TẠP CHÍ KHOA HỌC, Đại học Huế, Tập 74A, Số 5, (2012), 39-52

LUẬT KẾT HỢP MỜ DỰA TRÊN NGỮ NGHĨA ĐẠI SỐ GIA TỬ *

Nguyễn Công Hào[1], Nguyễn Công Đoàn[2]

[1] Trung tâm Công nghệ thông tin, Đại học Huế

[2]Phòng Công nghệ thông tin, Huyện ủy Gò Dầu, Tây Ninh

Tóm tắt. Luật kết hợp mờ đã được nhiều tác giả quan tâm nghiên cứu theo nhiều cách tiếp cận khác nhau và đã có nhiều kết quả công bố. Tuy nhiên, đối với việc khai phá dữ liệu mờ với nhiều kiểu dữ liệu khác nhau để tìm ra luật kết hợp mờ nào đó phù hợp là vấn đề khó và phức tạp. Vì vậy, trong bài báo này, với nhiều ưu điểm của đại số gia tử, chúng tôi trình bày một phương pháp mới để xử lý luật kết hợp mờ sử dụng đại số gia tử đơn giản và trực quan hơn.

1. Đặt vấn đề

Một trong những chức năng được đề cập rất nhiều trong khai phá dữ liệu là khai phá sự kết hợp giữa các mẫu trong dữ liệu hay còn gọi là luật kết hợp. Trong thời kỳ đầu luật kết hợp chỉ đơn giản là khai phá sự hiện diện của một mẫu *A* thì dẫn đến sự xuất hiện mẫu *B*. Sau đó, luật kết hợp được phát triển để khai phá quan hệ có thuộc tính số lượng giữa các mẫu và được gọi là luật kết hợp số lượng. Một số khái niệm được bổ sung vào dữ liệu để khai phá luật kết hợp ở mức tổng quát, ...

Khai phá luật kết hợp là một trong những phương pháp khai phá tri thức từ CSDL và đã nhận được nhiều sự quan tâm trong giới khoa học máy tính và công nghệ tri thức. Thuật toán đầu tiên và nổi tiếng là Apriori do tác giả Agrawal cùng các cộng sự đề xuất, ban đầu nó được ứng dụng vào việc khai phá luật kết hợp trong lĩnh vực thương mại. Luật kết hợp không chỉ dừng lại những ứng dụng trong thương mại mà đã có những ứng dụng rộng rãi trong các lĩnh vực khác như trong y khoa, quản lý, thương mại và công nghiệp… Một minh họa trong CSDL của ngành y tế có một luật "*Nếu* có thai *thì* người đó là Phụ nữ" luật này đúng với độ tin cậy 100%, nhưng cũng chính vì vậy mà đây không phải là điều mới mẻ cần phải khai phá. Các luật mới đúng 100% rất hiếm khi xảy ra trong quá trình hoạt động và nhập liệu, mà thường là đã được phân tích rất kỹ khi xây dựng. Luật kết hợp cũng là luật suy diễn "*Nếu … thì …*" nhưng sẽ có thêm từ tố *thông thường* hoặc *gần như* hoặc *phần lớn* hoặc *số phần trăm* nào đó.

Việc xử lý dữ liệu mờ để khai phá dữ liệu trong các luật kết hợp mờ chủ yếu dựa

*Nghiên cứu được tài trợ bởi Quỹ hỗ trợ phát triển KHCN Quốc gia

trên lý thuyết. Tuy nhiên, theo cách sử dụng tập mờ có nhiều hạn chế do việc xây dựng các hàm thuộc và xấp xỉ các giá trị ngôn ngữ bởi các tập mờ còn mang tính chủ quan, phụ thuộc nhiều vào ý kiến chuyên gia cho nên dễ mất thông tin. Mặt khác, bản thân các giá trị ngôn ngữ có một cấu trúc thứ tự nhưng khi ánh xạ gán nghĩa sang tập mờ, không bảo toàn cấu trúc đó nữa.

Vì vậy, để khắc phục hạn chế trên, bài báo tập trung nghiên cứu về luật kết hợp mờ dựa trên đại số gia tử (ĐSGT) nhằm mô phỏng chính xác hơn cấu trúc ngữ nghĩa của khái niệm mờ.

2. Một số kiến thức cơ sở

Cho một ĐSGT tuyến tính đầy đủ $\underline{\boldsymbol{X}} = (X, G, H, \Sigma, \Phi, \leq)$, trong đó $Dom(X) = X$ là miền các giá trị ngôn ngữ của thuộc tính ngôn ngữ X được sinh tự do từ tập các phần thử sinh $G = \{1, c^+, W, c^-, 0\}$ bằng việc tác động tự do các phép toán một ngôi trong tập H, Σ và Φ là hai phép tính với ngữ nghĩa là cận trên đúng và cận dưới đúng của tập $H(x)$, tức là $\Sigma x = supremum\ H(x)$ and $\Phi x = infimum\ H(x)$, trong đó $H(x)$ là tập các phần từ sinh ra từ x, còn quan hệ $\leq$ là quan hệ sắp thứ tự tuyến tính trên X cảm sinh từ ngữ nghĩa của ngôn ngữ. Ví dụ, nếu ta có thuộc tính *Luong* là "Lương thu nhập của nhân viên trong một tháng", thì *Dom*(Luong) = {*high, low, very high, more high, possibly high, very low, possibly low, less low,...*}, $G = \{0, low, W, high, 1\}$, $H = \{very, more, possibly, less\}$ và $\leq$ một quan hệ thứ tự cảm sinh từ ngữ nghĩa của các từ trong *Dom*(Luong), chẳng hạn ta có *very high* > *high*, *more high* > *high*, *possibly high* < *high*, *less high* < *high*, ... Cho tập các gia tử $H = H^- \cup H^+$, trong đó $H^+ = \{h_1,..., h_p\}$ và $H^- = \{h_{-q}, ..., h_{-1}\}$, với $h_1 < ... < h_p$ và $h_{-1} < ... < h_{-q}$, trong đó $p, q > 1$. Ký hiệu fm: $\underline{\boldsymbol{X}} \to [0,1]$ là độ đo tính mờ trên ĐSGT $\underline{\boldsymbol{X}}$. Với mỗi $x \in \underline{\boldsymbol{X}}$, I($x$) là khoảng mờ của x và $|I(x)| = fm(x)$. Khi đó,

