Vì vậy, ta có thể tỉa các dự tuyển thỏa một số điều kiện nào đó nhằm tránh những xử lý không cần thiết, giảm được thời gian và không gian tìm kiếm. Ta có các luật cắt tỉa sau:

Luật cắt tỉa 1. [9] Nếu $X \leftrightarrow Y$, thì Y có thể được cắt tỉa khỏi dàn tìm kiếm.

Luật cắt tỉa 2. [9] Nếu X là khóa, thì X có thể được tỉa khỏi dàn tìm kiếm.

Luật cắt tỉa 3. Các phụ thuộc hàm $XY \to \{X^+\backslash XY \cup Y^+\backslash XY\}$ không cần kiểm tra (theo Nhận xét 4).

Nhận xét 5. Luật cắt tỉa 3 chỉ áp dụng được với phụ thuộc hàm truyền thống. Đối với phụ thuộc hàm xấp xỉ, ta xây dựng một tập $D(X)$:

$$D(X) = \{A \in R | 0 < g_3(X \to A) \leqslant \varepsilon,\ A \notin X,\ \varepsilon \in (0, 1],\ \varepsilon \text{ là ngưỡng xấp xỉ}\},$$

hay $D(X)$ có thể được định nghĩa theo độ phụ thuộc k như sau:

$$D(X) = \{A \in R | X \to^k A \text{ đúng}, \xi \leqslant k < 1,\ \xi \in [0, 1),\ \xi \text{ là ngưỡng phụ thuộc}\}$$

với

$$\xi = 1 - \varepsilon - \frac{\sum\limits_{U \in \pi_X} \max\{|V| | V \in \pi_{X \cup \{A\}},\ V \subset U\}}{|r|} \text{ (theo Tính chất 1).}$$

Với cách xây dựng $D(X)$ như trên, ta có tính chất sau.

Tính chất 8. $\forall X, Y \subseteq R,\ D(X) \cup D(Y) \subseteq D(XY)$.

Chứng minh. $\forall A \in D(X) \cup D(Y) \Rightarrow A \in D(X)$ hay $A \in D(Y)$.

- Nếu $A \in D(X)$ thì $X \to^k A$ đúng, $k \geqslant \xi$. Mà $XY \to X$ đúng nên $XY \to^{k'} A$ đúng, với $k' \geqslant k \geqslant \xi$ (theo Tính chất 3) suy ra $A \in D(XY)$.

- Nếu $A \in D(Y)$ thì $Y \to^{k'} A$ đúng, $k' \geqslant \xi$. Mà $XY \to Y$ đúng nên $XY \to^{k''} A$ đúng, với $k'' \geqslant k' \geqslant \xi$ suy ra $A \in D(XY)$.

Ta có, $\forall A \in D(X) \cup D(Y) \Rightarrow A \in D(XY)$. Vậy, $D(X) \cup D(Y) \subseteq D(XY)$. ■

Từ đó, để thu hẹp không gian tìm kiếm đối với các phụ thuộc hàm xấp xỉ, ta có một luật cắt tỉa như sau.

Luật cắt tỉa 4. Những phụ thuộc $XY \to^{k'} \{D(X) \cup D(Y)\},\ k' \geqslant \xi,\ \xi \in (0, 1)$ (hay $XY \to \{D(X) \cup D(Y)\}$ ứng với ngưỡng lỗi ε) không cần kiểm tra.

Luật cắt tỉa 5. [9] Cho $r(A_1, ..., A_n)$. Khi đó, nếu tồn tại một tập con $B_1B_2...B_{\ell-1}$ chứa trong $A_1A_2...A_\ell$ sao cho $B_1B_2...B_{\ell-1} \to B_\ell$ đúng, $(B_\ell \in \{A_1, ..., A_\ell\}$ và $A_i \neq A_j,\ B_i \neq B_j, \forall i \neq j)$, thì ta không cần kiểm tra các phụ thuộc hàm $A_1A_2...A_l \to A_j, \forall j = \ell + 1, ..., n$.

