MỤC LỤC

CÁC KÝ HIỆU VÀ CỤM TỪ VIẾT TẮT SỬ DỤNG TRONG LUẬN VĂN

Ký hiệu	Mô tả
$\mathscr{A}$	Hệ thông tin hay bảng quyết định
A, B	Tập các thuộc tính trong hệ thông tin
D	Tập thuộc tính quyết định trong hệ thông tin
a	Một thuộc tính điều kiện trong tập thuộc tính điều kiện của hệ thông tin
V_a	Tập giá trị của thuộc tính điều kiện
U	Tập đối tượng (tập tổng thể) trong hệ thông tin
RED	Tập rút gọn
$\varnothing$	Rỗng
$\subseteq$	Bị chứa trong
$\in$	Thuộc (là phần tử của)
$\geq$	Lớn hơn hoặc bằng
$\leq$	Nhỏ hơn hoặc bằng
$\neq$	Khác
$\cup, \cap$	Phép hợp, giao của một tập hợp
Viết tắt	**Mô tả**
CSDL	Cơ sở dữ liệu
KDD	Knowledge Discovery in Database
RS	Rough Set
GDT	Generalization Distribution Table
ILP	Inductive Logic Programming
GrC	Granular Computing

PHẦN MỞ ĐẦU

Lý thuyết tập thô do Z.Pawlak đề xuất vào đầu những năm 80 của thập kỉ XX đã được áp dụng ngày càng rộng rãi trong lĩnh vực khám phá tri thức trong các cơ sở dữ liệu. Trong những năm gần đây, lý thuyết tập thô được nhiều nhóm nghiên cứu hoạt động trong lĩnh vực tin học nói chung và khai phá tri thức từ cơ sở dữ liệu nói riêng nghiên cứu và áp dụng trong thực tế [1,4,6,9,10]. Lý thuyết tập thô được phát triển trên nền tảng cơ sở toán học vững chắc giúp cung cấp những công cụ hữu ích để giải quyết những bài toán phân lớp dữ liệu, phát hiện luật ... Những phương pháp dựa trên lý thuyết tập thô đặc biệt hữu ích đối với những bài toán với dữ liệu mơ hồ, không chắc chắn. Ngoài ra, lý thuyết tập thô cho phép trình diễn một mô hình hình thức về tri thức. Mô hình này được xác định như họ các mối quan hệ "không phân biệt được", nhờ đó tri thức được định nghĩa một cách rõ ràng theo nghĩa toán học và có thể được phân tích và xử lý bằng những công cụ toán học.

Trong lý thuyết tập thô, dữ liệu được biểu diễn thông qua hệ thông tin, hay bảng quyết định; ý tưởng chính trong việc phân tích dữ liệu theo tiếp cận tập thô xuất phát từ những khái niệm về sự xấp xỉ tập, về quan hệ "không phân biệt được". Từ những bảng dữ liệu lớn với dữ liệu dư thừa, không hoàn hảo, dữ liệu liên tục, hay dữ liệu biểu diễn dưới dạng ký hiệu, lý thuyết tập thô cho phép khai phá tri thức từ những loại dữ liệu như vậy nhằm phát hiện ra những quy luật tiềm ẩn từ khối dữ liệu này. Tri thức được biểu diễn dưới dạng các luật, mẫu mô tả mối quan hệ bị che dấu trong dữ liệu. Trong lý thuyết tập thô, chất lượng của thông tin được đo bằng cách sử dụng khái niệm tập xấp xỉ trên và xấp xỉ dưới. Nhằm thu hẹp nhiều nhất chính xác thông tin, ý tưởng “rút gọn” được sử dụng để cho phép loại bỏ những thông tin dư thừa, không cần thiết mà vẫn giữ được ý

nghĩa. Sau khi tìm được những quy luật chung nhất biểu diễn dữ liệu, người ta có thể tính toán độ mạnh, độ phụ thuộc giữa các thuộc tính trong hệ thông tin.

Theo Skowron và NingZong [9], cách tiếp cận lý thuyết tập thô để phân tích dữ liệu có rất nhiều lợi điểm quan trọng như:

- Cho phép xử lý hiệu quả bảng dữ liệu lớn, loại bỏ dữ liệu dư thừa, dữ liệu không hoàn hảo, dữ liệu liên tục,
- Hiệu quả trong việc tìm kiếm những mẫu tiềm ẩn trong dữ liệu,
- Sử dụng được tri thức kinh nghiệm,
- Nhận ra các mối quan hệ mà khi sử dụng các phương pháp thống kê khác không phát hiện được,
- Sử dụng quan hệ thứ lỗi trong quá trình phát hiện mẫu,
- Làm việc hiệu quả trên tập dữ liệu rút gọn,
- Cách giải thích rõ ràng và dễ hiểu.

Với những lợi điểm quan trọng trên của lý thuyết tập thô, chúng tôi đã giành thời gian để nghiên cứu và tìm hiểu về lý thuyết này. Ý tưởng "***Phát hiện luật theo tiếp cận tập thô***" được chọn làm đề tài nghiên cứu khoa học để làm luận văn thạc sĩ. Luận văn đi sâu tìm hiểu ý tưởng và cở sở toán học của lý thuyết tập thô, từ những hiểu biết về lý thuyết cũng như ứng dụng thực tế của tập thô trong lĩnh vực khai phá dữ liệu, chúng tôi đưa ra những nhận xét đối sánh giữa phát hiện luật theo tiếp cận tập thô và phát hiện luật kết hợp. Thông qua tìm hiểu và khai thác bộ công cụ ROSETTA (do Aleksander Øhrn và cộng sự thuộc nhóm nghiên cứu tri thức thuộc khoa Khoa học máy tính và thông tin của trường đại học Norwegian, Trondheim, Na-uy cùng nhóm Logic thuộc ĐHTH Warsaw, Ba-lan xây dựng), luận văn cũng đưa ra một số đề xuất ứng dụng thử nghiệm lý thuyết tập thô vào việc hỗ trợ quyết định bài toán xuất nhập cảnh tại sân bay Nội Bài.

Phương pháp nghiên cứu chủ yếu của luận văn là khảo sát, phân tích nội dung các bài báo khoa học về lý thuyết tập thô và ứng dụng được công bố vào những năm gần đây. Từ các kết quả nghiên cứu lý thuyết kết hợp với những vấn đề đặt ra trong bài toán thực tế, luận văn cũng đề xuất phương pháp thử nghiệm giải quyết vấn đề khám phá luật trong thực tế.

Luận văn được trình bày gồm có phần mở đầu, ba chương và phần kết luận. Trong chương một, chúng tôi tập trung chủ yếu vào giới thiệu tổng quan về quá trình khám phá tri thức theo tiếp cận tập thô. Các khái niệm cơ bản trong lý thuyết tập thô như: hệ thông tin, bảng quyết định, khái niệm không phân biệt được, tập xỉ trên tập xỉ dưới và miền biên ... được trình bày. Nội dung của chương này được tổng hợp từ các tài liệu [1,4,9,10].

Trong chương hai, luận văn tập trung giới thiệu về khám phá luật kết hợp theo cách tiếp cận thông thường và khám phá luật theo tiếp cận tập thô để từ đó đưa ra những nhận xét đối sánh về sự tương đồng hoặc khác biệt nhau trong các tính chất cơ bản của hai cách tiếp cận. Mục II.2.3 đưa ra mối liên hệ giữa mẫu và luật theo tiếp cận tập thô [5], dựa trên những mối quan hệ đó, chúng tôi đưa ra một số nhận xét đối sánh giữa khám phá luật kết hợp và khám phá luật theo tiếp cận tập thô. Kết quả đáng chú ý là mối tương đồng giữa độ mạnh trong luật theo tiếp cận tập thô và độ hỗ trợ của luật kết hợp.

Trong chương ba, luận văn đưa ra một số mô hình ứng dụng của mẫu được phát hiện từ dữ liệu theo tiếp cận tập thô [5]. Từ kết quả nghiên cứu trình bày trong chương một và chương hai, thông qua công cụ ROSETTA, chúng tôi đề xuất việc ứng dụng luật kết hợp theo tiếp cận tập thô vào thực tế trong bài toán quản lý thông tin khách xuất nhập cảnh tại cửa khẩu và nhận được một số luật tương đối hợp lý.

Luận văn được thực hiện dưới sự hướng dẫn của Tiến sĩ Hà Quang Thuỵ - Bộ môn Các Hệ thống Thông tin, Khoa Công nghệ. Em xin bày tỏ lòng biết ơn sâu sắc tới Thầy đã hướng dẫn và có ý kiến chỉ dẫn quý báu trong quá trình em làm luận văn. Em xin chân thành cảm ơn PGS. Nguyễn Quốc Toản, PGS. TS. Hồ Thuần đã cho nhiều ý kiến quý báu để bản luận văn được hoàn thiện hơn. Em xin cảm ơn các thầy giáo trong bộ môn Các Hệ thống Thông tin, nhóm seminar "Data mining và KDD". Em cũng xin cảm ơn các thầy cô giáo trong Khoa, cán bộ thuộc phòng Khoa học và Đào tạo sau Đại học, Khoa Công nghệ đã tạo điều kiện trong quá trình học tập và nghiên cứu tại Khoa. Cuối cùng xin bày tỏ lòng cảm ơn tới những người thân trong gia đình, bạn bè đã động viên và giúp đỡ để tôi hoàn thành bản luận văn này.

CHƯƠNG 1. TỔNG QUAN VỀ KHÁM PHÁ TRI THỨC THEO TIẾP CẬN TẬP THÔ

I.1. HỆ THÔNG TIN VÀ TẬP THÔ

I.1.1. Một số khái niệm

I.1.1.1. Khái niệm về hệ thông tin

Trong hoạt động hàng ngày, đặc biệt khi thu thập dữ liệu vào các kho dữ liệu (datawarehousing), ta thường gặp các tập hợp dữ liệu được miêu tả bởi một bảng, trong đó hàng biểu diễn "bản ghi" (một phần tử, một trường hợp, một sự kiện hay đơn giản là biểu diễn một đối tượng), còn các cột biểu diễn một thuộc tính (một biến, một quan sát, một tính chất ...). Từ những năm đầu của thập kỷ 1980, Pawlak hình thức hóa bảng kiểu này thành khái niệm ***hệ thông tin*** (information system) [1,5, 9, 10].

Định nghĩa 1.1. Hệ thông tin là cặp $\mathcal{A}$ = (U,A) trong đó U là một tập hữu hạn khác rỗng các ***đối tượng*** và A là một tập hữu hạn khác rỗng các ***thuộc tính***, trong đó a: U $\rightarrow$ V_a với mọi $a \in$ A. Tập V_a được gọi là ***tập giá trị*** của a.

- **Ví dụ**: Có một hệ thông tin thể hiện như trong bảng 1. Có 7 đối tượng (Mỗi đối tượng ở đây là một khách Xuất Nhập Cảnh) và 3 thuộc tính: *Tới nước, Nơi sinh, Tôn giáo*.

	Tới nước	*Nơi sinh*	*Tôn giáo*
x_1	Mỹ	Hà nội	Có
x_2	Mỹ	Hải phòng	Có
x_3	Pháp	Sài gòn	Không
x_4	Pháp	Sài gòn	Không
x_5	Đức	Đà nẵng	Có
x_6	Mỹ	Đà nẵng	Không
x_7	Pháp	Đà nẵng	Không

Bảng 1. Một ví dụ về hệ thông tin

Chúng ta nhận thấy trường hợp các đối tượng khác nhau x_3 và x_4, lại có các giá trị thuộc tính giống nhau: đây là trường hợp ***không phân biệt được*** các đối tượng nếu chỉ sử dụng thông tin từ các thuộc tính đã cho. Tính không phân biệt được là một trong những yếu tố của sự mập mờ. Có thể nhận thấy tính mập mờ từ việc không phân biệt được: nếu chỉ xem xét các thuộc tính trên đây thì hai đối tượng x_3 và x_4 là hoàn toàn giống nhau, tuy nhiên như sau này chúng ta thấy, x_3 khi xuất cảnh cần phải xem xét trong khi đó với x_4 thì không cần làm điều đó.

I.1.1.2. Khái niệm bảng quyết định

Trong nhiều ứng dụng, người ta đã biết nội dung kết quả của việc phân lớp là quyết định phân lớp. Tri thức (chỉ dẫn quyết định) phân lớp được thể hiện bằng một thuộc tính riêng biệt được gọi là ***thuộc tính quyết định*** trong hệ thông tin. Trong trường hợp đó, hệ thông tin được gọi là *hệ quyết định* [1,5,9,10].

Định nghĩa 1.2. Bảng (hệ) quyết định là hệ thông tin bất kỳ có dạng $\mathcal{A}$ = (U, A$\cup${d}) (hay $\mathcal{A}$ = (U, A,{d})), với $d \notin$ A là *thuộc tính quyết định*. Các thuộc tính thuộc A được gọi là *thuộc tính điều kiện* hay *điều kiện*.

Thuộc tính quyết định có thể có nhiều hơn hai giá trị, tuy nhiên thông dụng là kiểu giá trị nhị phân. Quá trình khám phá ra mối quan hệ giữa thuộc tính quyết định theo thuộc tính điều kiện trong bảng quyết định thuộc vào loại ***học máy có hướng dẫn***, trong đó thể hiện diển hình nhất là "học qua ví dụ".

U	*Tới nước*	*Nơi sinh*	*Tôn giáo*	*Xem xét*
x_1	Mỹ	Hà nội	Có	Cấm
x_2	Mỹ	Hải phòng	Có	Không
x_3	Pháp	Sài gòn	Không	Không
x_4	Pháp	Sài gòn	Không	Cấm
x_5	Đức	Đà nẵng	Có	Không
x_6	Mỹ	Đà nẵng	Không	Cấm
x_7	Pháp	Đà nẵng	Không	Không

Bảng 2. *CXN* - Một bảng quyết định

Ví dụ. Bảng 2 mô tả một bảng quyết định bao gồm 7 đối tượng (trường hợp), một thuộc tính quyết định là *Xem xét* và 3 thuộc tính *Tới nước, Nơi sinh, Tôn giáo*.