Định nghĩa 2.1. Với mỗi $x \in \underline{\boldsymbol{X}}$, độ dài của x được ký hiệu $|x|$ và xác định như sau:

(a) Nếu $x = c^+$ hoặc $x = c^-$ thì $|x| = 1$.

(b) Nếu $x = hx'$ thì $|x| = 1 + |x'|$, với mọi $h \in H$.

Mệnh đề 2.1. Độ đo tính mờ fm và độ đo tính mờ của gia tử $\mu(h)$, $\forall h \in H$, có các tính chất sau:

(a) $fm(hx) = \mu(h) fm(x)$, $\forall x \in \underline{\boldsymbol{X}}$

(b) $fm(c^-) + fm(c^+) = 1$

(c) $\sum_{-q \leq i \leq p, i \neq 0} fm(h_i c) = fm(c)$, *trong đó* $c \in \{c^-, c^+\}$

(d) $\sum_{-q \leq i \leq p, i \neq 0} fm(h_i x) = fm(x)$, $x \in \underline{\boldsymbol{X}}$

(e) $\sum \{\mu(h_i) : -q \leq i \leq -1\} = \alpha$ *và* $\sum \{\mu(h_i) : 1 \leq i \leq p\} = \beta$, *trong đó* $\alpha, \beta > 0$ *và* $\alpha + \beta = 1$.

Định nghĩa 2.2. Cho $I = \{i_1, i_2, \ldots..i_m\}$là tập những mục và $D = \{T_1, T_2, \ldots..T_n\}$ là một tập những giao tác với những mục trong I. Ta gọi một giao tác T chứa X nếu $X \subseteq T$, với X là tập một vài mục trong I. Một luật kết hợp là luật có dạng: $X \Rightarrow Y$, trong đó $X \subset I$, $Y \subset I$ và $X \cap Y = \varnothing$.

Luật $X \Rightarrow Y$ có độ hỗ trợ là tỷ lệ giao tác T có chứa $X \cap Y$, nó xác định cách thực hiện thường xuyên luật được áp dụng cho tập giao tác T: $supp(X \Rightarrow Y) = |X \cap Y|/n$, trong đó: $X \cap Y$ là số giao tác mà chứa tất cả mục của luật, n là tổng số giao tác. Độ hỗ trợ này là một biện pháp hữu ích để xác định xem một tập các mục xảy ra thường xuyên trong một CSDL hay không? Luật $X \Rightarrow Y$ có độ tin cậy mô tả tỷ lệ các giao tác chứa X và cũng chứa Y: $conf(X \Rightarrow Y) = |X \cap Y|/X$.

3. Một số thuật toán luật kết hợp mờ

Để thuận tiện trong việc trình bày thuật toán, chúng tôi sử dụng một số ký hiệu sau:

Bảng 3.1. *Các ký hiệu sử dụng trong thuật toán*

Ký hiệu	Ý nghĩa	Ký hiệu	Ý nghĩa
D	CSDL giao tác ban đầu	C_K	Tập mục ứng viên có k phần tử
I	Tập các thuộc tính trong D	L_K	Tập mục phổ biến có k phần tử
T	Tập các giao tác trong D	FI	Tập tất cả các tập mục phổ biến được khai phá
D_F	Tập dữ liệu giao tác được làm mờ	T_{FIP}	Cây tiên tố các mục mờ
I_F	Tập các thuộc tính (tập mục) trong D_F	C_1	Tập những mục trong tập dữ liệu
T_F	Tập các giao tác (bản ghi) trong D_F	$T_{FIP}L[k]$	Tập mục phổ biến có k phần tử trong cây T_{FIP}
C_k	Tập các thuộc tính có kích thước k	FI	Tập tất cả các tập mục phổ biến được khai phá
F_k	Tập các thuộc tính phổ biến có kích thước k	$Prefixitem$	Danh sách các mục tiền tố
F	Tập tất cả các thuộc tính phổ biến	$Item.Sub$	Trỏ đến mục con đầu tiên trong danh sách mục
α	Ngưỡng *minsupp*	$Item.Next$	Trỏ đến mục kế tiếp có cùng tiền tố với mục này
γ	Ngưỡng *minconf*		

3.1. Thuật toán luật kết hợp mờ

Thuật toán khai phá luật kết hợp mờ được thực hiện theo 3 bước chính như sau:

Bước 1: Chuyển đổi từ CSDL dạng quan hệ sang CSDL mờ, CSDL mờ được tính toán từ CSDL ban đầu thông qua hàm thuộc của các tập mờ tương ứng với từng thuộc tính.