4.3.4. Thuật toán

Thông qua dàn tìm kiếm, tại mức $L_1 = \{\{A\}|A \in R\}$, tất cả các phụ thuộc hàm và phụ thuộc hàm xấp xỉ dạng $X \to A,\ |X| = 1,\ A \in R,\ A \notin X$ được phát hiện và lưu trữ trong tập F_1. Tiếp theo, F_1 và L_1 được sử dụng để sinh tiếp mức L_2. Tại mức L_2, các phụ thuộc dạng $X \to A,\ |X| = 2,\ A \in R,\ A \notin X$ được phát hiện. Cứ tiếp tục quá trình như thế cho đến khi tất cả những dự tuyển ở mức $L_{|R|-1}$ được kiểm tra hoặc $L_\ell = \{X\big||X| = \ell\} = \emptyset,\ \ell \leqslant n - 1$.

Một số định nghĩa sử dụng trong thuật toán:

Closure'(X): Bao đóng không tầm thường của tập X.

FDset: Tập những phụ thuộc hàm.

EQset: Tập các dự tuyển tương đương dạng $X \leftrightarrow Y$.

KEYset: Tập những khóa.

L_ℓ: Tập những dự tuyển tại mức ℓ.

AFDset: Tập những phụ thuộc hàm xấp xỉ.

COM-PARTset(ℓ): Tập các phân hoạch thu gọn có được tại mức ℓ.

- *Thuật toán chính*

Đầu vào: $r(R),\ R = \{A_1, A_2, ..., A_n\}$.

Đầu ra: Các FD và AFD đầy đủ, không tầm thường trên r.

$L_1 := R$

$\ell := 1$

While $L_\ell \neq \emptyset$ do

For each $X_i \in L_\ell$ do

COMPUTE-NONTRIVIAL-CLOSURE(X_i)

OBTAIN-FDANDKEY(X_i)

OBTAIN-EQSET(L_ℓ)

PRUNE-CANDIDATES(L_ℓ)

GENERATE-NEXT-LEVEL (L_ℓ)

Display ($FDset, AFDset, EQset, KEYSet$)

- *Sinh mức*

Từ các luật cắt tỉa trên, ta có thủ tục sinh mức như sau:

Đầu vào: L_ℓ.

Đầu ra:

- $L_{\ell+1}$.

- Các tập $Closure'[X_{ij}]$ và $D[X_{ij}]$ có được từ $Closure'[X_i]$, $Closure'[X_j]$, $D[X_i]$ và $D[X_j]$ với $X_i,\ X_j \in L_\ell$.

- Tập $Keyset$ có được từ mức 1 đến mức ℓ.

Procedure GENERATE-NEXT-LEVEL(L_ℓ)

1 For each $X_i \in L_\ell$ do

2 For each $(X_j \in L_\ell)$ and $(i < j)$ do

3 If $(X_i[1] = X_j[1], ..., X_i[\ell - 1] = X_j[\ell - 1])$ and $(X_i[\ell] < X_j[\ell])$ then

4 $X_{ij} := X_i \cup X_j$

5 If $\exists A \in X_{ij} : (X_{ij} \backslash \{A\} \to A) \in FDset$ then xóa X_{ij}

6 Else

7 $Closure'[X_{ij}] := (Closure'[X_i])\backslash\{X_j\} \cup (Closure'[X_j])\backslash\{X_i\}$
8 $D[X_{ij}] := D[X_i] \cup D[X_j]$
9 If $(R = X_{ij} \cup Closure'[X_{ij}])$ then $KEYset := KEYset \cup X_{ij}$
10 Else
11 $L_{\ell+1} := L_{\ell+1} \cup X_{ij}$

Thủ tục sinh mức GENERATE-NEXT-LEVEL dùng Luật cắt tỉa 5 ở dòng 5, Luật cắt tỉa 3 ở dòng 7 và Luật cắt tỉa 4 ở dòng 8.

- Tính bao đóng không tầm thường, dự tuyển vế phải của AFD và sinh các phân hoạch

Bây giờ, chúng tôi xây dựng thủ tục để sinh các phân hoạch, tính bao đóng không tầm thường và các dự tuyển vế phải của phụ thuộc hàm xấp xỉ như sau:

Đầu vào: Tập thuộc tính Xi thuộc L_ℓ

Đầu ra: - Các phân hoạch có được ở mức $\ell + 1$
- Các tập $Closure'[X_i], D[X_i]$.