Chúng ta tiếp tục quan sát trường hợp cặp hai đối tượng là x_3 và x_4 vẫn là cặp có các giá trị giống nhau theo thuộc tính điều kiện, nhưng kết quả quyết định đối với hai đối tượng là khác nhau.

Như vậy một tri thức được tổng hợp từ bảng quyết định trên đây sẽ là luật có dạng "Nếu có *Tới nước* là Mỹ, *Nơi sinh* là Hà nội và có tôn giáo thì *Xem xét* là Cấm" tức là Nếu một khách Xuất Nhập Cảnh xuất cảnh đến Mỹ, *Nơi sinh* là Hà nội và có tôn giáo thì sẽ bị cấm Xuất Nhập cảnh Việt Nam. Trong những thuộc tính có thể của tập các luật được xây dựng, sự cực tiểu hoá (*minimality*- độ dài giả thiết của luật là cực tiểu) là một trong những vấn đề quan trọng [5].

Chú ý. Tổng quát, có thể có nhiều thuộc tính quyết định và khi đó bảng quyết định có dạng $\mathcal{A}$ = (U, Con∪Dec), với Con là tập các *thuộc tính điều kiện* hay *điều kiện* còn Dec là tập các *thuộc tính quyết định* (trong đó Con∩Dec = ∅) [1].

I.1.1.3. Quan hệ không phân biệt được trong hệ thông tin

Một trong những cơ sở toán học của lý thuyết tập thô là quan hệ không phân biệt được (một quan hệ tương đương) trong hệ thông tin.

Cho U là tập các đối tượng, một quan hệ nhị phân R ⊆ U × U trên U được gọi là:

- *Phản xạ* nếu mọi đối tượng đều có quan hệ với chính nó *xRx*,
- *Đối xứng* nếu *xRy* thì *yRx,*
- *Bắc cầu* nếu *xRy* và *yRz* thì *xRz*

Một quan hệ R có cả ba tính chất phản xạ, đối xứng và bắc cầu được gọi là một *quan hệ tương đương*. Quan hệ tương đương R sẽ chia (phân hoạch) tập tổng thể U thành các *lớp tương đương*. Lớp tương đương của phần tử $x \in$ U, kí hiệu là [x], chứa tất cả các đối tượng $y \in$ U mà *xRy*.

Như đã được đề cập trong phần trước, lý thuyết tập thô quan tâm đến quan hệ không phân biệt được [5, 9, 10]. Cho hệ thông tin $\mathcal{A}$ = (U, A), quan hệ không phân biệt được được trình bày như dưới đây.

Định nghĩa 1.3. Với tập con bất kỳ B $\subseteq$ A, tồn tại một quan hệ tương đương (kí hiệu là $IND_{\mathcal{A}}$(B)) được xác định như sau:

$$IND_{\mathcal{A}}(B)=\{(x,x') \in U^2 \mid \forall a \in B:\ a(x) = a(x')\}$$

$IND_{\mathcal{A}}$(B) được gọi là *quan hệ không phân biệt được* theo nghĩa nếu như hai đối tượng x, x' mà (x,x') $\in IND_{\mathcal{A}}$(B) thì x và x' là không phân biệt được lẫn nhau bởi các thuộc tính trong B.

Tính chất tương đương của $IND_{\mathcal{A}}$(B) là dễ dàng kiểm tra theo định nghĩa. Trong nhiều trường hợp khi hệ thông tin đã hoàn toàn xác định, ta dùng cách viết *IND(B)* hay *IND* thay cho cách viết $IND_{\mathcal{A}}$(B) và cũng dùng cách nói là *tính không phân biệt được theo B.*

Lớp tương đương theo quan hệ không phân biệt được B được biểu diến là $[x]_B$. Ký tự $\mathcal{A}$ trong quan hệ không phân biệt được thường bị bỏ qua nếu nó đã rõ ràng trong hệ thông tin.

- **Ví dụ.** Xét **bảng 2** minh hoạ cho một quan hệ không phân biệt được. Nếu không xem xét thuộc tính tôn giáo thì các tập con khác rỗng của các thuộc tính điều kiện là {*Tới nước*}, {*Nơi sinh*} và {*Tới nước*, *Nơi sinh*}. Xem xét thuộc tính {*Tới nước*}, các đối tượng x_3 và x_4 thuộc vào cùng một lớp tương đương và không có khả năng phân biệt được. Ba quan hệ *IND* xác định phân hoạch thành từng phần tập tổng thể.

  $IND(\{Tới\ nước\}) = \{\{x_1,x_2,x_6\},\{x_3,x_4,x_7\},\{x_5\}\}$

  $IND(\{Nơi\ sinh\}) = \{\{x_1\},\{x_2\},\{x_3,x_4\},\{x_5,x_6,x_7\}\}$

  $IND(\{Tới\ nước, Nơi\ sinh\}) = \{\{x_1\},\{x_2\},\{x_3,x_4\},\{x_5\},\{x_6\},\{x_7\}\}$

I.1.1.4. Tập mô tả được và ngôn ngữ mô tả tập

Z. Pawlak đã đưa ra khái niệm tập mô tả được [1] trong hệ thông tin $\mathcal{A}$ = (U, A). Xét R là quan hệ không phân biệt được với trường hợp đặc biệt khi B = A gồm tất cả các thuộc tính. Lớp tương đương theo quan hệ R được gọi là tập sơ cấp [1,9] và gọi E là tập hợp các tập sơ cấp. Tương ứng với quan hệ R, Pawlak đưa ra khái niệm hạng thức (term) trong ngôn ngữ L dùng để mô tả các tập trong hệ thông tin [1]. Ngôn ngữ L bao gồm hai nội dung: hạng thức (term) trong ngôn ngữ đó và ngữ nghĩa của một hạng thức được xác định như dưới đây.

Định nghĩa 1.4. [1, 9] Hạng thức thuộc L được định nghĩa đệ quy như sau:

(1) 0 và 1 là các hạng thức (hạng thức hằng),

(2) Nếu $a \in A$ và $v \in V_a$ thì (a,v) là một hạng thức,

(3) Nếu t, t_1, t_2 là các hạng thức thì $\bar{t}$, $t_1 \vee t_2$, $t_1 \wedge t_1$ cũng là các hạng thức.

Định nghĩa 1.5. [1, 9] Hạng thức t có ngữ nghĩa σ (t) thông qua ánh xạ σ từ L vào 2^U (tập các tập con của U) được xác định như sau:

(1) $\sigma(0) = \varnothing$ và $\sigma(1) = U$

(2) $\sigma((a,v)) = \{ x \in U : a(x)=v\}$

(3) $\sigma(\bar{t}) = U - \sigma(t)$; $\sigma(t_1 \vee t_2) = \sigma(t_1) \cup \sigma(t_2)$; $\sigma(t_1 \wedge t_2) = \sigma(t_1) \cap \sigma(t_2)$

Hạng thức dạng $t = \bigwedge_{a \in A}(a,v_a)$ được gọi là hạng thức *dạng chuẩn*. Tồn tại các hạng thức dạng chuẩn nhưng có ngữ nghĩa rỗng. Gọi L_{NF} là tập hợp các hạng thức dạng chuẩn có ngữ nghĩa khác rỗng. Các kết quả sau đây đã được khẳng định trong [1].

Mệnh đề 1.1. Tồn tại sự tương ứng 1-1 giữa tập E các tập sơ cấp với tập các hạng thức dạng chuẩn có ngữ nghĩa khác rỗng L_{NF} theo nghĩa dưới đây:

(1) Với bất kỳ $e \in E$, tồn tại duy nhất hạng thức $t \in L_{NF}$ sao cho $\sigma(t) = e$,

(2) Với bất kỳ hạng thức t trong L_{NF} thì e =σ (t) là tập sơ cấp.

Thông qua hệ thông tin và ngôn ngữ L chúng ta có thể "mô tả" được các tập con các đối tượng. Pawlak đã đưa ra khái niệm về tập mô tả được trong hệ thông tin như định nghĩa dưới đây.

Định nghĩa 1.6. Một tập con X khác rỗng các đối tượng được gọi là tập mô tả được khi và chỉ khi X là hợp của các tập sơ cấp trong hệ thông tin (Trường hợp đặc biệt là tập rỗng cũng được coi là một tập mô tả được).

Mệnh đề dưới đây là kết quả suy suy diễn từ mệnh đề 1.1. và định nghĩa 1.6.

Mệnh đề 1.2. Tập X là mô tả được khi và chỉ khi tồn tại một hạng thức t trong L để cho σ(t) = X.

Mệnh đề 1.2 cho thấy ý nghĩa của khái niệm "mô tả được" của tập X là chúng ta có thể dùng một hạng thức trong ngôn ngữ L để "mô tả" tập X đó.

Theo các định nghĩa và mệnh đề trên đây thì không phải tập con nào của U cũng là tập mô tả được, có nghĩa là tồn tại các tập con các đối tượng không là tập mô tả được. Khái niệm tập thô được Pawlak đề xuất được dùng để chỉ dẫn đến các tập như thế và đã mở ra một mô hình ứng dụng rất rộng rãi trong lĩnh vực khai phá dữ liệu và khám phá tri thức trong cơ sở dữ liệu [1,4,5,9,10].

I.1.2. Tập thô trong không gian xấp xỉ

I.1.2.1. Tập xấp xỉ trên, xấp xỉ dưới và miền biên

Một quan hệ tương đương cho một cách phân hoạch tập các đối tượng (tập tổng thể), trong đó mỗi lớp tương đương được gọi là một tập sơ cấp và theo định nghĩa 1.6, chúng ta có các tập mô tả được. Vấn đề đặt ra là hãy tìm phương pháp sử dụng phân hoạch đã cho từ một quan hệ tương đương để "mô tả" các tập con đối tượng mà không phải là tập mô tả được.

Đối sánh với bảng quyết định, chúng ta chú ý tới quan hệ không phân biệt được $IND_{\mathcal{A}}$(B) tương ứng với tập các thuộc tính điều kiện B (B $\subseteq$ A), quan hệ này phân hoạch tập đối tượng thành các lớp tương đương $[x]_B$. Gọi X là tập concác đối tượng có cùng giá trị tại thuộc tính quyết định ***d***. Trong nhiều trường hợp, tập X như vậy không là mô tả được bởi vì tồn tại các lớp tương đương $[x]_B$ bao gồm cả các phần tử thuộc X và cả các phần tử không thuộc X.

Ví dụ, cho bảng quyết định trong **bảng 2** và lấy tập B là tập các thuộc tính điều kiện, tập X bao gồm các đối tượng cần xem xét khi cho xuất, nhập cảnh. Xét lớp tương đương đương chứa hai đối tượng x_3 và x_4, chúng có cùng giá trị trên tập thuộc tính điều kiện nhưng giá trị trên thuộc tính quyết định lại khác nhau, có nghĩa là tập X đang xét không phải là tập mô tả được.

Trong định nghĩa 1.6 về tập mô tả được chúng ta xem xét tập X với các lớp tương đương sinh ra do quan hệ $IND_{\mathcal{A}}$(B). Phát triển việc đối sánh đó, ý tưởng về tập thô đã được nảy sinh. Tuy rằng, chúng ta không thể xác định tính chất để mô tả tập X (những khách cần xem xét khi Xuất Nhập Cảnh) một cách chính xác và rõ ràng (không mô tả được tập này), nhưng lại có thể "mô tả" được tập các khách ***chắc chắn*** cần phải xem xét (tập {x1, x6}) hoặc tập các khách Xuất Nhập Cảnh ***có khả năng*** cần phải xem xét (tập {x1, x3, x4, x6}) và cuối cùng là tập các khách Xuất Nhập Cảnh thuộc vùng ***ranh giới*** giữa các trường hợp chắc chắn và khả năng (tập {x3, x4}). Nếu vùng biên này không rỗng thì tập này được gọi là tập thô. Hình thức hóa ý tưởng này được diễn tả như dưới đây.

Định nghĩa 1.7. Giả sử $\mathcal{A}$ = (U, A) là một hệ thông tin và B $\subseteq$ A và X $\subseteq$ U. Các tập xấp xỉ của X theo thông tin có từ B, được xác định như dưới đây:

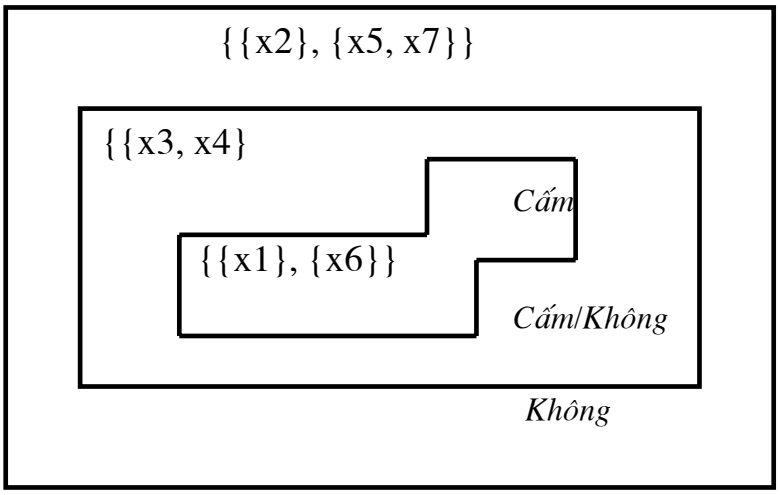
(1) Tập B-*xấp xỉ dưới* của X, kí hiệu là $\underline{B}X$, là tập $\underline{B}X = \{x \mid [x]_B \subseteq X\}$

(2) Tập B-*xấp xỉ trên* của X, kí hiệu là $\overline{B}X$, là tập $\overline{B}X = \{x \mid [x]_B \cap X \neq \varnothing\}$.