Bước 2: Tìm tất cả các tập thuộc tính mờ phổ biến dạng $< X,A >$ có độ hỗ trợ lớn hơn độ hỗ trợ cực tiểu của người dùng nhập vào $fs(< X, A >) \geq minsupp$.

Bước 3: Sinh các luật kết hợp mờ tin cậy từ các tập phổ biến đã tìm thấy ở bước thứ hai. Nếu $< X,A >$ là một tập thuộc tính mờ phổ biến thì luật kết hợp mờ được sinh từ X có dạng: X' *ts* $A' \xrightarrow{fc} X \setminus X'$ *ts* $A \setminus A'$, trong đó X' là tập con khác rỗng của X. $X \setminus X'$ là hiệu của hai tập hợp X và X', fc là độ tin cậy của luật thỏa $fc \geq minconf$, A' là tập con khác rỗng của A và là tập mờ tương ứng với các thuộc tính trong X', $A \setminus A'$ là hiệu của hai tập hợp A và A'.

Vào: *CSDL D với tập thuộc tính I và tập giao tác T, ngưỡng minsupp, minconf.*

Ra: *Tập các luật kết hợp mờ tin cậy.*

Phương pháp:

(1) **begin**
(2) (D_F, I_F, T_F) = FuzzyMaterialization(D, I, T);
(3) F_1 = Counting(D_E, I_F, T_F, *minsupp*);
(4) k = 2;
(5) **while** ($F_{k-1} \neq \emptyset$) {
(6) C_k = Join($\mathbf{F_{k-1}}$);
(7) C_k = Prune(C_k);
(8) F_k = Cheking(C_k, D_F, *minsupp*);
(9) $F = F \cup F_k$;
(10) $k = k + 1$;
(11) }
(12) GenerateRules(F, *minconf*);
(13) **end**

Hàm FuzzyMaterialization(D, I, T): Thực hiện chuyển đổi từ CSDL D ban đầu sang CSDL D_F với các thuộc tính được gắn thêm các tập mờ và giá trị các thuộc tính ở các bản ghi T được ánh xạ thành một giá trị thuộc khoảng [0, 1] thông qua hàm thuộc của các tập mờ tương ứng với các thuộc tính. Giả sử thuộc tính *Số-lượng* được phân vào ba vùng mờ với tên gọi cho từng vùng mờ là {*thấp, trung bình, cao*}.

Hàm Counting(D_F, I_F, T_F, *minsupp*): Tạo ra F_1 là tất cả các tập phổ biến có một phần tử (lực lượng bằng 1). Các tập thuộc tính phổ biến này phải có độ hỗ trợ lớn hơn hoặc bằng *minsupp*.

Hàm Join(F_{k-1}): Thực hiện kết nối các cặp các thuộc tính mờ từ tập các thuộc

tính mờ phổ biến F_{k-1} phần tử (lực lượng $k-1$), cách kết nối sử dụng trong hàm Join được thể hiện thông qua ngôn ngữ SQL. Hàm Prune(C_k): Sử dụng tính chất "mọi tập con khác rỗng của tập phổ biến cũng là tập phổ biến và mọi tập chứa tập không phổ biến đều là tập không phổ biến", để cắt tỉa những thuộc tính nào trong C_k có tập con lực lượng $k-1$ không thuộc tập các tập thuộc tính phổ biến F_{k-1}. Hàm Checking(C_k, D_F, *minsupp*): Duyệt qua CSDL D_F để cập nhật độ hỗ trợ cho các tập thuộc tính trong C_k. Sau khi duyệt xong, Checking sẽ chỉ chọn những tập mục phổ biến (có độ hỗ trợ lớn hơn hoặc bằng *minsupp*) để đưa vào trong F_k. Hàm GenerateRules(F, *minconf*): Sinh luật kết hợp mờ tin cậy từ tập các tập phổ biến F.

3.2. Thuật toán luật kết hợp tổng quát mờ AFAR

Thuật toán này được phát triển từ thuật toán kinh điển Apriori phục vụ cho việc khai phá luật kết hợp mờ. Dạng thuật toán này sử dụng để xác định tập phổ biến dữ liệu trong khai phá dữ liệu.

Vào: D_F *tập dữ liệu giao tác được làm mờ, ngưỡng minsupp* α, *minconf* γ.

Ra: *Tập các luật kết hợp mờ được khai phá.*

Phương pháp:

(1) **begin**
(2) //Duyệt qua các tập mục trong D_F để tính $\sum Count$
(3) $C_1 = \{\emptyset\}$;
(4) $L_1 = \{\emptyset\}$;
(5) **foreach** $T_i \in D_F$ **do**
(6) **foreach** $item \in T_i$ **do**
(7) $C_1.item.Count \mathrel{+}= item.Count$; //Giá trị thuộc [0, 1]
(8) //Giai đoạn xác định L_1
(9) **foreach** $item \in C_1$ **do**
(10) **if** $(item.Count \geq \alpha)$ **then**
(11) $L_1 = L_1 \cup \{item\}$;
(12) $k := 2$; //chỉ mục của tập phổ biến
(13) **while** $(L_{k-1} \neq \emptyset)$ **do**
(14) $C_k = Phatsinh C_k(L_{k-1})$; //phát sinh C_k từ L_{k-1}
(15) **foreach** $itemset\ T_i \in D_F$ **do**
(16) **foreach** $itemset \in C_k$ **do**
(17) **if** $(itemset \in T_i)$ **do**
(18) min = MAX_FLOAT; //tìm min của các mục
(19) **foreach** $item \in itemset$ **do**
(20) **if** $(T_i.item.Count < min)$ **do**
(21) $min = T_i.item.Count$;
(22) $C_k.itemset.Count \mathrel{+}= min$;
(23) **end if**
(24) **end foreach**
(25) **end foreach**
(26) //xác định lại L_k
(27) **foreach** $itemset \in C_k$ **do**