Procedure COMPUTE-NONTRIVIAL-CLOSURE(X_i)
1 For each $A \in R - X_i - Closure'[X_i]$
2 If $\hat{\pi}_{X_iA} \notin COM - PARTset(\ell + 1)$ then
3 $\hat{\pi}_{X_iA} := \hat{\pi}_{X_i}.\hat{\pi}_A$ // $\hat{\pi}_{X_i} \in COM - PARTset(\ell)$
4 $COM - PARTset(\ell + 1) := COM - PARTset(\ell + 1) \cup \{\hat{\pi}_{X_iA}\}$
5 If $A \notin D[X_i]$ then
6 If $(g_3(X_i \to A) \leqslant \varepsilon)$ and $(g_3(X_i \to A) > 0)$ then $D[X_i] := D[X_i] \cup \{A\}$
7 If $(g_3(X_i \to A) = 0)$ then $Closure'[X_i] := Closure'[X_i] \cup \{A\}$
8 Huỷ $COM - PARTset(\ell)$

Thuật toán tính các phân hoạch thu gọn $\hat{\pi}_A$, $A \in R$ trực tiếp từ CSDL. Các phân hoạch thu gọn $\hat{\pi}_X$, $|X| \geqslant 2$ được tính như là tích của hai phân hoạch đã có ở các mức trước (theo Tính chất 2). Trong thủ tục này, ta dùng các phân hoạch thu gọn thay cho các phân hoạch, điều này làm giảm được độ phức tạp tính toán.

Ở thuật toán FD-Mine trong [9], các phân hoạch được tính sau khi một tập thuộc tính mức mới được sinh ra. Tuy nhiên, việc tính toán này chưa hiệu quả vì khi ta kiểm tra các phụ thuộc hàm ở mức ℓ, thì đòi hỏi chúng ta phải có các phân hoạch của các tập thuộc tính ở mức ℓ và $\ell + 1$. Do đó, chúng tôi tính toán các phân hoạch của các tập thuộc tính ở mức $\ell + 1$ ngay trong thủ tục COMPUTE-NONTRIVIAL-CLOSURE của AFD-Mine, trước khi xem xét các phụ thuộc hàm ở mức ℓ. Và, việc tính toán các phân hoạch như vậy là đúng đắn. Để làm sáng tỏ điều này, chúng tôi đề xuất một tính chất sau.

Tính chất 9. *Nếu các tập thuộc tính mức $(\ell + 1)$ được sinh ra ở thủ tục sinh mức GENERATE-NEXT-LEVEL trong AFD-Mine, thì các phân hoạch của chúng đã được tính toán trong thủ tục COMPUTE-NONTRIVIAL-CLOSURE ở mức ℓ.*

Chứng minh. Giả sử $\exists X_{ij} \in L_{\ell+1}$, X_{ij} được sinh ra từ X_i, X_j với $X_i, X_j \in L_\ell$ sao cho $\hat{\pi}_{X_{ij}}$ không được tính toán trong thủ tục COMPUTE-NONTRIVIAL-CLOSURE ở mức ℓ của AFD - Mine, nghĩa là $\hat{\pi}_{X_{ij}} \notin COM - PARTset(\ell + 1)$.

Từ $\hat{\pi}_{X_{ij}} \notin COM - PARTset(\ell + 1)$, suy ra $\exists A \in Closure'[X_i]$ sao cho $X_{ij} = X_i \cup \{A\}$, nên $X_i \to A$ đúng hay $\exists A \in X_{ij} : X_{ij}\backslash\{A\} \to A$ đúng. Do đó $X_{ij} \notin L_{\ell+1}$. Điều này mâu thuẩn với giả thiết $X_{ij} \in L_{\ell+1}$. Vậy, Tính chất 9 được chứng minh. ■

- Tính các phụ thuộc hàm, phụ thuộc hàm xấp xỉ và khóa

Đầu vào: tập thuộc tính X_i thuộc L_ℓ.

Đầu ra: Tập các FDset, AFDset, Keyset có được từ mức 1 đến mức ℓ.

Procedure OBTAIN-FDANDKEY(X_i)

$FDset := FDset \cup \{X_i \to Closure'[X_i]\}$

$AFDset := AFDset \cup \{X_i \to D[X_i]\}$

If $(R = X_i \cup Closure'[X_i])$ then $KEYset := KEYset \cup \{X_i\}$

- *Tính các dự tuyển tương đương [9]*

Đầu vào: L_ℓ.

Đầu ra: Tập các dự tuyển tương đương EQset có được từ mức 1 đến mức ℓ.