Đối tượng trong $\underline{B}X$ chắc chắn được phân lớp là thành viên của X theo tri thức cơ sở từ B (tập $\underline{B}X$ có thể được gọi là ***tập chắc chắn***), trong khi đối tượng trong $\overline{B}X$ chỉ có khả năng được phân lớp là thành viên của X theo tri thức cơ sở trong B (tập $\overline{B}X$ có thể được gọi là ***tập khả năng***). Tập $BN_B(X) = \overline{B}X - \underline{B}X$ được gọi là B-*vùng biên* của X, do vậy chúng ta không thể phân loại (và cũng không thể loại bỏ) các đối tượng trong tập đó vào trong X trên tri thức cơ sở trong B. Tập U - $\overline{B}X$ được gọi là B-*vùng ngoài* của X bao gồm các đối tượng chắc chắn không thuộc X (trên tri thức cơ sở có được từ B[1]). Một tập được gọi là *thô* hoàn toàn nếu vùng biên của nó là không rỗng.

a) Ví dụ

Trường hợp chung nhất là để tổng hợp xác định kết quả (hay lớp quyết định) trong các thuộc tính điều kiện. Giả sử W={*x* | *Xem xét*(*x*) = *Cấm*} như ví dụ minh



Hình 1. Xấp xỉ tập khách cần xem xét khi Xuất Nhập Cảnh, sử dụng 2 thuộc tính điều kiện *Tới nước* và *Nơi sinh*.

[1] Ký tự B được xem là tập con B của các thuộc tính trong A. Nếu một tập con khác được chọn ví dụ như F $\subseteq$ A thì cũng có các khái niệm như: F-*vùng biên*, F-*xấp xỉ* trên và F-*xấp xỉ dưới*.

hoạ trên **bảng 2**. Ta thu được vùng xấp xỉ dưới $\underline{A}W = \{x_1,x_6\}$, xấp xỉ trên $\overline{A}W = \{x_1,x_3,x_4,x_6\}$, vùng biên $BN_A(W)=\{ x_3,x_4\}$ và vùng biên ngoài U - $\overline{A}W = \{x_2,x_5,x_7\}$.

Do đó mà tập kết quả *Xem xét* là thô vì vùng biên là không rỗng.

b) Các tính chất của sự xấp xỉ.

Trong [9, 10] đã trình bày các tính chất sau đây về tập xấp xỉ:

(1) $\underline{B}X \subseteq X \subseteq \overline{B}X$,

(2) $\underline{B}(\varnothing) = \overline{B}(\varnothing)$, $\underline{B}(U) = \overline{B}(U) = U$,

(3) $\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$,

(4) $\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$,

(5) Nếu $X \subseteq Y$ thì $\underline{B}(X) \subseteq \underline{B}(Y)$ và $\overline{B}(X) \subseteq \overline{B}(Y)$,

(6) $\underline{B}(X \cup Y) \supseteq \underline{B}(X) \cup \underline{B}(Y)$,

(7) $\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$,

(8) $\underline{B}(-X) = -\overline{B}(X)$,

(9) $\overline{B}(-X) = -\underline{B}(X)$,

(10) $\underline{B}(\underline{B}(X)) = \overline{B}(\underline{B}(X)) = \underline{B}(X)$,

(11) $\overline{B}(\overline{B}(X)) = \underline{B}(\overline{B}(X)) = \overline{B}(X)$,

Trong đó ký hiệu -X biểu thị cho U-X.

Có thể nhận thấy là tập xấp xỉ trên và xấp xỉ dưới của một tập có vẻ ngoài tương đồng với *phần trong* và *bao đóng* của tập hợp trong tôpô hình học được sinh ra bởi quan hệ không phân biệt được.

c) Bốn loại tập thô cơ bản

Người ta phân tập thô thành 4 loại [9]:

- *X xác định thô thực sự theo B nếu* $\underline{B}X \neq \varnothing$ *và* $\overline{B}X \neq U$,

- *X là không xác định bên trong theo B nếu* $\underline{B}X = \varnothing$ *và* $\overline{B}X \neq U$,
- *X là không xác định bên ngoài theo B nếu* $\underline{B}X \neq \varnothing$ *và* $\overline{B}X = U$,
- *X là không xác định thực sự theo B nếu* $\underline{B}X = \varnothing$ *và* $\overline{B}X = U$.

Giải thích bằng trực giác thì sự phân lớp này có nghĩa như sau:

- Nếu X xác định thô thực sự theo B nghĩa là chúng ta có thể quyết định rằng một số thành phần của U mà chúng thuộc X và cho một số phần tử của U mà chúng thuộc -X, sử dụng B.
- Nếu X là không xác định nội tại bên trong theo B có nghĩa là chúng ta có thể quyết định rằng một số phần tử của U mà chúng thuộc -X nhưng không thể quyết định cho bất kỳ phần tử của U nào có thuộc X không, sử dụng B.
- Nếu X là không xác định bên ngoài theo B có nghĩa là chúng ta có thể quyết định rằng một số phần tử của U mà chúng thuộc X nhưng không thể quyết định cho bất kỳ phần tử của U nào có thuộc X không, sử dụng B.
- Nếu X là không xác định thực sự theo B có nghĩa là chúng ta quyết định rằng bất kỳ phần tử của U có thuộc X hay -X không, sử dụng B.

d) Độ đo liên quan biên xấp xỉ

Tập thô được chỉ số hoá bởi hệ số sau:

$$\alpha_B(X) = \frac{\left|\underline{B}(X)\right|}{\left|\overline{B}(X)\right|},$$

$\alpha_B(X)$ được gọi là *độ đo liên quan biên xấp xỉ của X*, với $|X|$ biểu diễn lực lượng của X $\neq \varnothing$. Có thể thấy được $0 \leq \alpha_B(X) \leq 1$. Nếu $\alpha_B(X)=1$ thì X *đúng hoàn toàn* đối với B, ngược lại nếu $\alpha_B(X)<1$ thì X là *thô* đối với B.

I.1.2.2. Hàm thô và một số độ đo phụ thuộc có liên quan

Trong lý thuyết tập hợp cổ điển, mỗi thành viên thuộc một tập hợp hoặc không. Hàm thành viên (hàm thuộc) là hàm đặc trưng của tập hợp nhận một trong hai giá trị 0 và 1. Trong tập thô, ý tưởng của hàm thành viên thì khác. Hàm thành viên thô xác định mức độ giao nhau liên quan giữa tập X và lớp tương đương $[x]_B$ chứa x, nó được định nghĩa như sau:

$$\mu_X^B : U \to [0,1] \text{ và được xác định } \mu_X^B(x) = \frac{|[x]_B \cap X|}{|[x]_B|}$$

Hàm thô có thể được hiểu như một sự ước lượng tần số cơ bản của Pr($x \in X \mid x$, B) (xác xuất điều kiện mà đối tượng x thuộc tập X), với lớp tương đương IND(B). Các công thức cho tập xấp xỉ trên và xấp xỉ dưới có thể được suy ra từ hàm thô với mức chính xác tuỳ ý $\pi \in \left(\frac{1}{2},1\right]$ [10] như sau:

$$\underline{B}_\pi X = \left\{x \middle| \mu_X^B(x) \geq \pi\right\}$$

$$\overline{B}_\pi X = \left\{x \middle| \mu_X^B(x) \rangle 1-\pi\right\}$$

Trường hợp đặc biệt $\pi = 1.0$

Các khái niệm về sự xấp xỉ được xây dựng dựa trên tri thức nền cơ bản. Có thể thấy rằng các khái niệm này liên quan đến các đối tượng (ẩn) không nhìn thấy. Do đó nó rất hữu ích để xác định sự xấp xỉ biểu hiện bằng tham số với các tham số phù hợp trong quá trình tìm kiếm cho các khái niệm từ sự xấp xỉ tập. Ý tưởng này là chủ đạo cho việc xây dựng các khái niệm về sự xấp xỉ sử dụng phương pháp tập thô.

I.2. KHÁM PHÁ TRI THỨC THEO TIẾP CẬN TẬP THÔ

I.2.1. Tính phụ thuộc thuộc tính trong hệ thông tin

I.2.1.1. Tính phụ thuộc thuộc tính

Trong quá trình phân tích dữ liệu, một vấn đề quan trọng cần quan tâm đó là khám phá sự phụ thuộc giữa các thuộc tính trong hệ thông tin [1, 4, 9]. Tập các thuộc tính D phụ thuộc hoàn toàn vào tập các thuộc tính C biểu thị là $C \Rightarrow D$, nếu tất cả các giá trị thuộc tính từ D được xác định duy nhất bởi các giá trị thuộc tính trong C. Nói cách khác D phụ thuộc hoàn toàn vào C, nếu tồn tại phụ thuộc hàm giữa các giá trị của D và C.

Sự phụ thuộc có thể được định nghĩa như sau: Giả sử D và C là các tập con của A. Ta nói rằng D phụ thuộc vào C với *mức k* ($0 \le k \le 1$) biểu thị là $C \Rightarrow_k D$ nếu:

$$k = \gamma(C, D) = \frac{|POS_C(D)|}{|U|}, \text{với } POS_C(D) = \bigcup_{X \in U/D} \underline{C}(X),$$

được gọi là một *C-vùng khẳng định* của phân hoạch U/D đối với C, là tập tất cả các phần tử của U mà có thể được phân loại duy nhất thành khối của phân hoạch U/D với ý nghĩa của C.

$$\gamma(C, D) = \sum_{X \in U/D} \frac{|\underline{C}(X)|}{|U|}.$$

Nếu $k = 1$ ta nói rằng D phụ thuộc hoàn toàn vào C, và nếu $k<1$ ta nói rằng D phụ thuộc một phần vào C.

Hệ số k diễn tả tỉ lệ của các thành phần trong tập tổng thể, với sự phân loại thành khối của phân hoặc U/D, các thuộc tính sử dụng trong C gọi là *mức phụ thuộc*.

Dễ nhận ra rằng nếu D phụ thuộc hoàn toàn vào C thì $IND(C) \subseteq IND(D)$. Điều này có nghĩa là phân hoạch được sinh ra bởi C tốt hơn phân hoạch được sinh ra bởi D.

Tóm lại: D là phụ thuộc hoàn toàn (hay một phần) vào C nếu tất cả (một số) phần tử của tập tổng thể có thể được phân loại duy nhất thành khối của phân hoạch U/D, sử dụng C.

I.2.1.2. Tập thuộc tính rút gọn và tập thuộc tính nhân

Một hệ thông tin (ví dụ với một bảng quyết định) có thể không lớn nhưng rất có thể nó bị dư thừa thông tin ít nhất trong 2 trường hợp sau:

- Các đối tượng giống nhau hoặc không phân biệt được có thể xuất hiện nhiều lần trong bảng.
- Một số thuộc tính có thể là dư thừa.

Trong mục I.1.1.3, luận văn có đề cập đến xu hướng tự nhiên của việc giảm bớt dữ liệu bằng cách nhận biết các lớp tương đương, ví dụ như các đối tượng không có khả năng phân biệt sử dụng các thuộc tính có sẵn. Việc ghi lại dữ liệu sẽ được thực hiện chỉ từ một thành phần của lớp tương đương là cần thiết để miêu tả toàn bộ lớp. Một xu hướng khác trong việc rút gọn dữ liệu là chỉ giữ lại những thuộc tính mà bảo toàn quan hệ không phân biệt được và tập xấp xỉ. Những thuộc tính còn lại mà khi vứt bỏ chúng đi không ảnh hưởng đến sự phân lớp, đó là những thuộc tính dư thừa. Còn lại các tập con các thuộc tính và chúng là tối thiểu gọi là các tập rút gọn. Việc tính toán các lớp tương đương là không khó. Số tập rút gọn của hệ thông tin với m thuộc tính có thể bằng $\left(\frac{m}{\lfloor m/2 \rfloor}\right)$ [4]. Có nghĩa là việc tính toán tập rút gọn là không đơn giản, nó không thể tính toán nhanh được bằng máy tính. Thực tế nó là một trong những vấn đề khó giải quyết trong

phương pháp luận lý thuyết tập thô. Tuy nhiên, tồn tại một số phương pháp kinh nhgiệm tốt để tính toán, ví dụ như dựa trên thuật toán di truyền tính toán tập rút gọn có hiệu quả trong thời gian chấp nhận được, trừ khi số các thuộc tính là quá lớn.

Xem xét các thuộc tính có thể rút gọn được và không thể rút gọn được trong bảng quyết định.

Giả sử với bảng quyết định $\mathcal{A}$ = (U, A∪D) với thuộc tính $a \in A$ tập các thuộc tính điều kiện, U là tập tổng thể và D thuộc tính quyết định. Thuộc tính *a* có thể rút gọn được trong $\mathcal{A}$ nếu: $POS_A(D) = POS_{(A-\{a\})}(D)$, các trường hợp còn lại thì không thể rút gọn thuộc tính a trong $\mathcal{A}$.

$\mathcal{A}$ = (U, A∪ D) là rút gọn được nếu tồn tại các thuộc tính $a \in A$ là rút gọn được trong $\mathcal{A}$.

Tập các thuộc tính R ⊆ A được gọi là *tập đã gọn* của A nếu $\mathcal{A}'$ = (U, R∪D) là rút gọn và $POS_R(D) = POS_A(D)$.

Tập tất cả các thuộc tính không thể biến mất trong $\mathcal{A}$ biểu diễn là CORE(A) (gọi là tập nhân) và được xác định như sau:

$$CORE(A) = \cap RED(A)$$

với RED(A) là tập tất cả các *tập rút gọn* của A.