(28) **if** $(itemset.Count \geq \alpha)$ **then**
(29) $L_k = L_k\{itemset\}$;
(30) **end while**
(31) $FI = \cup L_k$; //*FI* chứa tất cả các tập mục phổ biến *k* phần tử $(k > 1)$
(32) //tạo các luật từ tập phổ biến
(33) **foreach** $itemset \in FI$ **do**
(34) **foreach** $item \in itemset$ **do**
(35) $supp = itemset.Count$;
(36) $conf = itemset.Count / item.Count$;
(37) **if** $(conf \geq \gamma)$ **then**
(38) *Xuất luật*: $itemset \setminus \{item\} \Rightarrow item, supp, conf$
(39) **end foreach** *item*
(40) **end foreach** *itemset*
(41) **end**

Thủ tục phát sinh ứng viên *PhatSinhCk*:

(1) PhatSinh$C_k(L_{k-1})$
(2) **begin**
(3) **foreach** $i, j \in L_{k-1}$ **and** $i \neq j$ **do**
(4) **if** $(t.item_1 = j.item_1 \; and \; i.item_2 = j.item_2 \ldots i.item_{k-2} = j.item_{k-2} and \; i.item_{k-1} < j.item_{k-1}$ **do**
(5) //kết thành bộ tập mục có *k* phần tử
(6) $itemset = \{i.item_1, i.item_2, \ldots, i.item_{k-2}, i.item_{k-1}, j.item_{k-1}\}$
(7) //kiểm tra xem có quan hệ *anct/desct* ở bước *k = 2*
(8) **if** $(k = 2 \; and \; Ancestor(itemset.item_1, itemset.item_2))$
(9) **continue**;
(10) //kiểm tra xem tất cả phần tử trong tập con *k – 1* phần tử có thuộc về tập L_{k-1} hay không
(11) **if** $(\forall \; sub\, item_{k-1} \; của \; itemset \in L_{k-1})$ do
(12) $C_k := C_k \cup \{itemset\}$;
(13) **end if**
(14) **end foreach**
(15) **return** C_k //trả về tập mục ứng viên có *k* phần tử
(16) **end**

Đặc trưng của thuật toán là quá trình khởi tạo ứng viên và xác định tập phổ biến *k* phần tử. Quá trình khởi tạo ứng viên của bước *k* sẽ sử dụng kết quả tập phổ biến của bước *k* – 1, trong bước *k* = 2 thuật toán sẽ thực hiện việc kiểm tra xem hai phần tử trong bộ này có tồn tại mối quan hệ *anct* hay *desc* không, nếu có thì loại bộ này. Chỉ cần xét bước *k* = 2, các bước còn lại không cần phải xét điều kiện đó, do các tập mục đều được khởi tạo từ tập *k* = 2. Nhược điểm phức tạp của thuật toán là phải duyệt CSDL theo từng tập mục ứng viên phát sinh. Nếu một CSDL lớn thì việc duyệt CSDL cho từng tập mục ứng viên là rất tốn kém, phức tạp.

3.3. Thuật toán luật kết hợp tổng quát mờ EFAR

Vào: D_F tập dữ liệu giao tác được làm mờ, ngưỡng *minsupp* α, *minconf* γ.

Ra: Tập các luật kết hợp mờ được khai phá.

Phương pháp:

(1) **begin**
(2) //Duyệt qua CSDL giao tác để xác định các mục phổ biến
(3) $T_{FIP} = \mathbf{null}$; //khởi tạo cây *FIP*
(4) $k = 1$;
(5) **foreach** $T_i \in D_F$ **do**
(6) **foreach** $item \in T_i$ **do**
(7) $C_1.item.Count \mathrel{+}= item.Count$; // giá trị thuộc [0, 1]
(8) //Giai đoạn chọn các mục phổ biến trong C_1 đưa vào cây *FIP*
(9) **foreach** $item \in C_1$ **do**
(10) **if** $(item.Count \geq \alpha)$ **then**
(11) $T_{FIP}.Add(item, k)$; //đưa mục vào cây *FIP* ở cấp thứ *k* = 1 theo thứ tự giảm dần *Count*
(12) $k = 2$;
(13) **while** $(T_{FIP}[k-1] \neq \emptyset)$ **do** //nếu cây *FIP* còn khả năng phát triển
(14) //phát sinh tập mục ứng viên ở cấp *k* cho cây *FIP*
(15) PhatSinh(T_{FIP}, k);
(16) //duyệt qua giao tác để xác định độ hỗ trợ của các tập mục vừa phát sinh trong *FIP* ở cấp thứ *k*
(17) **foreach** $T_i \in D_F$ **do**
(18) **foreach** $itemset \in T_{FIP}.L[k]$ **do**
(19) **if** $(itemset \in T_i)$ **do**
(20) min = MAX_FLOAT; //tìm min của các mục
(21) **foreach** $item \in itemset$ **do**
(22) **if** $(T_i.item.Count < min)$ **do**
(23) $\mathbf{min} = T_i.item.Count$;
(24) $T_{FIP}.L[k].itemset.Count \mathrel{+}= min$;
(25) **end if**
(26) **end foreach** itemset
(27) **end foreach** T_i
(28) //xác định loại bỏ những tập mục không đủ minsupp
(29) **foreach** $itemset \in T_{FIP}.L[k]$ **do**
(30) **if** $(itemset.Count \leq \alpha)$ **then**
(31) $T_{FIP}.L[k].remove(itemset)$; //xóa mục khỏi cây
(32) $k = k + 1$; //tăng số cấp của cây *FIP*
(33) **end while**
(34) $FI = \cup T_{FIP}.L[k]$; //*FI* chứa tất cả các tập mục phổ biến *k* mục $(k > 1)$
(35) //phát sinh các luật từ tập phổ biến
(36) **foreach** $itemset \in FI$ **do**
(37) **foreach** $item \in itemset$ **do**
(38) $supp = itemset.Count$;
(39) $conf = itemset.Count \,/\, item.Count$;
(40) **if** $(conf \geq \gamma)$ **then**
(41) *Xuất luật*: $itemset \setminus \{item\} \Rightarrow item, supp, conf$
(42) **end foreach** *item*
(43) **end foreach** *itemset*
(44) **end**