Procedure OBTAIN-EQSET(L_ℓ)

For each $X_i \in L_\ell$ do

For each $((X \to Closure'[X]) \in FDset)$ and $(X \neq X_i)$ do

$Z := X \cap X_i$

If $(Closure'[X] \supseteq X_i - Z)$ and $(Closure'[X_i] \supseteq X - Z)$ then

$EQset := EQset \cup \{X \leftrightarrow X_i\}$

- *Cắt tỉa các dự tuyển [9]*

Đầu vào: L_ℓ.

Đầu ra: L_ℓ.

Procedure PRUNE-CANDIDATES(L_ℓ)

For each $X_i \in L_\ell$ do

If $\exists X \leftrightarrow X_i \in EQset$ then xoá X_i khỏi L_ℓ //Luật cắt tỉa 1

If $\exists X_i \in KEYset$ then xóa X_i khỏi L_ℓ

Nếu X là khóa , thì thuật toán tỉa X khỏi dàn tìm kiếm nhằm đảm bảo tính đầy đủ cho các phụ thuộc hàm và phụ thuộc hàm xấp xỉ ở mức tiếp theo.

4.3.5. Độ phức tạp thuật toán AFD-Mine

Độ phức tạp của thuật toán AFD-Mine phụ thuộc vào số thuộc tính $|R|$, kích thước của các mức và số bộ $|r|$. Trường hợp xấu nhất xảy ra khi không có phụ thuộc hàm nào được tìm thấy và tất cả những kết hợp của các thuộc tính đều được kiểm tra. Khi đó, ta có $2^{|R|}$ tập thuộc tính được sinh ra và việc tính toán một phân hoạch cho một tập thuộc tính mất thời gian $O(|r|)$. Do đó, độ phức tạp của thuật toán AFD-Mine trong trường hợp xấu nhất là $O(|r|.2^{|R|})$.

4.4. Mối liên hệ giữa Tane và AFD-Mine

Qua hai thuật toán Tane và AFD-Mine, ta thấy rằng giữa chúng có mối liên hệ với nhau như sau:

Tính chất 10. *Luật cắt tỉa 3 của AFD-Mine tương đương với việc loại bỏ A khỏi $C^+(X)$ khi phụ thuộc hàm $X\backslash\{A\} \to A$ đúng trong Tane.*

Chứng minh:

- Giả sử, các phụ thuộc hàm $XY \to (X^+\backslash X)\backslash Y \cup (Y^+\backslash Y)\backslash X$ (Luật cắt tỉa 3) không cần kiểm tra ở AFD-Mine, ta chứng minh chúng cũng không cần kiểm tra ở Tane.

Thật vậy, $\forall A \in (X^+\backslash X)\backslash Y \cup (Y^+\backslash Y)\backslash X$:

+ Với bất kỳ $A \in (X^+\backslash X)\backslash Y$ suy ra $X \to A$ đúng $\Rightarrow (X \cup \{A\})\backslash\{A\} \to A$ đúng. Do đó, theo thuật toán Tane, ta có thể loại A khỏi $C^+(X \cup \{A\})$, suy ra $A \notin C^+(X \cup \{A\})$. Mà $C^+(XY \cup \{A\}) \subseteq C^+(X \cup \{A\})$ (do $X \cup \{A\} \subseteq XY \cup \{A\}$), suy ra $A \notin C^+(XY \cup \{A\})$. Do

đó, ta không cần kiểm tra các phụ thuộc hàm $(XY \cup \{A\})\backslash\{A\} \to A$ (theo thuật toán Tane), suy ra các phụ thuộc hàm dạng $XY \to (X^+\backslash X)\backslash Y$ không cần kiểm tra ở thuật toán Tane.

+ Với bất kỳ $A \in (Y^+\backslash Y)\backslash X$, tương tự như trên ta cũng không cần kiểm tra các phụ thuộc hàm $XY \to (Y^+\backslash Y)\backslash X$ ở thuật toán Tane.

Do đó, các phụ thuộc hàm $XY \to (X^+\backslash X)\backslash Y \cup (Y^+\backslash Y)\backslash X$ không kiểm tra ở thuật toán Tane. (7)

- Giả sử loại A khỏi $C^+(X)$ khi $X\backslash\{A\} \to A$ đúng, $A \in X$, $A \in C^+(X)$. Khi đó, $A \notin C^+(Y)$, $\forall Y \supset X$ và các phụ thuộc hàm $Y\backslash\{A\} \to A$, $Y \supset X$ không cần kiểm tra ở thuật toán Tane. Ta chứng minh các phụ thuộc hàm này cũng không cần kiểm tra ở AFD-Mine.