Ví dụ 1. Tập thuộc tính rút gọn và thuộc tính nhân biểu diễn như sau:

	Nơi sinh	*Tôn giáo*	*Tới nước*	*Xem xét*
x_1	Sài gòn	Có	Mỹ	Cấm
x_2	Sài gòn	Có	Pháp	Nghi ngờ
x_3	Sài gòn	Có	Đức	Cấm
x_4	Hà nội	Có	Mỹ	Không
x_5	Hà nội	Không	Pháp	Không
x_6	Hà nội	Có	Đức	Cấm

Tập rút gọn Red1 = *{Tôn giáo, Tới nước}*

	Tôn giáo	*Tới nước*	*Xem xét*
x_1, x_4	Có	Mỹ	Cấm
x_2	Có	Pháp	Nghi ngờ
x_3, x_6	Có	Đức	Cấm
x_5	Không	Pháp	Không

Tập rút gọn thứ 2 Red2 = {*Nơi sinh, Tới nước*}

	Nơi sinh	*Tới nước*	*Xem xét*
x_1	Sài gòn	Mỹ	Cấm
x_2	Sài gòn	Pháp	Nghi ngờ
x_3	Sài gòn	Đức	Cấm
x_4	Hà nội	Mỹ	Không
x_5	Hà nội	Pháp	Không
x_6	Hà nội	Đức	Cấm

Tập thuộc tính nhân CORE = {*Nơi sinh, Tới nước*} ∩ {*{Tôn giáo, Tới nước* } = {*Tới nước*}.

I.2.1.3. Ma trận phân biệt được và hàm phân biệt được

Xem xét bảng quyết định (**bảng 3**). Giả sử $\mathcal{A}$ = (U, A ∪ D) với

U = {x_1,x_2,x_3 .... x_7}

A = {*Tới nước, Số hộ chiếu, Tôn giáo, Nơi sinh*}

D = {*Cấm xuất nhập*}

Ví dụ có một tập rút gọn {*Số hộ chiếu, Tôn giáo*} phân biệt được các đối tượng trong trường hợp giống nhau cũng như tập đầy đủ các đối tượng được xem xét.

	Tới nước	*Số hộ chiếu*	*Tôn giáo*	*Nơi sinh*	*Xem xét*
x_1	Mỹ	PT1234	Có	Hà nội	Cấm
x_2	Mỹ	NG1234	Có	Sài gòn	Không cấm
x_3	Pháp	NG1234	Có	Đà nẵng	Không cấm
x_4	Đức	CV1234	Có	Sài gòn	Cấm
x_5	Đức	PT1234	Có	Sài gòn	Không cấm
x_6	Đức	CV1234	Có	Hà nội	Cấm
x_7	Mỹ	CV1234	Không	Đà nẵng	Cấm
x_8	Pháp	NG1234	Không	Hà nội	Không cấm

Bảng 3. Một ví dụ bảng quyết định chưa rút gọn

Ma trận phân biệt được của $\mathcal{A}$ ký hiệu là là M($\mathcal{A}$) là một ma trận đối xứng $n \times n$ với phần tử c_{ij} cho như sau:

$$c_{ij} = \begin{cases} \{a \in A : a(x_i) \neq a(x_j)\} & \text{nếu } \exists d \in D\left[d(x_i) \neq d(x_j)\right] \\ \lambda & \text{nếu } \forall d \in D\left[d(x_i) = d(x_j)\right] \end{cases}$$

với $1 \leq j \leq i \leq n$ thì x_i, x_j thuộc A- vùng khẳng định của D.

c_{ij} là tập tất cả các thuộc tính điều kiện mà phân loại x_i, x_j thành các lớp khác nhau.

Hàm phân biệt được $f_{\mathcal{A}}$ cho một hệ thông tin $\mathcal{A}$ là một hàm kiểu Boolean của m biến logic $a_1^*,...,a_m^*$ (tương ứng với các thuộc tính a_1, ..., a_m) được xác định như sau với c_{ij}={ a^* | $a \in c_{ij}$}

$$f_{\mathcal{A}}(a_1^*,...,a_m^*) = \Lambda\left\{\vee c_{ij}^* \middle| 1 \leq j \leq i \leq n, c_{ij} \neq \varnothing\right\}$$

Với $\vee c_{ij} = \perp$(false) nếu $c_{ij} \neq \varnothing$

$\vee c_{ij} = t$(true) nếu $c_{ij} = \lambda$

I.2.2. Quá trình khám phá tri thức theo cách tiếp cận tập thô

Tìm kiếm tri thức từ dữ liệu đã và đang là vấn đề rất được rất nhiều người quan tâm [9, 10]. Việc tìm kiếm tri thức từ kho dữ liệu khổng lồ đã được giải quyết theo nhiều phương pháp trong đó nổi bật lên là phương pháp khai phá tri thức theo cách tiếp cận tập thô do Z.Pawlak đề xuất vào những năm 80 của thế kỉ XX. Phương pháp này đặc biệt hiệu quả đối với những tập dữ liệu rất lớn với nhiều kiểu dữ liệu khác nhau. Nó cũng có khả năng làm việc tốt với dữ liệu không chắc chắn, không hoàn hảo hoặc dữ liệu hay thay đổi mà đôi khi cần phải suy đoán (sử dụng tri thức nền).

I.2.2.1. Sự rời rạc hoá dựa trên tập thô và lập luận logic

Xuất phát từ thực tế đó, các tác giả [3, 9] đã đưa ra một số phương pháp khai phá dữ liệu một cách hiệu quả chẳng hạn như sử dụng phương pháp rời rạc hoá dữ liệu dựa trên tập thô và lập luận logic. Phương pháp này được đưa ra để giải quyết điểm yếu của loại dữ liệu hỗn tạp với những giá trị liên tục, hay giá trị mang tính chất tượng trưng bằng cách phân chia các giá trị thuộc tính thành các khoảng. Tuy nhiên, có rất nhiều phương pháp được sử dụng để rời rạc hoá dữ liệu như: Sử dụng phương pháp lập luận logic, thuật toán NAIVE, thuật toán Semi-NAIVE,... nhưng người ta vẫn chưa tìm được một phương pháp chung nhất cho việc rời rạc hoá, việc lựa chọn phương pháp tuỳ thuộc rất nhiều vào dữ liệu cần xử lý.

Khi sử dụng phương pháp rời rạc hoá có nghĩa là chúng ta chấp nhận sai số trong dữ liệu. Ví dụ nhiệt độ thường được đo bởi một con số thực, tuy nhiên người ta có thể phân chia nó thành một, hai hoặc nhiều khoảng hữu hạn (Nhiệt độ cao, thấp, trung bình); Một ví dụ khác là việc đo nhịp tim các bác sĩ thường phân biệt những khoảng 68 đến 72 nhịp/phút là bình thường, hoặc 120 đến 140 nhịp/phút là cao, 48 đến 56 nhịp/phút là thấp. Có thể thấy rằng việc chọn các khoảng thích hợp và phân chia các giá trị thuộc tính mang tính chất tượng trưng là một vấn đề phức tạp phụ thuộc nhiều vào số các thuộc tính điều kiện được đưa vào quá trình rời rạc hoá.

I.2.2.2. Lựa chọn thuộc tính dựa trên tập thô với phương pháp đánh giá kinh nghiệm

Một cơ sở dữ liệu thường chứa rất nhiều các thuộc tính dư thừa và không cần thiết cho việc tìm kiếm tri thức trong dữ liệu. Nếu các thuộc tính dư thừa không được loại bỏ thì không những độ phức tạp về thời gian tìm kiếm tri thức là

rất lớn mà chất lượng tri thức tìm được cũng không cao. Mục tiêu của việc lựa chọn thuộc tính là tìm ra những tập thuộc tính tối ưu trong cơ sở dữ liệu, dựa vào đó, việc sinh luật và phân lớp có thể đạt được hiệu quả cao nhất mà chỉ sử dụng những tập thuộc tính con đã được lựa chọn.

Tư tưởng cơ bản của việc lựa chọn thuộc tính sử dụng tập thô với phương pháp đánh giá kinh nghiệm như sau [9]:

- Lựa chọn các thuộc tính trong nhân (CORE) làm tập con ban đầu
- Tại mỗi bước, lựa chọn các thuộc tính sử dụng tiêu chuẩn đánh giá trong quá trình khám phá luật bởi bảng phân bố tổng quát trong tập thô (phần 2.2.3).
- Dừng lại khi tập con các thuộc tính được chọn là một tập rút gọn.

Số lượng của các tập rút gọn có thể là 2^{N-1} trong đó N là số các thuộc tính. Việc lựa chọn tập rút gọn tối ưu từ các tập rút gọn có thể là rất tốn thời gian do đó phải sử dụng phương pháp kinh nghiệm. Đặc điểm chính của phương pháp lựa chọn thuộc tính dựa trên tập thô với phương pháp đánh giá kinh nghiệm là nó có thể tìm ra các tập con thuộc tính nhanh và hiệu quả từ cơ sở dữ liệu lớn, các thuộc tính được lựa chọn không làm giảm đi tính ưu việt của thuật toán quy nạp nhiều lắm.

Có hai phương pháp lựa chọn thuộc tính thường được sử dụng đó là lọc và bọc. Tư tưởng chính phương pháp thứ nhất (phương pháp lọc) là lựa chọn các thuộc tính tối thiểu trong những thuộc tính đó, chọn ra những thuộc tính có độ phù hợp cao hơn theo tiêu chuẩn sau:

- Lựa chọn các thuộc tính làm cho số các trường hợp thoả mãn tăng nhanh (đạt được tập con với số thuộc tính là càng nhỏ càng tốt)
- Chọn các thuộc tính có ít giá trị khác nhau (để dảm bảo số các trường hợp được bảo phủ bởi luật cầng nhiều càng tốt)

Lợi điểm của phương pháp này là tốc độ nhanh tuy nhiên, nó không tận dụng được tính ưu việt của thuật toán quy nạp. Phương pháp thứ hai sử dụng thuật toán quy nạp cho việc đánh giá, tư tưởng chính của phương pháp này là sử dụng 3 cách tìm kiếm: tìm kiếm toàn bộ, tìm kiếm kinh nghiệm và tìm kiếm không xác định. Lợi điểm của phương pháp bọc là tận dụng được tính ưu việt của thuật toán quy nạp tuy nhiên nó có độ phức tạp thời gian cao.

I.2.2.3. Khám phá luật bởi bảng phân bố tổng quát dựa trên tập thô

A. Skowron và Ning Zong [9] đã đưa ra phương pháp khám phá luật sử dụng bảng phân bố tổng quát dựa trên tập thô, với ý tưởng như sau:

- Từ bảng quyết định xây dựng bảng phân bố tổng quát
- Dựa trên bảng phân bố tổng quát này sinh các vector phân biệt được
- Tạo ra các tập rút gọn từ các vector phân biệt được
- Sinh ra các luật bao phủ tất cả các trường hợp

Đặc điểm chính của bảng phân bố tổng quát dựa trên tập thô là:

- Bảng phân bố tổng quát mô tả quan hệ xác suất giữa các trường hợp có thể và các bộ sinh có thể.
- Những trường hợp không thấy trong quá trình khai phá dữ liệu, sự không chắc chắn của luật bao gồm cả khả năng dự đoán trước các trường hợp của nó được thể hiện rõ ràng trong độ mạnh của luật.
- Hướng tìm kiếm có thể được lựa chọn một cách mềm dẻo, có thể sử dụng tri thức nền làm cơ sở cho việc tạo bảng phân bố tổng quát và quá trình khai phá.

I.2.3. Khám phá mẫu trong hệ thông tin

Hiện nay, các nhóm nghiên cứu về khai phá dữ liệu đang nghiên cứu và tìm kiếm những phương pháp tìm ra những khuôn mẫu từ liệu (gọi là mẫu) [5, 6, 9]. Người ta quan tâm đến những mẫu quan hệ phức tạp hơn được rút ra một cách tự

động từ dữ liệu. Trong trường hợp đơn giản thì mẫu là một vector giá trị có độ dài đủ lớn của một số thuộc tính được hỗ trợ bởi số lượng đủ nhiều các đối tượng. Bài toán tìm kiếm mẫu tối ưu có độ phức tạp tính toán lớn đòi hỏi phải có thuật toán đánh giá kinh nghiệm đủ tốt để rút ra những mẫu gần tối ưu một cách hiệu quả từ những kho dữ lớn. Một lớp quan trọng của phương pháp tìm kiếm mẫu từ dữ liệu được dựa trên các khuôn mẫu quan hệ. Những khuôn mẫu này được xác định từ một bảng dữ liệu cho trước sử dụng quan hệ thứ lỗi trong một số lớp quan hệ thứ lỗi giả định trước. Một quan hệ thứ lỗi là tối ưu nếu tập các tham số miêu tả quan hệ này cho phép xây dựng những khuôn mẫu dữ liệu thích hợp trên bảng dữ liệu cho trước.

Có nhiều ứng dụng cho việc tìm khuôn mẫu từ dữ liệu. Một số có thể sử dụng để phân tách các bảng dữ liệu lớn. Tập dữ liệu hỗ trợ một mẫu cho trước có thể được coi là phổ biến trong một miền con của tập đối tượng tổng thể bởi vì nó chứa rất nhiều các đối tượng có cùng một thuộc tính. Bảng dữ liệu lớn có thể được phân chia thành một cây nhị phân của các mẫu hoặc khuôn mẫu. Mỗi nút của cây phụ thuộc vào một bước phân tách. Quá trình phân chia dừng lại khi một bảng con được gắn với một lá có kích cỡ vừa đủ đối với một phương pháp sinh luật quyết định hiện có. Người ta áp dụng những phương pháp tìm kiếm mẫu quyết định từ các bảng quyết định gắn với các lá đã có dựa trên cách tiếp cận tập thô. Quá trình phân lớp cho một đối tượng mới bắt đầu bằng việc tìm ra đường đi trên cây bằng cách so sánh các mẫu. Sau đó đối tượng được phân lớp dựa trên luật quyết định được sinh ra từ bảng con gắn với các lá ở trên đường đó.

Người ta cũng thảo luận về các chiến lược tìm kiếm khuôn mẫu có trong các lớp quyết định. Quá trình này có thể được coi như việc tìm luật quyết định xấp xỉ mạnh ngầm định.