Hàm PhatSinh(T_{FIP}, k) phát sinh các nút ở cấp k:

```
(1)  begin
(2)      foreach item ∈ T_FIP.L[1] do //duyệt từ cấp thứ nhất
(3)          level = 1;
(4)          prefixItem = {∅}; //chứa các mục tiền tố
(5)          while (item.sub != null and level < k) do
(6)              prefixItem = prefixItem ∪ {item}; //thêm vào tiền tố
(7)              item = item.sub; //duyệt xuống dưới
(8)              level = level + 1; //tăng số cấp đang duyệt
(9)          end while
(10)     //kiểm tra nếu danh sách tiền tố có đủ số mục để kết hay không
(11)         if (prefixItem.Count < k − 2) continue; //qua mục khác
(12)     //duyệt qua từng nút ở cấp k – 1 để kết hợp thành tập mục có k phần tử
(13)         item_i = item;
(14)         while (item_i != null) do
(15)             item_j = item_i.Next; //item_j là mục kế tiếp của item_i
(16)             while (item_j != null) do //duyệt cho đến mục cuối
(17)                 if (k = 2 and Ancestor(item_i, item_j))
(18)                     continue;
(19)     //thêm nút mới có mục là item_j vào vị trí nút con bên dưới của item_i
(20)                 item_i.AddSub(item_j);
(21)                 item_j = item_j.Next; //qua item_j kế
(22)             end while item_j
(23)             item_i = item_i.Next; //qua item_i kế
(24)         end while item_i
(25)     end foreach item
(26) end
```

Thuật toán EFAR thì khắc phục được nhược điểm của thuật toán AFAR. Số lần duyệt qua CSDL trong thuật toán EFAR được xác định bằng số tập mục phổ biến k phần tử. Thuật toán này dựa vào cây tiền tố để thực hiện quá trình khai phá các tập mục phổ biến, nên việc khai phá được kết hợp với việc xây dựng cây tiền tố. Cây tiền tố đóng vai trò chính trong thuật toán EFAR, không chỉ là một cấu trúc dữ liệu lưu trữ hiệu quả mà còn góp phần rất nhiều vào việc phát sinh tập ứng viên phục vụ cho quá trình khai phá. Thuật toán EFAR chỉ duyệt CSDL theo từng tập mục được phát sinh. Vì số tập mục k phần tử phụ thuộc vào số chiều trong một giao tác nên số tập này không nhiều. Số lần duyệt giao tác dữ liệu trong thuật toán EFAR sẽ không đáng kể. Do đó dẫn đến thời gian thực hiện khai phá các mẫu phổ biến của thuật toán EFAR sẽ nhanh hơn thuật toán AFAR.

3.4. Đánh giá luật kết hợp tổng quát mờ sử dụng đại số gia tử

Trong phần này, bước đầu chúng tôi trình bày cách đánh giá luật kết hợp tổng quát mờ sử dụng đại số gia tử từ giai đoạn mở rộng cây phân lớp. Cách đánh giá này xem mỗi phần tử của ĐSGT là một vùng mờ. Do quá trình sinh vùng mờ dựa vào cấu trúc của ĐSGT nên việc đánh giá đơn giản, trực quan và hiệu quả hơn. Các bước thực hiện như sau:

Bước 1: Xem miền trị thuộc tính mờ là một ĐGST (giải sử ký hiệu Dom(*B*)). Chuyển đổi các giá trị

trong Dom(*B*)) về [0,1].

Bước 2: Với mỗi $x \in [0,1]$ sẽ tương ứng với mỗi phần tử *y* trong ĐGST (Sử dụng hàm ngược trong ĐSGT).

Bước 3: Dựa vào vùng mờ *y* để đánh giá luật kết hợp mờ tổng quát.

Ví dụ 3.1. Bảng giao tác minh họa dựa vào bảng 3.2 sau khi đã được mở rộng theo cây phân lớp.