Thật vậy, từ $X\backslash\{A\} \to A$ đúng và $A \in X$, suy ra $A \in (X\backslash\{A\})^+$. Do đó, $A \in (Y\backslash\{A\})^+$, $\forall Y\backslash\{A\} \supset X\backslash\{A\}$. Điều này có nghĩa là các phụ thuộc hàm $Y\backslash\{A\} \to A$ luôn đúng ở AFD-Mine và ta không cần kiểm tra chúng. (8)

Từ (7) và (8) suy ra Tính chất 10 được chứng minh. ■

Tính chất 11. *Với bất kỳ hai mức $(\ell - 1)$ và ℓ nào đó, Luật cắt tỉa 5 của thuật toán AFD-Mine tương đương với việc loại bỏ $R\backslash X$ khỏi $C^+(X)$ khi một phụ thuộc hàm $X\backslash\{A\} \to A$ đúng ở Tane.*

Chứng minh:

- Giả sử $\exists B_1B_2...B_{\ell-1} \subset A_1A_2...A_\ell$ sao cho $B_1B_2...B_{\ell-1} \to B_\ell$ đúng. Khi đó, ở AFD-Mine, các phụ thuộc hàm $A_1A_2...A_\ell \to A_j$, $j = \ell + 1, ..., n$ không cần kiểm tra. Ta chứng minh, các phụ thuộc hàm này cũng không cần kiểm tra ở thuật toán Tane.

Thật vậy, nếu $\exists B_1B_2...B_{\ell-1} \subset A_1A_2...A_\ell$ sao cho $B_1B_2...B_{\ell-1} \to B_\ell$ đúng $(B_\ell \in \{A_1, A_2, ..., A_\ell\})$ suy ra $(B_1B_2...B_{\ell-1}B_\ell)\backslash\{B_\ell\} \to B_\ell$ đúng.

Đặt $X = B_1B_2...B_{\ell-1}B_\ell$, $A = B_\ell$ suy ra $X\backslash\{A\} \to A$ đúng, nên ta có thể loại $R\backslash X$ khỏi $C^+(X)$ ở thuật toán Tane. Do đó $R\backslash X = \{A_{\ell+1}, ..., A_n\} \notin C^+(X)$, suy ra $A_{\ell+1} \notin C^+(X \cup \{A_{\ell+1}\}), ..., A_n \notin C^+(X \cup \{A_n\})$. Từ đó, ta không cần kiểm tra các phụ thuộc hàm $X = A_1A_2...A_\ell \to A_j$, $j = \ell + 1, ..., n$ ở thuật toán Tane. (9)

- Ngược lại, giả sử $X = A_1A_2...A_\ell$ và ta loại $R\backslash\{X\}$ khỏi $C^+(X)$ khi $X\backslash\{A\} \to A$, $A \in X$ đúng, nghĩa là loại các A_j, $j = \ell + 1, ..., n$ khỏi $C^+(X)$. Khi đó, $A_{\ell+1} \notin C^+(X \cup \{A_{\ell+1}\}), ..., A_n \notin C^+(X \cup \{A_n\})$, suy ra $A_1A_2...A_\ell \to A_j$, $j = \ell + 1, ..., n$ không cần kiểm tra ở Tane. Ta chứng minh, các phụ thuộc hàm này cũng không cần kiểm tra trong AFD-Mine.

Thật vậy, từ $X\backslash\{A\} \to A$, $A \in X$ đúng suy ra tồn tại $B_\ell = A$, $B_\ell \in A_1A_2...A_\ell$ sao cho $(A_1A_2...A_\ell)\backslash\{B_\ell\} \to B_\ell$ đúng, nghĩa là $\exists B_1B_2...B_{\ell-1} \subset A_1A_2...A_\ell : B_1B_2...B_{\ell-1} \to B_\ell$ đúng. Do đó, theo Luật cắt tỉa 5, các phụ thuộc hàm $A_1A_2...A_\ell \to A_j$, $j = \ell + 1, ..., n$ không cần kiểm tra ở thuật toán AFD-Mine. (10)

Từ (9) và (10) suy ra Tính chất 11 được chứng minh. ■

Ngoài những luật cắt tỉa giống với thuật toán Tane, thuật toán AFD-Mine còn sử dụng thêm một luật cắt tỉa nữa, gọi là cắt tỉa các dự tuyển tương đương (Luật cắt tỉa 1). Điều này giúp thuật toán AFD-Mine giảm được số lần kiểm tra các phụ thuộc hàm và phụ thuộc hàm xấp xỉ trong CSDL hơn thuật toán Tane.