Các phương pháp này có thể được dùng để tìm luật quyết định xấp xỉ tổng hợp từ các bảng dữ liệu. Bản chất xấp xỉ của những luật này được mô tả bởi một số ràng buộc. Luật quyết định mạnh có thể được hiểu giống như trong trường hợp của sự kết hợp nhưng cũng có thể được mô tả bởi một số các ràng buộc khác ví dụ việc giả định một đặc trưng của luật quyết định xấp xỉ đã được tổng hợp được bảo đảm bởi các mẫu hay các khuôn mẫu đã được tìm ra.

I.3. KẾT LUẬN CHƯƠNG I

Phát hiện luật theo tiếp cận lý thuyết tập thô do Z.Pawlak đề xuất đầu tiên vào những năm 80 của thập kỷ XX. Đây là một trong những phương pháp đang được nhiều nhà khoa học nghiên cứu và sử dụng trong quá trình khám phá tri thức từ dữ liệu. Các khái niệm nền tảng trong lý thuyết tập thô là hệ thông tin, bảng quyết định, quan hệ không phân biệt được, tập xấp xỉ và sự phụ thuộc thô. Phát hiện luật là một trong những kỹ thuật cơ bản và hiệu quả của khai phá dữ liệu. Hiện tượng dữ liệu không đầy đủ, dư thừa hoặc không chính xác, dữ liệu dạng ký hiệu có thể tồn tại trên thực tế gây ảnh hưởng không tốt tới quá trình phát hiện ra tri thức chính xác từ dữ liệu. Việc sử dụng tri thức nền (hay tri thức kinh nghiệm) trong việc lựa chọn luật có thể làm giảm bớt số thuộc tính cần xem xét tạo luật từ đó làm giảm độ phức tạp tính toán của quá trình khám phá tri thức.

CHƯƠNG 2. KHÁM PHÁ LUẬT THEO TIẾP CẬN TẬP THÔ VÀ ĐỐI SÁNH VỚI KHÁM PHÁ LUẬT KẾT HỢP

II.1. KHÁM PHÁ LUẬT KẾT HỢP, NỘI DUNG CƠ BẢN CỦA KHÁM PHÁ TRI THỨC TRONG CƠ SỞ DỮ LIỆU

II.1.1. Luật kết hợp

Khảo sát hệ thống gồm tập các phiếu bán hàng của một công ty với sự hạn chế là chúng ta mới chỉ quan tâm đến tên các mặt hàng xuất hiện trong phiếu bán hàng và hy vọng rằng tồn tại mối liên quan nào đó giữa các mặt hàng trong một hệ thống như vậy. Điều đó có nghĩa là miền giá trị của một thuộc tính là {0, 1} hay {sai, đúng}. Luật kết hợp được xuất phát từ những mệnh đề có dạng: "*98% khách hàng* mà *mua tạp chí thể thao* thì đều *mua các tạp chí về ôtô*". Kiểu mô tả như vậy cho phép cung cấp hồ sơ thông tin chung về khách hàng để công ty đó có thể sử dụng trong các chiến dịch tiếp thị. Trong các hệ thống đang được nghiên cứu, tập tên tất cả các thuộc tính (còn gọi là mục - item; trong hệ thống bán hàng, mỗi thuộc tính tương ứng với một mặt hàng cần được bán) được ký hiệu là $\mathcal{I}$.

Cho X là một tập con các thuộc tính ($X \subseteq \mathcal{I}$), lúc đó X được gọi là tập mục (itemset). Số thuộc tính (số mục) trong tập X được gọi là cỡ của tập mục X. Nếu X có cỡ k thì X được gọi là k-tập mục.

Theo cách diễn đạt thông thường, luật kết hợp được viết dưới dạng $X \Rightarrow Y \mid (c,s)$ với:

- X và Y là các *tập mục* và $X \cap Y = \varnothing$,
- c là *độ tin cậy* của luật,
- s là *độ hỗ trợ* của luật

Độ tin cậy của luật biểu thị *độ mạnh luật* được tính bằng tỷ lệ phần trăm các bản ghi mà tất cả các thuộc tính trong Y đều có giá trị đúng trong số tất cả các bản ghi mà tất cả các thuộc tính trong X đều có giá trị đúng.

Độ hỗ trợ của luật là độ đo có ý nghĩa thống kê của luật, tức là tỷ lệ phần trăm các bản ghi mà tất cả các thuộc tính trong $X \cup Y$ có giá trị đúng.

Để minh họa, chúng ta xem xét một tập dữ liệu bán hàng tại siêu thị. Trong đó, các bản ghi (phiếu bán hàng) thể hiện các mặt hàng được bán trong siêu thị như "*Sữa, Bơ, Bánh mì, Xà phòng, Nước ép trái cây*".

Luật kết hợp dạng {*Bánh mì, Sữa*} $\Rightarrow$ {*Nước ép trái cây*} $\mid$(0.98, 0.70) có nghĩa là:

- có tới 70% số lượt khách hàng mua cả ba mặt hàng *Bánh mì, Sữa, Nước ép trái cây,*
- và 98% số lượt khách hàng nếu mua *Bánh mì* và *Sữa* thì cũng mua kèm thêm *Nước ép trái cây*.

Dưới đây, chúng ta sẽ trình bày khái niệm luật kết hợp một cách hình thức hơn. Giả sử $\mathcal{I} = \{i_1, i_2, ..., i_m\}$ là một tập toàn bộ các mục (item). Trong ví dụ trên, $\mathcal{I}$ chính là tập tên các mặt hàng), $\mathcal{D}$ là một tập các giao tác trong đó mỗi giao tác $T \in \mathcal{D}$ chính là một tập các mục $T \subseteq \mathcal{I}$ (trong ví dụ trên, mỗi giao tác T tương ứng với một phiếu mua hàng, T gồm tên các mặt hàng có trong phiếu mua hàng đó). Mỗi giao tác được liên kết với một định danh duy nhất (được gọi là *TID*) của nó. Giao tác T chứa X (tập các mục trong $\mathcal{I}$) được biểu diễn bằng quan hệ $X \subseteq T$.

Định nghĩa 2.1 (Luật kết hợp)

Luật kết hợp là một biểu diễn dạng $X \Rightarrow Y$ với $X \subset \mathcal{I}$, $Y \subset \mathcal{I}$ và $X \cap Y = \varnothing$.

Định nghĩa 2.2 (Độ hỗ trợ của một tập mục)

Cho X là một tập mục. Độ hỗ trợ của X, kí hiệu là supp(X), là đại lượng tần số các giao tác có chứa X trong tập tất cả các giao tác.

$\text{supp}(X) = \frac{card(\{T : X \subseteq T\})}{card(D)}$ trong đó ***card*** là hàm tính số lượng (cardinal).

Mệnh đề 2.1.

Nếu $A \subseteq B$ với A, B là các tập mục thì $supp(A) \geq supp(B)$.

Kết quả này nhận được từ lập luận rằng là mỗi giao dịch trong $\mathcal{D}$ nếu đã hỗ trợ B thì tất yếu hỗ trợ A.

Định nghĩa 2.3 (Độ hỗ trợ và độ tin cậy của luật kết hợp)

Độ hỗ trợ của luật kết hợp $X \Rightarrow Y$, ký hiệu là $supp(X \Rightarrow Y)$, được xác định theo: $supp(X \Rightarrow Y) = \text{supp}(X \cup Y)$

Độ tin cậy của luật kết hợp $X \Rightarrow Y$, ký hiệu là $conf(X \Rightarrow Y)$, được xác định theo: $conf(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$

Nhận xét: Độ tin cậy của luật kết hợp có dạng một "xác suất có điều kiện" của sự kiện xuất hiện Y khi đã xuất hiện X.

Độ hỗ trợ mang ý nghĩa "độ mạnh" theo nghĩa ảnh hưởng của luật kết hợp trong toàn bộ hệ thống, độ tin cậy mang ý nghĩa về tính tin cậy của phát biểu "nếu X thì Y". Khái niệm tập phổ biến như trình bày trong phần sau cho thấy mục tiêu "có giá trị" của khám phá luật kết hợp.

II.1.2. Một số cơ sở toán học khai phá luật kết hợp

II.1.2.1. Tập phổ biến

Định nghĩa 2.4 (Tập phổ biến)

Tập mục $X \subseteq \mathcal{I}$ thoả mãn *supp(X)* ≥ *minsup* với *minsup* là độ hỗ trợ tối thiểu cho trước thì X được gọi là tập phổ biến.

Khái niệm tập phổ biến cho biết rằng, chúng ta chỉ khám phá các luật có "độ ảnh hưởng" vượt quá một ngưỡng nào đó hay cũng vậy, chúng ta bỏ qua các luật ít có ảnh hưởng.

Từ mệnh đề 2.1 và định nghĩa tập phổ biến, nhận được hệ quả sau đây.

Hệ quả. 2.1. Cho A, B là hai tập mục, $A \subseteq B$.

a. Nếu B là tập phổ biến thì A cũng là tập phổ biến.

b. Nếu A là tập không phổ biến thì B cũng là tập không phổ biến.

II.1.2.2. Khai phá luật kết hợp dựa trên tập phổ biến

Khai phá luật kết hợp trong cơ sở dữ liệu đã thu hút sự chú ý của nhiều nhóm nghiên cứu về KDD [2, 7]. Mục tiêu là sinh ra tất cả các luật có độ hỗ trợ và độ tin cậy lớn hơn độ hỗ trợ tối thiểu cho trước (gọi là *minsup*) và độ tin cậy cho trước (gọi là *minconf*). Bài toán chia ra làm 2 bước:

- Sinh ra tất cả các tập mục có đỗ hỗ trợ lớn hơn *minsup* (các tập phổ biến).
- Với mỗi tập phổ biến, sinh ra tất cả các luật có độ tin cậy lớn hơn *minconf*.

Việc sinh ra tất cả các luật dựa trên tập phổ biến (bước 2) có thể được giải quyết tóm tắt như sau: Với mỗi tập phổ biến X và một tập con Y của X ($Y \subset X$), xem xét tập X' = X\Y bao gồm các phần tử của X mà không thuộc Y. Nếu tỷ số giữa độ hỗ trợ của X với độ hỗ trợ của X' mà lớn hơn *minconf* thì sinh ra luật $X' \Rightarrow Y$.

Việc sinh ra luật kết hợp bằng cách sử dụng tất cả các tập phổ biến tương đối đơn giản, tuy nhiên việc phát hiện ra tất cả các tập phổ biến cùng với những giá trị độ hỗ trợ của chúng lại là một bài toán khó nếu lực lượng của tập dữ liệu là lớn.

Thông thường một siêu thị có m (m lên đến hàng nghìn) mặt hàng (mục), số lượng các tập mục khác nhau sẽ là 2^m, do đó việc tính toán độ hỗ trợ cho các tập mục đòi hỏi nhiều thời gian.

Để giảm bớt không gian tìm kiếm tổ hợp, thuật toán tìm luật kết hợp có thể khai thác 2 tính chất của tập phổ biến đã được phát biểu trong hệ quả 2.1.

Đây là các đặc điểm có thể sử dụng cho thuật toán cơ sở tìm tất cả các tập phổ biến, giống như thuật toán Apriori [2], có thể tóm tắt những bước chính như sau:

1- Tìm tập tất cả các tập phổ biến có cỡ là 1 (Tính độ hỗ trợ của mọi 1-tập mục bằng việc quét toàn bộ cơ sở dữ liệu. Hủy đi các 1-tập mục không là tập phổ biến).
2- Mở rộng 1-tập mục phổ biến nhận được từ bước 1 để có được các 2-tập mục bằng cách lần lượt bổ sung thêm một mục vào 1-tập mục phổ biến để sinh ra tất cả các 2-tập mục cho việc lựa chọn tiếp theo. Tính độ hỗ trợ của các 2-tập mục được sinh ra và loại bỏ tất cả các 2-tập mục không là tập phổ biến.
3- Lặp lại các bước trên cho đến bước thứ k, tập phổ biến (k-1) được mở rộng thành k-tập mục và kiểm tra tính phổ biến.

Quá trình trên được lặp lại cho đến khi không tìm được tập phổ biến mới. Có một số thuật toán dựa trên các bước chính này đã được giới thiệu, chúng khác nhau chủ yếu bởi việc sinh ra các tập mục cho các lần kiểm tra tiếp theo và cách tính toán độ hỗ trợ của các tập mục đó.

II.2. QUÁ TRÌNH KHÁM PHÁ TRI THỨC THEO TIẾP CẬN TẬP THÔ

II.2.1. Quá trình khám phá luật trong bảng quyết định

II.2.1.1. Luật trong bảng quyết định

Giả sử $\mathcal{A}$ = (U, A∪{d}) là một bảng quyết định; X biểu thị sự kết hợp giữa các từ nhận dạng (descriptors) bao hàm trong các thuộc tính điều kiện A; Y biểu thị một từ nhận dạng d=v trong đó v là bất kỳ một giá trị nào của thuộc tính quyết định d [5, 9].

Định nghĩa 2.5 (Luật theo tiếp cận tập thô)

Một luật quyết định có dạng "Nếu X thì Y" được biểu diễn bởi $\mathbf{X \rightarrow Y}$ với **S** biểu thị *độ mạnh* của luật được tính theo công thức trong phần II.2.1.2.

II.2.1.2. Hai đặc trưng của luật: Độ mạnh và độ nhiễu của luật

Cho luật $\mathbf{X \rightarrow Y}$, độ mạnh của luật này, ký hiệu là S (X → Y) được xác định theo công thức sau:

$$\mathbf{s(X)(1\text{-}r(X \rightarrow Y))}$$

Với s (X) được là độ mạnh của X được xác định như dưới đây.

* Trong trường hợp không sử dụng tri thức kinh nghiệm, s(X) được tính như sau:

$$s(X) = s(PG_k) = \sum_l p(Pl_l \mid PG_k) = \frac{N_{ins-rel}(PG_k)}{N_{PG_k}}$$

Với,

$N_{ins-rel}(PG_k)$ là số các đối tượng quan sát thoả mãn trong lần thứ i .