Bảng 3.2. *Các giao tác được mở rộng theo cây phân lớp*

TID	Món hàng, Số lượng
1	(Bia, 3) (Mì, 4) (Áo Sơ mi, 2) *(Nước uống, 3) (Thực phẩm, 7)(Quần áo, 2)*
2	(Rượu, 3) (Mì, 7) (Áo khoác, 7) *(Nước uống, 3) (Thực phẩm, 10) (Quần áo, 7)*
3	(Rượu, 2) (Mì, 10) (Áo Sơ mi, 5) *(Nước uống, 2) (Thực phẩm, 10) (Quần áo, 5)*
4	(Mì, 10) (Áo Sơ mi, 10) *(Thực phẩm, 10) (Quần áo, 10)*
5	(Bia, 7) (Áo khoác, 10) *(Nước uống, 7) (Thực phẩm, 7) (Quần áo, 10)*
6	(Rượu, 2) (Mì, 10) (Áo khoác, 10) *(Nước uống, 2) (Thực phẩm,10)(Quần áo,10)*

Trước tiên, chúng tôi xem miền trị của thuộc tính mờ là một đại số gia tử và biến đổi các giá trị số lượng về giá trị trong [0,1] tương ứng, được xác định như sau:

$\underline{\boldsymbol{X}}_{\text{Soluong}} = (\boldsymbol{X}_{\text{Soluong}}, G_{\text{Soluong}}, H_{\text{Soluong}}, \leq)$, với G_{Soluong}= {*cao, thấp*}, H^{+}_{Soluong} = {*hơn, rất*}, H^{-}_{Soluong} ={*khả năng, ít*}, với *rất* > *hơn* và *ít* > *khả năng*, W_{Soluong}=0.6. Khi đó: *fm*(*thấp*) = 0.6, *fm*(*cao*) = 0.4, *fm*(*rất*) = 0.15, *fm*(*hơn*) = 0.25, *fm*(*khả năng*) = 0.25, *fm*(*ít*) = 0.35, chọn Dom(Soluong) = [0, 13]

Ta có *fm*(*rất thấp*) = 0.09, *fm*(*hơn thấp*) = 0.15, *fm*(*khả năng thấp*) = 0.15, *fm*(*ít thấp*) = 0.21. Vì *rất thấp* < *hơn thấp* < *thấp* < *khả năng thấp* < *ít thấp* nên *I*(*rất thấp*) = [0, 0.09], *I*(*hơn thấp*) = [0.09, 0.24], *I*(*khả năng thấp*) = [0.24, 0.39], *I*(*ít thấp*) = [0.39, 0.6]. Ta có *fm*(*rất cao*) = 0.06, *fm*(*hơn cao*) = 0.1, *fm*(*khả năng cao*) = 0.1, *fm*(*ít cao*) = 0.14. Vì *ít cao* < *khả năng cao* < *cao* < *hơn cao* < *rất cao* nên *I*(*ít cao*) = [0.6, 0.7], *I*(*khả năng cao*) = [0.7, 0.8], *I*(*hơn cao*) = [0.8, 0.9], *I*(*rất cao*) = [0.9, 1].

Ta có *Dom*(*Soluong*) = {2, 3, 4, 5, 7, 8, 9, 10}, bằng phương pháp chuyển đổi giá trị thuộc *Dom*(*Soluong*) thành giá trị thuộc [0,1]. Ta có *Dom*(*Soluong*) = {0.15, 0.23, 0.30, 0.38, 0.53, 0.61, 0.69, 0.76}. Vì [0.09, 0.24] = *I*(*hơn thấp*) nên 0.23 = *hơn thấp*, [0.39, 0.6] = *I*(*ít thấp*) nên 0.53 = *ít thấp*, [0.7, 0.8] =*I*(*khả năng cao*) nên 0.76 = *khả năng cao*. Do đó ta có bảng tập mục mờ hóa thuộc tính số lượng như sau:

Bảng 3.3. *Phân lớp mờ thuộc tính số lượng*

TID	Tập mục mờ
1	(0.23/Bia.*Hơn thấp*) (0,30/Mì.*Ít thấp*) (0.15/Áo Sơ mi.*Hơn thấp*) *(0.23/Nước uống.Hơn thấp) (0.53/Thực phẩm.Ít thấp)(0.15/Quần áo.Hơn thấp)*
2	(0.23/Rượu.*Hơn thấp*) (0.53/Mì.*Ít thấp*) (0.53/Áo khoác.*Ít thấp*) *(0.23/Nước uống.Hơn thấp) (0.76/Thực phẩm.Khả năng cao) (0.53/Quần áo.Ít thấp)*
3	(0.15/Rượu.*Hơn thấp*) (0.76/Mì.*Khả năng cao*) (0.38/Áo Sơ mi.*Ít thấp*) *(0.15/Nước uống.Hơn thấp) (0.76/Thực phẩm.Khả năng cao) (0.38/Quần áo.Ít thấp)*
4	(0.76/Mì.*Khả năng cao*) (0.76/Áo Sơ mi.*Khả năng cao*) *(0.76/Thực phẩm.Khả năng cao) (0.76/Quần áo.Khả năng cao)*
5	(0.53/Bia.*Ít thấp*) (0.76/Áo khoác.*Khả năng cao*) *(0.53/Nước uống.Ít thấp) (0.53/Thực phẩm.Ít thấp) (0.76/Quần áo.Khả năng cao)*
6	(0.15/Rượu.*Hơn thấp*) (0.76/Mì.*Khả năng cao*) (0.76/Áo khoác.*Khả năng cao*) *(0.15/Nước uống.Hơn thấp) (0.76/Thực phẩm.Khả năng cao)(0.76/Quần áo.Khả năng cao)*