5. KẾT LUẬN

Việc nghiên cứu và phát hiện các phụ thuộc hàm xấp xỉ trong CSDL là một trong những vấn đề đang được các nhà khoa học quan tâm. Trong bài báo này, chúng tôi đã nghiên cứu

và xây dựng một thuật toán để phát hiện các phụ thuộc hàm xấp xỉ trong CSDL theo cách tiếp cận lý thuyết tập thô và đã đạt được một số kết quả sau:

1. Đề xuất Tính chất 1 để thể hiện mối quan hệ giữa độ phụ thuộc k và độ đo lỗi g_3 của phụ thuộc hàm. Tính chất này giúp ta hiểu sâu sắc hơn về phụ thuộc hàm xấp xỉ và thuận tiện trong quá trình phân tích, xây dựng thuật toán phát hiện các phụ thuộc hàm xấp xỉ.

2. Sau khi kiểm chứng độ đo lỗi g_3, chúng tôi thấy rằng nó không còn đúng trên phân hoạch thu gọn. Từ đó, chúng tôi xây dựng một công thức mới để tính độ đo lỗi g_3 trên các phân hoạch thu gọn.

3. Đưa ra Phản ví dụ 1 để minh chứng sai sót của Tính chất 3.2 trong thuật toán "FD-Mine [9]". Sau đó, ta có Tính chất 8, Luật cắt tỉa 3 và Luật cắt tỉa 4.

4. Tính chất 5 và Tính chất 6 chứng tỏ việc cắt tỉa dự tuyển tương đương không ảnh hưởng đến các phụ thuộc hàm xấp xỉ.

5. Xây dựng Thủ tục COMPUTE-NONTRIVIAL-CLOSURE (sinh các phân hoạch, tính bao đóng không tầm thường và các dự tuyển vế phải của AFD). Sau đó, đề xuất Tính chất 9 để chứng tỏ rằng việc xây dựng thủ tục này là đúng đắn.

6. Tìm mối liên hệ giữa hai thuật toán Tane và AFD-Mine.

Lời cảm ơn. Tôi xin trân trọng cảm ơn PGS. TS Hồ Thuần và các phản biển đã có những góp ý xác đáng, giúp tôi hoàn thiện bài báo này.

TÀI LIỆU THAM KHẢO

[1] L.B. Cristofor, A Rough Set Based Generalization of Functional Denpendencies, Department of Math and Computer Science, UMass/Boston, 2000.

[2] J. Demetrovics, L. Libkin, I.B. Muchnik, Functional Dependencies in Relational Databases: A lattice point of view, *Discrete Applied Mathematics* **40** (1992) 155–185.

[3] Ho Thuan, Contribution to The Theory of Relational Databases, Tanulmanyok 184/1986, Studies 1984/1986.

[4] Y. Huhtala, J. Karkkainen, P. Porkka, H. Toivonnen, Tane: An Efficient Algorithm for Discovery Functional and Approxiamate Dependencies, *The Computer Journal* **42** (3) (1999) 100–111.

[5] J. Kivinen, H. Mannila, Approximate Inference of Functional Dependencies from Relations, *Theoretical Computer Science* **149** (1) (1995) 129–149.

[6] Nguyễn Đăng Khoa, Vũ Huy Hoàng , Phụ thuộc hàm suy rộng trên cơ sở lý thuyết tập thô, *Tạp chí Tin học và Điều khiển học* **20** (2004) 91–98.

[7] Phan Trí Tuệ , "Nghiên cứu và phát hiện các phụ thuộc hàm suy rộng trong CSDL theo cách tiếp cận tập thô", Luận văn thạc sĩ, Trường Đại Học Công Nghệ, Hà Nội, 2004.

[8] J.D. Ullman, *Principles of Database and Knowledge - Base systems*, Vol. 1&2, Computer Science Press, 1986.

[9] H. Yao, H.J. Hamilton, C.J. Butz, *FD_Mine: Discovering Functional Dependencies in a Database Using Equivalences*, Department of Computer Science, University of Regina, Canada, 2002.

Nhận bài ngày 30 - 5 - 2006
Nhận lại sau sửa ngày 12 - 12 -2006