* Trong trường hợp sử dụng tri thức kinh nghiệm, độ mạnh s(X) được tính như sau:

$$s(X) = s(PG_k) = \sum_l p_{bk}(PI_l \setminus PG_k) = \frac{\sum BKF(PI_l \setminus PG_k)}{N_{PG_k}}$$

Độ nhiễu r(X→ Y) được tính như sau:

$$\boldsymbol{r}(\mathbf{X} \rightarrow \mathbf{Y}) = \frac{N_{ins-rel}(X) - N_{ins-class}(X,Y)}{N_{ins-rel}(X)}$$

Với $N_{ins-class}(X,Y)$ là số các đối tượng thuộc lớp Y trong các trường hợp thoả mãn bộ sinh X.

II.2.1.3. Quá trình khám phá luật

Quá trình dưới đây thực hiện theo phương pháp được trình bày trong [9]. Giả sử có bảng quyết định $\mathcal{A}$ = (U, A ∪ {d}) miêu tả như sau:

U	*Tới nước*	*Nghề nghiệp*	*Nơi sinh*	*Xem xét*
u_1	Mỹ	Công nhân	Hà Nội	Cấm
u_2	Mỹ	Kĩ sư	Hà Nội	Cấm
u_3	Mỹ	Công nhân	Hà Nội	Cấm
u_4	Pháp	Kĩ sư	Sài Gòn	Không
u_5	Mỹ	Công nhân	Hà Nội	Không
u_6	Mỹ	Nông dân	Hà Nội	Không
u_7	Pháp	Kĩ sư	Hà Nội	Cấm

Bảng gồm các thuộc tính điều kiện là *Tới nước, Nghề nghiệp, Nơi sinh*.

Tập giá trị của thuộc tính *Tới nước* là: $V_{Tới\ nước}$ = *{Mỹ,Pháp}*

Tập giá trị của thuộc tính *Nghề nghiệp*: $V_{Nghề\ nghiệp}$ = *{ Công nhân, Kĩ sư, Nông dân}*

Tập giá trị của thuộc tính *Nơi sinh* là: $V_{Nơi\ sinh}$ = *{Hà Nội, Sài Gòn }*

Thuộc tính quyết định là *Xem xét*, tập giá trị là $V_{Xem\ xét}$ = *{cấm,không}*

Bảng quyết định tương ứng miêu tả trong GDT-RS (bảng phân bố tổng quát) như sau:

F(x) *G(x)*	*Mỹ* *Công nhân* *Sài Gòn*	*Mỹ* *Công nhân* *Hà Nội*	---------	*Pháp* *Công nhân* *Sài Gòn*	-------	*Pháp* *Nông dân* *Hà Nội*
**Công nhân Sài Gòn*	1/2		---------	1/2	--------	
**Công nhânHàNội*		1/2			--------	
**Kĩ sư Sài Gòn*					--------	
**Kĩ sư Hà Nội*					--------	
**Nông dân Sài Gòn*					--------	
**Nông dân Hà Nội*					--------	1/2
*Mỹ *Sài Gòn*	1/3				--------	
----------	----------				--------	
*Pháp Kĩ sư**					--------	
*Pháp Nông dân**					--------	1/2
***Sài Gòn*	1/6			1/6	--------	
----------	----------				--------	
*Mỹ***	1/6	1/6			--------	
*Pháp **				1/6	--------	1/6

Trong đó: *F(x)* là các đối tượng có thể (*PI*)

G(x) là các bộ sinh có thể (*PG*)

$G(x) \rightarrow F(x)$ là quan hệ xác suất giữa *PI* và *GI* và được xác định là:

$$p(PI_j | PG_i) = \begin{cases} \dfrac{1}{N_{PG_i}} & \text{nếu } PI_j \neq PG_i \\ 0 & \text{Các trường hợp khác} \end{cases}$$

Trong đó $N_{PG_i} = \prod_{k \in \{l | PG[l]=*\}} n_k$ là số *PI* thoả mãn *PG* thứ *i*.

a) Từ bảng quyết định trên xét trường hợp có tỷ lệ nhiễu là = 0.

U	*Tới nước*	*Nghề nghiệp*	*Nơi sinh*	*Xem xét*
$u_1' \begin{Bmatrix} u_1 \\ u_3 \\ u_5 \end{Bmatrix}$	Mỹ	Công nhân	Hà Nội	Cấm, Cấm, Không
u_2	Mỹ	Kĩ sư	Hà Nội	Cấm
u_3	Mỹ	Công nhân	Hà Nội	Cấm
u_4	Pháp	Kĩ sư	Sài Gòn	Không
u_5	Mỹ	Công nhân	Hà Nội	Không
u_6	Mỹ	Nông dân	Hà Nội	Không
u_7	Pháp	Kĩ sư	Hà Nội	Cấm

$\Downarrow$

U	*Tới nước*	*Nghề nghiệp*	*Nơi sinh*	*Xem xét*
u_1'	Mỹ	Công nhân	Hà Nội	$\perp$
u_2	Mỹ	Kĩ sư	Hà Nội	Cấm
u_4	Pháp	Kĩ sư	Sài Gòn	Không
u_6	Mỹ	Nông dân	Hà Nội	Không
u_7	Pháp	Kĩ sư	Hà Nội	Cấm

Ta có : $r_{\{cấm\}}(u_1') = 1 - \frac{2}{3} = 0.33$ và $r_{\{không\}}(u_1') = 1 - \frac{1}{3} = 0.67$

Đặt $T_{nhiễu} = 0$ thì $r_{\{cấm\}}(u_1') = 0.33 > T_{nhiễu}$ và $r_{\{không\}}(u_1') = 0.67 > T_{nhiễu}$

như vậy là $d(u_1') = \perp$

- Tạo vector phân biệt cho u_2

Sử dụng công thức [3] $c_{ij} = \begin{cases} \{a \in A : a(x_i) \neq a(x_j)\} & \text{nếu } \exists d \in D[d(x_i) \neq d(x_j)] \\ \lambda & \text{nếu } \forall d \in D[d(x_i) = d(x_j)] \end{cases}$

Vector phân biệt cho u_2 được tính như sau:

$m_{2,1}$ = *{Nghề nghiệp}*

$m_{2,2} = \lambda$

$m_{2,4}$ = *{Tới nước, Nơi sinh}*

$m_{2,6}$ = {Nghề nghiệp}

$m_{2,7} = \lambda$

	$u_1^{'}$	u_2	u_4	u_6	u_7
u_2	Nghề nghiệp	λ	Tới nước, Nơi sinh	Nghề nghiệp	λ

- Tìm tập rút gọn cho u_2

$f_T(u_2) = (Nghề\ nghiệp) \wedge T \wedge (Tới\ nước \vee Nơi\ sinh) \wedge (Nghề\ nghiệp) \wedge T$

$= (Nghề\ nghiệp) \wedge (Tới\ nước \vee Nơi\ sinh)$

$= (Tới\ nước \wedge Nghề\ nghiệp\ ) \vee (Nghề\ nghiệp \wedge Nơi\ sinh)$

- Tạo luật cho u_2

$f_T(u_2) = \underset{\{Mỹ, Kĩ\ sư\}}{\underline{(Tới\ nước \wedge Nghề\ nghiệp\ )}} \vee \underset{\{Kĩ\ sư, Hà\ Nội\}}{\underline{(Nghề\ nghiệp \wedge Nơi\ sinh)}}$

$s(\{Mỹ, Kĩ\ sư\} = 0.5$

$r(\{Mỹ, Kĩ\ sư\} \rightarrow Cấm) = 0$

$(Mỹ, Kĩ\ sư\} \rightarrow Cấm\ với\ S = (1 \times 1/2) \times (1\text{-}0) = 0.5$

$s(\{Kĩ\ sư, Hà\ Nội\} = 1$

$r(\{Kĩ\ sư, Hà\ Nội\} \rightarrow Cấm) = 0$

$\{Kĩ\ sư, Hà\ Nội\} \rightarrow Cấm\ với\ S = (2 \times 1/2) \times (1\text{-}0) = 0$

- Tạo vector phân biệt cho u_4

Vector phân biệt cho u_4 được tính như sau:

$m_{4,1'}$ = {Tới nước, Nghề nghiệp, Nơi sinh}

$m_{4,2}$ = {Tới nước, Nơi sinh}

$m_{4,4} = \lambda$

$m_{4,6} = \lambda$

$m_{4,7}$ = {Nơi sinh}

	$u_1^{'}$	u_2	u_4	u_6	u_7
u_4	*Tới nước, Nghề nghiệp, Nơi sinh*	*Tới nước, Nơi sinh*	λ	λ	*Nơi sinh*

- Tìm tập rút gọn cho u_4

$f_T(u_4)$ = (*Tới nước* $\vee$ *Nghề nghiệp* $\vee$ *Nơi sinh*) $\wedge$ (*Tới nước* $\vee$ *Nơi sinh*) $\wedge$ T $\wedge$ T $\wedge$ (*Nơi sinh*)

= (*Nơi sinh*)

- Tạo luật cho u_4

$f_T(u_4) = (\underline{Nơi\ sinh})$
$\{Sài\ Gòn\}$

$s(SG) = 1/6$

$r(\{Sài\ Gòn\} \rightarrow Không) = 0$

$\{SG\} \rightarrow Không$ với $S = (1 \times 1/6) \times (1 - 0) = 0.167$

Sau lần lượt các bước tạo vector phân biệt, tạp tập rút gọn, tạo luật cho u_6, u_7 ta có luật cho tất cả các trường hợp như sau:

u_2: $\{Mỹ, Kĩ\ sư\} \rightarrow Cấm$, với $S=0.5$

$\{Kĩ\ sư, Hà\ Nội\} \rightarrow Cấm$ với $S=1$

u_4:$\{Sài\ Gòn\} \rightarrow Không$, với $S=0.167$

u_6:$\{Nông\ dân\} \rightarrow Không$, với $S=0.25$

u_7: $\{Mỹ, Hà\ Nội\} \rightarrow Cấm$, với $S=0.5$

$\{Kĩ\ sư, Hà\ Nội\} \rightarrow Cấm$ với $S=1$

Bộ sinh thuộc lớp *Cấm*

	u_2 *Mỹ,kĩ sư,Hà Nội*	u_7 *Pháp,Kĩ sư,Hà Nội*
**,Kĩ sư,Hà Nội*	1/2	1/2
Pháp,,Hà Nội*		1/3
*Mỹ, Kĩ sư,**	1/2	

{Kĩ sư,Hà Nội} $\rightarrow$ *Cấm* *với S=1* u_2,u_7

{Mỹ, Hà Nội} $\rightarrow$ *Cấm,* *với S=0.5* u_7

{Mỹ, Kĩ sư} $\rightarrow$ *Cấm,* *với S=0.5* u_2

Bộ sinh thuộc lớp *Không*

	u_4 *Mỹ,Nông dân,Hà Nội*	u_6 *Pháp,Kĩ sư,Sài Gòn*
**,*,Sài Gòn*		1/6
,Nông dân,	1/4	

{Sài Gòn} $\rightarrow$ *Không* *với S=1/6* u_4

{Nông dân} $\rightarrow$ *Không,* *với S=1/4* u_6

Các luật sinh ra với tỷ lệ nhiễu = 0 ($T_{nhiễu} = 0$)

- Các luật chắc chắn

{Sài Gòn} $\rightarrow$ *Không* *với S=1/6* u_4

{Nông dân} $\rightarrow$ *Không,* *với S=1/4* u_6

{Kĩ sư,Hà Nội} $\rightarrow$ *Cấm* *với S=1* u_2,u_7

- Các luật có thể

Công nhân $\rightarrow$ *Cấm* *với S = (1/4)(1/2)*

Mỹ&Công nhân $\rightarrow$ *Cấm* *với S = (1/2)(2/3)*

Mỹ&Hà Nội $\rightarrow$ *Cấm* *với S = (1/3)(2/3)*

Công nhân&Hà Nội $\rightarrow$ *Cấm* *với S = (1/2)(2/3)*

Các trường hợp bao phủ: u_1, u_3, u_5

b) Xét trường hợp có tỷ lệ nhiễu là > 0

U	*Tới nước*	*Nghề nghiệp*	*Nơi sinh*	*Xem xét*
$u_1' \begin{Bmatrix} u_1 \\ u_3 \\ u_5 \end{Bmatrix}$	Mỹ	Công nhân	Hà Nội	Cấm, Cấm, Không
u_2	Mỹ	Kĩ sư	Hà Nội	Cấm
u_3	Mỹ	Công nhân	Hà Nội	Cấm
u_4	Pháp	Kĩ sư	Sài Gòn	Không
u_5	Mỹ	Công nhân	Hà Nội	Không
u_6	Mỹ	Nông dân	Hà Nội	Không
u_7	Pháp	Kĩ sư	Hà Nội	Cấm

$\Downarrow$

U	*Tới nước*	*Nghề nghiệp*	*Nơi sinh*	*Xem xét*
u_1'	Mỹ	Công nhân	Hà Nội	Cấm
u_2	Mỹ	Kĩ sư	Hà Nội	Cấm
u_4	Pháp	Kĩ sư	Sài Gòn	Không
u_6	Mỹ	Nông dân	Hà Nội	Không
u_7	Pháp	Kĩ sư	Hà Nội	Cấm

Ta có : $r_{\{cấm\}}(u_1') = 1 - \frac{2}{3} = 0.33$ và $r_{\{không\}}(u_1') = 1 - \frac{1}{3} = 0.67$

Đặt $T_{nhiễu} = 0.5$ thì $r_{\{cấm\}}(u_1') = 0.33 < T_{nhiễu}$

như vậy là $d(u_1') = Cấm$

- Luật sinh ra từ tất cả các trường hợp

u_1': *{Công nhân}* $\rightarrow$ *Cấm, S=1/4*2/3=0.167*

u_2: *{Mỹ Kĩ sư}* $\rightarrow$ *Cấm, S=0.5*

{Kĩ sư Hà Nội} $\rightarrow$ *Cấm, S=1*

u_4: *{Sài Gòn}* $\rightarrow$ *Không, S=0.167*

u_6: *{Nông dân}* $\rightarrow$ *Không, S=0.25*

u_7: *{PhápHà Nội}* $\rightarrow$ *Cấm, S=0.5*

{Kí sưHN} $\rightarrow$ *Cấm, S=1*

Ví dụ nếu sử dụng tri thức kinh nghiệm từ bảng sau:

	Mỹ Công nhân Sài Gòn	*Mỹ Công nhân Hà Nội*	*Mỹ Kĩ sư Sài Gòn*	*Mỹ Kĩ sư Hà Nội*	*Mỹ Nông dân Sài Gòn*	*Mỹ Nông dân Hà Nội*	...	*Pháp Nông dân Hà Nội*
*Mỹ Công nhân **	1/2	1/2						
*Mỹ Kĩ sư**			1/2	1/2				
*Mỹ *Hà Nội*		1/3		1/3		1/3		
*Mỹ ***	1/6	1/6	1/6	1/6	1/6	1/6		

$f_T(u_2) =$ *(Tới nước* $\wedge$ *Nghề nghiệp)* $\vee$ *(Nghề nghiệp* $\wedge$ *Nơi sinh)*

{Mỹ Kĩ sư} *{Kĩ sư HàNội}*

{MỹKĩ sư} $\xleftarrow{1/2}$ *Mỹ Kĩ sư Sài Gòn*

{Mỹ Kĩ sư} $\xleftarrow{1/2}$ *Mỹ Kĩ sư Hà Nội*(u_2)

với S({Mỹ Kĩ sư})=0.5 và r({Mỹ Kĩ sư} $\rightarrow$ *Cấm) =0*

với tri thức là *Mỹ* $\Rightarrow$ *Hà Nội*, chắc chắn 100% ta sẽ có độ mạnh các luật thay đổi như sau:

{Mỹ Kĩ sư} $\xleftarrow{0\%}$ *Mỹ Kĩ sư Sài Gòn*

{Mỹ Kĩ sư} $\xleftarrow{100\%}$ *Mỹ Kĩ sư Sài Gòn* (u_2)

với S({Mỹ Kĩ sư})=1 và r({Mỹ Kĩ sư} $\rightarrow$ *Cấm) = 0*

Sự thay đổi độ mạnh của luật

$\Downarrow$

	Mỹ Công nhân Sài Gòn	*Mỹ Công nhân Hà Nội*	*Mỹ Kĩ sư Sài Gòn*	*Mỹ Kĩ sư Hà Nội*	*Mỹ Nông dân Sài Gòn*	*Mỹ Nông dân Hà Nội*	*...*	*Pháp Nông dân Hà Nội*
*Mỹ Công nhân **	0	1						
*Mỹ Kĩ sư**			0	1				
*Mỹ*Hà nội*		1/3		1/3		1/3		
*Mỹ * **	0	1/6	0	1/6	0	1/6		

II.2.1.4. Thuật toán tối ưu hoá các luật

Giả sử có bảng quyết định $\mathcal{A}$ = (U, A $\cup$ {d}) gồm n đối tượng và m thuộc tính, tỷ lệ nhiễu r. Câu hỏi đặt ra là tìm tập tối ưu các luật có cùng độ mạnh [9].

Bước 1: Các đối tượng với các giá trị thuộc tính điều kiện giống nhau được coi như một đối tượng gọi là *đối tượng ghép*.

Bước 2: Tính toán tỷ lệ nhiễu r cho mỗi đối tượng ghép.

Bước 3: Chọn một đối tượng u từ U và tạo một vector phân biệt được cho u.

Bước 4: Tìm tất cả các tập rút gọn cho đối tượng u sử dụng hàm phân biệt được.

Bước 5: Tạo các luật từ tập rút gọn cho u, và xem lại độ mạnh của mỗi luật.

Bước 6: Chọn luật tốt nhất từ các luật từ ***bước 5***, sử dụng phương pháp đánh giá kinh nghiệm khi lựa chọn luật.

Bước 7: U = U - {u}. Nếu U $\neq \varnothing$, thì quay lại ***bước 3***, trường hợp khác thì tiếp đến ***bước 8***.

Bước 8: Kết thúc nếu số các luật được chọn trong ***bước 6*** cho mỗi trường hợp là 1, trường hợp còn lại tìn một tập tối thiểu các luật mà chứa tất cả các trường hợp trong bảng quyết định.

Độ phức tạp thời gian của thuật toán:

O($mn^3 + mn^2N(G_T)$) với $N(G_T)$ là số lần sinh và nhỏ hơn $O(2^{m-1})$

Thuật toán này là có thể không phù hợp cho cơ sở dữ liệu mà số các thuộc tính là lớn. Để giải quyết vấn đề này các tác giả đã đưa ra phương pháp:

- Tìm kiếm tập rút gọn (tập con) của các thuộc tính điều kiện trong quá trình tiền xử lý
- Tìm giải pháp gần tối ưu sử dụng phương pháp tìm kiếm kinh nghiệm hiệu quả.

II.2.1.5. Thuật toán giải pháp gần tối ưu các luật

Các tác giả [9] đã đưa ra các bước thực hiện thuật toán gần tối ưu các luật như sau:

Bước 1: Đặt R = {}, COVERED = {} và SS = {*Định danh của tất cả các đối tượng*}. Với mỗi lớp D_c, chia bảng quyết định $\mathcal{A}$ thành 2 phần: Lớp hiện tại $\mathcal{A}_+$ và các lớp khác $\mathcal{A}_-$

Bước 2: Từ các giá trị thuộc tính v_{ij} của tất cả các đối tượng I_k (với v_{ij} là giá trị thứ j trong thuộc tính i, $I_k \in \mathcal{A}_+$, $I_k \in$ SS), chọn một giá trị v với số lần xuất hiện nhiều nhất trong các đối tượng trong $\mathcal{A}_+$ và ít nhất trong $\mathcal{A}_-$

Bước 3: Cập nhật giá trị v vào R

Bước 4: Xoá định danh đối tượng trong SS nếu đối tượng đó không chứa v.

Bước 5: Quay trở lại bước 2 cho đến khi tỷ lệ nhiễu nhỏ hơn giá trị r ban đầu.

Bước 6: Tìm tất cả tập con R' tối thiểu của R theo độ mạnh của chúng. Cập nhật (R' $\rightarrow$ D_c) vào RS. Đặt R = {}, chép các đối tượng trong SS vào COVERED và đặt SS = {Định danh của tất cả các đối tượng}\COVERED.

Bước 7: Quay lại ***bước 2*** cho đến khi tất cả các đối tượng của $\mathcal{A}_+$ ở trong COVERED.

Bước 8: Quay lại ***bước 1*** cho đến khi tất cả các lớp được xử lý.

Độ phức tạp của thuật toán này là : $O(m^2n^2)$

II.2.1.6. Tiêu chuẩn lựa chọn luật trong tập thô

Trong [9] cũng đưa ra một số tiêu chuẩn sau đây đối với việc lựa chọn luật:

- Chọn các luật mà bao phủ nhiều nhất có thể các trường hợp
- Chọn các luật mà có chứa ít nhất các thuộc tính có thể, nếu chúng bao phủ số các trường hợp giống nhau
- Chọn các luật với độ mạnh lớn, nếu chúng có giống nhau số các thuộc tính điều kiện và bao phủ số các trường hợp giống nhau.

II.2.2. Quá khám phá mẫu trong bảng quyết định

II.2.2.1. Khái niệm mẫu

Giả sử $\mathcal{A}$ = (U, A) là một hệ thông tin. Một *mẫu T* của $\mathcal{A}$ là công thức định đề bất kỳ $\Lambda(a_i = v_i)$ với $a_i \in$ A, $a_i \neq a_j$ với $i \neq j$, $v \in V_{a_i}$. Với A=$\{a_1,...,a_m\}$ có thể miêu tả bất kỳ mẫu như sau [5]:

$$T = (a_{i_1} = v_{i_1}) \wedge ... \wedge (a_{i_k} = v_{i_k})$$

được trình bày dưới dạng dãy $[x_1 ,..., x_m]$ mà tại vị trí p của dãy là v_p nếu $p = i_1 ,..., i_k$ còn tại vị trí còn lại là '*' (Tức là, $x_p = \begin{cases} v_p & p \in \{i_1,...,i_k\} \\ * & p \notin \{i_1,...,i_k\} \end{cases}$)

Một đối tượng x được gọi là thoả mãn $a = v$ (gọi a=v là *từ nhận dạng* hay *từ*) nếu $a(x) = v$ (đối tượng *x được gọi là* thoả mãn *mẫu* T nếu nó thoả mãn tất cả các *từ* trong mẫu).

Với mẫu T, ta định nghĩa *length*(T) - biểu thị số các *từ* khác nhau $a = v$ xuất hiện trong T và $fitness_{\mathcal{A}}$(T) - biểu thị *độ phù hợp* của mẫu, chính là số các đối tượng trong tập tổng thể U thoả mãn T. Nếu T gồm có một từ $a = v$ (chỉ cần viết $n_{\mathcal{A}}(a,v)$

hoặc n(a,v) thay vì $fitness_{\mathcal{A}}$(T)). *Độ đo chất lượng* của mẫu T được xác định bằng tích của độ phù hợp với số các từ khác nhau trong mẫu: $fitness_{\mathcal{A}}$(T) × *length*(T). Nếu s là một số nguyên thì $Template_{\mathcal{A}}(s)$ biểu diễn tập tất cả các mẫu của $\mathcal{A}$ với độ phù hợp là không nhỏ hơn s.

Ví dụ:

Giả sử $\mathcal{A}$ = (U,A ∪ {d}) là một bảng quyết định (Bảng 4)

T = (*Nơi sinh* = CHINA) ∧ (Tô*n giáo* = cao dai) ∧ (Đ*ến tới* = *101*) là một mẫu của $\mathcal{A}$ (T có thể biểu diễn là: [CHINA,*,cao dai,*, 101]) và các đối tượng x_1 và x_4 là phù hợp với T.

U	Thuộc tính điều kiện					Quyết định
	Nơi sinh	*Quốc tịch*	*Tôn giáo*	*Nghề nghiệp*	*đến tới*	*Xem xét*
x_1	**CHINA**	Trung quốc	**"cao dai"**	"Cong nhan"	**101**	1
x_2	TW	Đài loan	"cao dai"	"Ki su"	260	0
x_3	**CHINA**	Ma cao	"khong"	"Cong nhan"	260	1
x_4	CHINA	Trung quốc	"**cao dai**"	"Cong nhan"	**101**	1
x_5	HUE	Việt Nam	"cao dai"	"Giao vien"	103	0
Mẫu	**CHINA**	*	**"cao dai"**	*	**101**	

Bảng 4. Ví dụ về mẫu với *fitness* = 2 và *length* = 3

II.2.2.1. Hai bài toán mẫu cơ bản

Các tác giả Sinh Nguyen Hoa, Andrzej Skowron, Piotr Synak [5], đã đề xuất hai bài toán tìm kiếm mẫu. Bài toán thứ nhất là tìm kiếm mẫu với *độ phù hợp cực đại - maximal fitness* (độ dài mẫu cực đại - *maximal length*) với điều kiện *length (fitness)* nhỏ hơn hoặc bằng số L cho trước. Bài toán thứ hai là tìm kiếm mẫu với

độ chất lượng cực đại - maximal quality (sự kết hợp của *độ phù hợp* và *độ dài các từ khác nhau trong mẫu*).

a) Bài toán tìm mẫu với độ phù hợp cực đại

Mục tiêu chính của phần này là tập trung vào việc xem xét độ phức tạp tính toán của thuật toán tìm kiếm mẫu với *độ phù hợp cực đại*. Mẫu là ***L-tối ưu*** (*L-optimal*) nếu số các đối tượng phù hợp với nó là cực đại trong một tập các mẫu có độ dài mẫu bằng số L cho trước. Hai vấn đề đặt ra là bài toán quyết định mẫu có độ phức tạp tính toán NP đầy đủ và bài toán tối ưu là NP khó.

- *Bài toán quyết định mẫu được định nghĩa như sau*:

Bài toán mẫu phù hợp (Template Fitness Problem - TFP)

Giả thiết: Cho trước hệ thông tin $\mathcal{A}$ = (U, A) và hai số nguyên dương F, L

Câu hỏi: Có hay không mẫu T với độ dài mẫu bằng L và độ phù hợp không nhỏ thua F?

- *Bài toán tối ưu hoá tương ứng được định nghĩa như sau:*

Bài toán mẫu phù hợp tối ưu (Optimal Template Fitness Problem - OTFP)

Giả thiết: Cho trước hệ thông tin $\mathcal{A}$ = (U, A) và số nguyên dương L

Câu hỏi: Tìm một mẫu T với độ dài mẫu là L và độ phù hợp cực đại.

Các tác giả [5] xem xét một số ví dụ về bài toán NP đầy đủ điển hình để qua đó biểu diễn tính chất khó của bài toán mẫu với độ phù hợp (TFP).

- **Ví dụ 1*: Balanced Complete Bipartite Subgraph (BCBS)***

Giả thiết: Cho đồ thị G = ($V_1 \cup V_2$, E) được tách đôi đầy đủ, và số nguyên dương $K \le \min(|V_1|, |V_2|)$.

Câu hỏi: Liệu có tồn tại hai tập con $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ thoả mãn $|U_1| = |U_2| = K$ và $\{u,v\} \in E$ với mọi $u \in U_1$, $v \in U_2$?

BCBS là bài toán NP đầy đủ [10], các tác giả xem xét bài toán tiến của BCBS là CBS (**Complete Bipartite Subgraph**). Bài toán BCBS có độ phức tạp đa thức liên quan đến bài toán CBS, do đó tính NP-đầy đủ trong của bài toán CBS cũng chính là tính NP-đầy đủ của bài toán BCBS

- **Ví dụ 2*: Complete Bipartite Subgraph (CBS)***

Giả thiết: Cho đồ thị $G = (V_1 \cup V_2, E)$ được tách đôi đầy đủ, và hai số nguyên dương $K_1 \leq |V_1|$, $K_2 \leq |V_2|$.