Tiếp theo chúng tôi sẽ đếm vô hướng từng vùng mờ trong những giao tác, kết quả tính được gọi là số đếm của vùng mờ. Chẳng hạn, để tính số đếm của vùng mờ *Rượu*. *Hơn thấp* là (0 + 0.23 + 0.15 + 0 + 0 + 0.15) = 0.53 và kết quả như bảng sau:

Bảng 3.4. *Thống kê số đếm vùng mờ*

Vùng mờ	Số đếm	Vùng mờ	Số đếm
Bia.*Hơn thấp*	0.23	Áo khoác.*Hơn thấp*	0.0
Bia.*Ít thấp*	0.53	Áo khoác.*Ít thấp*	0.53
Bia.*Khả năng cao*	0.00	Áo khoác.*Khả năng cao*	1.52
Rượu.*Hơn thấp*	0.53	Nước uống.*Hơn thấp*	0.76
Rượu.*Ít thấp*	0.00	Nước uống.*Ít thấp*	0.53
Rượu.*Khả năng cao*	0.00	Nước uống.*Khả năng cao*	0.00
Mì.*Hơn thấp*	0.00	Thực phẩm.*Hơn thấp*	0.00
Mì.*Ít thấp*	0.83	Thực phẩm.*Ít thấp*	1.06
Mì.*Khả năng cao*	2.28	Thực phẩm.*Khả năng cao*	3.04
Áo Sơ mi.*Hơn thấp*	0.15	Quần áo.*Hơn thấp*	0.15
Áo Sơ mi.*Ít thấp*	0.38	Quần áo.*Ít thấp*	0.91
Áo Sơ mi.*Khả năng cao* năncaocacao	0.76	Quần áo.*Khả năng cao*	2.28

Chúng tôi sẽ chọn vùng mờ *có số đếm lớn nhất* cho từng *món hàng làm đại diện*. Với mỗi số đếm của bất kỳ vùng nào được chọn, sẽ kiểm tra lại với một ngưỡng *độ hỗ trợ nhỏ nhất*

Bảng 3.5. *Các vùng mờ được chọn khi lọc qua ngưỡng*

STT	Vùng mờ	Độ hỗ trợ
1	Thực phẩm.Khả năng cao	3.04
2	Quần áo.Khả năng cao	2.28
3	Mì.Khả năng cao	2.28
4	Áo khoác.Khả năng cao	1.52

Sau đó bảng giao tác chứa các vùng mờ có thể bỏ đi những vùng mờ không còn quan tâm và được rút gọn như sau:

Bảng 3.6. *Giao tác với vùng mờ rút gọn*

TID	Tập mục mờ
1	*(0.76/Thực phẩm.Khả năng cao)*
2	(0.76/Mì.Khả năng cao) *(0.76/Thực phẩm.Khả năng cao)*
3	(0.76/Mì.Khả năng cao) *(0.76/Thực phẩm.Khả năng cao) (0.76/Quần áo.Khả năng cao)*
4	(0.76/Áo khoác.Khả năng cao) *(0.76/Quần áo.Khả năng cao)*
5	(0.76/Mì.Khả năng cao) (0.76/Áo khoác.Khả năng cao) *(0.76/Thực phẩm.Khả năng cao)(0.76/Quần áo.Khả năng cao)*

Bước tiếp theo, chúng tôi sẽ tính số đếm mờ của bộ (*Thực phẩm.Khả năng cao* $\cap$ *Quần áo.Khả năng cao*) bằng cách duyệt qua từng giao tác (bảng 3.6) để lấy min ($\cap$) của từng giá trị mờ của hai vùng mờ và nhân tất cả các giá trị này lại, ta tính được số đếm mờ cho bộ trên như sau:

Bảng 3.7. *Số đếm mờ*

TID	Thực phẩm.Khả năng cao	Quần áo.Khả năng cao	Thực phẩm.Khả năng cao $\cap$ Quần áo.Khả năng cao
1	0.76	0	0
2	0.76	0	0
3	0.76	0.76	0.76
4	0	0.76	0
5	0.76	0.76	0.76
Count(Thực phẩm.Khả năng cao, Quần áo.Khả năng cao)			**0.58**

Tương tự như thế, chúng tôi được bảng có hai phần tử như sau:

Bảng 3.8. *Tập mục phổ biến có hai phần tử*

STT	Tập mục có 2 phần tử	Số đếm
1	(Thực phẩm.Khả năng cao, Quần áo.Khả năng cao)	0.58
2	(Thực phẩm.Khả năng cao, Áo khoác.Khả năng cao)	0.76
3	(Quần áo.Khả năng cao, Mì.Khả năng cao)	0.58
4	(Mì.Khả năng cao, Áo khoác.Khả năng cao)	0.76

Các luật được khởi tạo từ tập phổ biến này như sau:

Nếu *Thực phẩm = Khả năng cao* **thì** *Quần áo = Khả năng cao* (sup = 0.58)

Nếu *Thực phẩm = Khả năng cao* **thì** *Áo khoác = Khả năng cao* (sup = 0.76)

Nếu *Quần áo = Khả năng cao* **thì** *Mì = Khả năng cao* (sup = 0.58)

Nếu *Mì = Khả năng cao* **thì** *Áo khoác = Khả năng cao* (sup = 0.76)

Nếu *Quần áo =Khả năng cao* **thì** *Thực phẩm = Khả năng cao* (sup = 0.58)

Nếu *Áo khoác = Khả năng cao* **thì** *Thực phẩm = Khả năng cao* (sup = 0.76)

Nếu *Mì = Khả năng cao* **thì** *Quần áo = Khả năng cao* (sup = 0.58)

Nếu *Áo khoác = Khả năng cao* **thì** *Mì = Khả năng cao* (sup = 0.76)

Tính độ tin cậy của các luật bằng biểu thức sau:

$$Conf\,(A \Rightarrow B\,) = \frac{Count\,(A \cap B)}{Count\,(A)}$$

Ví dụ như tính độ tin cậy của luật (*Thực phẩm = Khả năng cao*) ⇒ (*Quần áo = Khả năng cao*) như sau: *Conf*(*Thực phẩm = Khả năng cao* ⇒ *Quần áo = Khả năng cao*) $= \frac{0.58}{3.04} = 0.19$

Bảng 3.9. *Độ tin cậy của các luật kết hợp*

STT	Luật kết hợp tổng quát mờ	Độ tin cậy
1	Thực phẩm = Khả năng cao ⇒ Quần áo = Khả năng cao	0.19
2	Thực phẩm = Khả năng cao ⇒ Áo khoác = Khả năng cao	0.25
3	Quần áo = Khả năng cao ⇒ Mì = Khả năng cao	0.25
4	Mì = Khả năng cao ⇒ Áo khoác = Khả năng cao	0.33
5	Quần áo = Khả năng cao ⇒ Thực phẩm = Khả năng cao	0.25
6	Áo khoác = Khả năng cao ⇒ Thực phẩm = Khả năng cao	0.5

7	Mì = Khả năng cao ⇒ Áo khoác = Khả năng cao	0.33
8	Áo khoác = Khả năng cao ⇒ Mì = Khả năng cao	0.5

Dựa vào độ tin cậy của các luật trong bảng 3.9 chúng ta có thể kết luận rằng hai luật ***Nếu*** *Áo khoác = Khả năng cao* ***thì*** *Thực phẩm = Khả năng cao và* **Nếu** *Áo khoác = Khả năng cao* **thì** *Mì = Khả năng cao* tốt hơn so với các luật còn lại.

4. Kết luận

Bài báo đã trình bày hai thuật toán khai phá luật kết hợp tổng quát mờ AFAR, EFAR để ứng dụng trong lĩnh vực khai phá dữ liệu và dữ liệu mờ. Từ những phân tích đặc điểm của 2 thuật toán trên, chúng tôi đã đề xuất phương pháp mới đánh giá luật kết hợp tổng quát mờ sử dụng đại số gia tử khá đơn giản và hiệu quả. Việc tối ưu hóa các tham số của hàm định lượng ngữ nghĩa trong đại số gia tử để nghiên cứu luật kết hợp mờ và xây dựng một ứng dụng trong thực tế sẽ được chúng tôi phát triển trong các bài báo sau.

TÀI LIỆU THAM KHẢO

[1]. Nguyễn Cát Hồ, Nguyễn Văn Long, *Làm đầy đủ đại số gia tử trên cơ sở bổ sung các phần tử giới hạn*, Tạp chí tin học và điều khiển học, (19), 1, (2003), 62-71.

[2]. Nguyễn Công Hào, *Một phương pháp xử lý giá trị khoảng trong cơ sở dữ liệu mờ*, Tạp chí Bưu chính Viễn thông và Công nghệ Thông tin. "Chuyên san các công trình nghiên cứu khoa học, nghiên cứu triển khai Công nghệ Thông tin – Truyền thông", Số 18, (2007), 68-74.

[3]. Hoàng Thị Lan Giao, *Bài giảng về Data Mining*, Trường Đại học Khoa học, Đại học Huế, 2010.

[4]. Fu, A. et al., *Finding fuzzy sets for the mining of fuzzy association rules for numerical attributes*, in Proceedings of 1st Intl. Symposium on Intelligent Data Engineering and Learning (IDEAL'98), (1998), 263-268.

[5]. R. Srikant and R. Agrawal, *Mining Quantitative Association Rules in Large Relational Tables*, Proc. of the ACM SIGMOD Conference on Management of Data, 1996.

[6]. R. Srikant, R. Agrawal, *Mining generalized association rules*, The Internat. Conf. on Very Large Databases, 1995.

[7]. Tzung-Pei Hong[a], Kuei-Ying Lin[b], Shyue-Liang Wang[b], *Fuzzy data mining for interesting generalized association rules*, Department of Electrical Engineering, National University of Kaohsiung, Der-Chung Road, Nan-Tzu District, Kaohsiung 811, Taiwan, ROC, 2002.

FUZZY ASSOCIATION RULE WITH HEDGE ALGEBRA BASED SEMANTICS

Nguyen Cong Hao[1], Nguyen Cong Doan[2]

[1]*Information Technology Center, Hue University*

[2]*Information Technology department , Go Dau, Tay Ninh*

Abstract. Fuzzy association rules have been researched by authors under several different approaches and results have been obtained[4-7]. However, fuzzy data mining with different data types to obtain respective fuzzy association rules is a difficult and complicated task. Thus, in this paper, we proposed a new method to processing fuzzy association rules with algebra based semantics.