Câu hỏi: Liệu có tồn tại hai tập con $U_1 \subseteq V_1$, $U_2 \subseteq V_2$ thoả mãn $|U_1| = K_1$, $|U_2| \geq K_2$ và $\{u,v\} \in E$ với mọi $u \in U_1$, $v \in U_2$?

- *Một số định lý và kết luận rút ra từ hai bài toán trên (Kết quả đã được chứng minh trong [5]):*

- **Định lý 2.1**: CBS là bài toán NP đầy đủ
- **Định lý 2.2**: TFP và CBS là tương đương theo độ phức tạp thời gian đa thức.
- **Kết luận 2.1**: *TFP là bài toán NP đầy đủ*
- **Định lý 2.3**: Nếu bài toán P ≠ NP thì OTFP là bài toán NP khó
- **Kết luận 2.2**: *Cho trước một bảng $\mathcal{A}$ = (U,A) và số nguyên dương F, L. Bài toán quyết định có tồn tại hay không một mẫu với độ phù hợp F và độ dài mẫu ít nhất L là bài toán NP đầy đủ.*
- **Kết luận 2.3**: *Cho trước một bảng $\mathcal{A}$ = (U,A) và số nguyên dương F. Bài toán tối ưu trong tìm kiếm mẫu T với độ phù hợp F và cực đại độ dài mẫu là bài toán NP khó.*

b) Bài toán tìm mẫu với độ chất lượng cực đại

Trong phần trước, luận văn đã đề cập đến độ phức tạp tính toán của thuật toán tìm kiếm mẫu tối ưu (ví dụ số các từ khác nhau mẫu phù hợp nhỏ hơn bằng một

số L cho trước với độ phù hợp cực đại). Chất lượng của mẫu có thể được xác định bằng tích giữa độ phù hợp với độ dài của mẫu hay có thể bằng tổng của độ phù hợp và độ dài của mẫu. Trong phần này, ta tập trung xem xét độ phức tạp tính toán của bài toán mẫu trong ngữ cảnh mới; mẫu là *tối ưu* nếu nó có độ chất lượng cực đại.

- *Bài toán tìm mẫu với chất lượng cực đại TQP (Template Quality Problem) được phát biểu như bài toán quyết định sau:*

Bài toán chất lượng mẫu (Template Quality Problem)

Giả thiết: Cho một hệ thông tin $\mathcal{A}$ = (U, A), với số nguyên K

Câu hỏi: Có tồn tại hay không một mẫu T trong $\mathcal{A}$ với độ đo chất lượng cao hơn K?

Giả sử bài toán TQP với độ đo chất lượng được xác định như sau (theo hàm cộng):

$$quality(T) = fitness(T) + length(T)$$

thì có thể được giải quyết trong thời gian đa thức. Tuy nhiên nếu chúng ta giả sử bài toán TQP với độ đo chất lượng được xác định như sau (theo hàm nhân):

$$quality(T) = fitness(T) \times length(T)$$

thì bài toán có độ phức tạp tính toán giống như bài toán NP đầy đủ, hiện vẫn là mở chưa được giải quyết.

- *Tối ưu hoá bài toán tìm mẫu với chất lượng cực đại OTQP (Optimal Template Quality Problem) được phát biểu như bài toán quyết định sau:*

Bài toán chất lượng mẫu tối ưu

Giả thiết: Thông tin hệ thống $\mathbf{A}$ = (U,A)

Câu hỏi: Tìm một mẫu T với độ đo chất lượng tốt nhất ($fitness(T) \times length(T)$ cực đại)

Trong [5] đưa ra phát biểu tương đương của bài toán OTQP hữu ích trong việc chứng minh tính chất NP-khó của nó.

Bài toán gán nhãn bản đồ (Labelled Subgraph Problem - LSP)

Input: Gán nhãn một cách không trực tiếp cho đồ thị G = (V,E,e) với hàm tô màu e: $E \to 2^X$ có các thuộc tính sau đây.

1. $\bigcup_{u,v \in V} e(u,v) = X$
2. $\forall_{u,v,w \in V} e(u,v) \cap e(v,w) \subseteq e(u,w)$

Output: Tìm $V' \subseteq V$ sao cho $|V'| \cdot \left| \bigcap_{u,v \in V'} e(u,v) \right|$ là cực đại.

Mệnh đề 2.2: Bài toán gán nhãn bản đồ (LSP) là tương đương đa thức với bài toán OTQP (đã được chứng minh trong [5]).

II.2.2.1. Các phương pháp sinh mẫu

Phần này tập trung xem xét một số phương pháp đánh giá kinh nghiệm để sinh mẫu gần tối ưu từ dữ liệu sử dụng thuộc tính quyết định trong bảng quyết định [5].

a) Tìm kiếm mẫu sử dụng trọng số

- ***Thuật toán trọng số đối tượng***

Ý tưởng của phương pháp này dựa trên quan sát rằng bất kỳ tập đối tượng $U_1 \subseteq U$ được sinh ra bởi tập $T(U_1)$ của các mẫu phù hợp với tất cả các đối tượng trong U_1. Giả sử T_{U_1} biểu thị mẫu với số độ dài mẫu cực đại trong các mẫu thuộc $T(U_1)$. Ta định nghĩa *độ đo chất lượng cục bộ* của mẫu T_{U_1} là tích giữa các yếu tố trong tập U_1 với số độ dài mẫu T_{U_1} ($card(U_1)$ x $length(U_1)$). T_{U_1} được gọi là *độ đo*

chất lượng cục bộ tối ưu (*local optimal*) nếu độ đo chất lượng cục bộ của nó là cực đại. Mục tiêu của phương pháp này là tìm một tập hợp con U_1 mà mẫu T_{U_1} được sinh ra bởi U_1 là tối ưu hoá cục bộ. Tập đối tượng U_1 được sinh ra bởi một mẫu có độ chất lượng cao nếu các đối tượng trong tập U_1 là tương tự nhau. Để thoả mãn mục đích này, ta tính toán trên mọi đối tượng trong hệ thông tin. Sử dụng thuật toán "tham lam" để ước tính đối tượng trong tập U_1. Bắt đầu từ tập rỗng $U_1 = \varnothing$, với mỗi đối tượng ta chọn ngẫu nhiên một trọng số và gắn vào tập U_1. Với một tập hợp mới U_1 mẫu T_{U_1} và độ đo chất lượng cục bộ của nó được tính toán. Nếu độ đo chất lượng của T_{U_1} là tốt hơn thì thuật toán tiếp tục, ngược lại sự quyết định phụ thuộc vào giá trị của biến điều khiển. Thuật toán sử dụng một kỹ thuật gọi là "mutation - sự hoán chuyển", một vài đối tượng được chọn sẽ bị xoá tại mỗi bước. Điều này giải quyết vấn đề giá trị lặp vô hạn. Dưới đây đưa ra một vài độ đo tương tự hữu ích trong mô tả trọng số đối tượng.

+ *Trọng số đối tượng phản ánh sự tương tự của các đối tượng*

Đặt $\mathcal{A}$ = (U,A) và x ∈ U, cho bất kỳ y ∈ U nào ta có:

$$g_{x,y} = \left| \{a \in A : a(x) = a(y)\} \right|$$

Số các thuộc tính mà có các giá trị tương đương *x* và *y*. Số này phản ánh "Tính chặt" của *y* tới *x*, bất kỳ thuộc tính $a \in A$ nào chúng ta có:

$$w_a(x) = \sum_{y:a(x)=a(y)} g_{x,y}$$

và cuối cùng trọng số:

$$w(x) = \sum_{a \in A} w_{a(x)}$$

ta có

$$w(x) = \sum_{y} g_{x,y}^2$$

+ *Trọng số đối tượng xuất phát từ giá trị thuộc tính thường xuất hiện*

Đặt $\mathcal{A}$ = (U,A) và $x \in$ U, cho bất kỳ $a \in$ A nào ta định nghĩa:

$$w_a(x) = n_{\mathcal{A}}(a,a(x)) \text{ và } w(x) = \sum_{a \in A} w_{a(x)}$$

Các thử nghiệm cho thấy những trọng số được kể trên hoàn toàn thoả mãn nhóm các đối tượng trong một mẫu trong khi nhiều giá trị "naive" của trọng số làm giảm bớt chất lượng của kết quả.

- ***Thuật toán trọng số thuộc tính***

Ý tưởng của phương pháp này rất giống với phương pháp "trọng số đối tượng", tuy nhiên các trọng số thích hợp sẽ được gắn kèm với tất cả các thuộc tính trong bảng quyết định. Với các thuộc tính mỗi giá trị của nó cũng chứa đựng một trọng số. Trong quá trình tìm kiếm mẫu, đầu tiên thuộc tính và giá trị của nó được chọn ngẫu nhiên đối với từng trọng số. Mỗi lần một thuộc tính mới và một giá trị thuộc tính được chọn, người ta tính toán độ phù hợp (fitness) của mẫu tìm được. Nếu tìm thấy một mẫu mới tốt hơn thì thuật toán tiếp tục, ngược lại thì phụ thuộc vào biến điều khiển. Thuật toán sử dụng kỹ thuật gọi là "sự hoán chuyển". Nó cho phép ta tránh được giá trị lặp vô hạn (local extrema).

Algorithm (Attribute Weight)

1. Initialize T = [*,*,....,*];
2. i = 1; k = 1; *fitness* = 0;
3. **while** điều kiện không thoả mãn

(a) Chọn ngẫu nhiên $r \in [0,1]$;

(b) **If** ($r < w_A(a_i)$ **and** T[i] = *) **then**

Chọn một số nguyên dương $l \in \{1,...,|V_{a_i}|\}$ mà

$$\sum_{k=1}^{l-1} w_{\mathcal{A}}^{a_i}(v_k^{a_i}) \le r \le \sum_{k=1}^{l} w_{\mathcal{A}}^{a_i}(v_k^{a_i});$$

T[i] = $v_l^{a_i}$;

Tính toán độ phù hợp mới (*new_fitness*) cho T;

if *new_fitness* ≤ *fitness* x *fit_coeff* **then**

T[i] = *;

else

fitness = *new_fitness;* Store(T);

end if;

(c) **If** k = *mutation_coeff* **then**

Đổi giá trị chọn ngẫu nhiên cho mẫu;

k = 0;

end if;

(d) i = i+1; k = k+1;

(e) **if** i=n **end if**; i=1;

end while

Đặt $\mathcal{A}$ = (U,A), $m = |U|$, $n = |A|$, có thể sắp xếp giá trị thuộc tính của $a \in A$ theo giá trị $n_{\mathcal{A}}(a,v)$ cho bất kỳ $a \in A$ nào, sau đó với v_i^a chúng ta biểu diễn giá trị thứ i của thuộc tính a bởi thứ tự sắp xếp. Giá trị v_i^a thường xuất hiện nhất trong A, chọn ngẫu nhiên thứ tự giữa giá trị v và u nếu $n_{\mathcal{A}}(a,v) = n_{\mathcal{A}}(a,u)$. Với bất kỳ thuộc tính $a \in A$ nào ta có:

$$w_{\mathcal{A}}(a) = \frac{m}{\sum_{i=1}^{|V_a|} i \bullet n_{\mathcal{A}}(a, v_i^a)}$$

$w_{\mathcal{A}}(a) \in (0,1]$ cho bất kỳ giá trị u của thuộc tính a, chúng ta định nghĩa trọng số của u như sau:

$$w^2_{\mathcal{A}}(u) = \frac{n_{\mathcal{A}}(a,u)}{m}$$

Ta có $w^2_{\mathcal{A}}(u) \in (0,1]$ và $\sum_{v \in V_a} w^a_{\mathcal{A}}(v) = 1$ cho bất kỳ $a \in$ A.

Người ta có thể quan tâm đến việc tìm kiếm mẫu với độ phù hợp nhỏ hơn nhưng với nhiều giá trị thuộc tính cố định. Trong trường hợp này mẫu ban đầu có thể được xác định như ở trong 3.a đến 3.e. Trong những trường hợp khác nhân tố quan trọng nhất có thể là chất lượng của mẫu mà không lưu tâm đến độ dài của mẫu. Liên hệ với điều này, mẫu ban đầu có thể được đặt bởi một giá trị bất kỳ. Fitness_coeff và Mutation_coeff phải được chọn qua thực nghiệm. Chúng cho phép ta thu được những kiểu mẫu khác nhau với số thuộc tính cố định thay đổi.

b) Sử dụng phương pháp Max (cực đại hoá) để lấy mẫu

Algorithm (Max I)
Input: 1 hệ thống thông tin $\mathcal{A}$ = (U,A) với $n = |U|$, $m = |A|$ và một số nguyên dương s.
Output: Một mẫu T lấy ra từ $Template_{\mathcal{A}}(s)$ với số các từ khác nhau nửa cực đại
Begin

1. T = ∅;
2. **while** ($length(T) < m$ **and** $fitness_{\mathcal{A}}(T) > s$ **do**
   (a) **for** a ∈ A
   Sắp xếp các đối tượng từ U đối với giá trị của a;
   Xác định giá trị v_a mà $n_{\mathcal{A}}(a,v_a) = \max_{v \in V_a}\{n_{\mathcal{A}}(a,v)\}$;
   endfor
   (b) Chọn $a = v_a$ mà $n_{\mathcal{A}}(a,v_a) = \max_{b \in A \setminus A(T)}\{n_{\mathcal{A}}(b,v)\}$ với A(T) là các thuộc tính xuất hiện trong T;
   (c) U = tập các đối tượng từ U phù hợp mẫu $a = v_a$;
   (d) A = A\{a}; T = T ∪ { $a = v_a$ };
   endwhile

End

Mục đích của phương pháp này là tìm kiếm mẫu dài nhất có thể với hệ số phù hợp không nhỏ hơn số *s*. Các tác giả đã đề xuất ra một phương pháp tìm kiếm kinh nghiệm gọi là "*Max Method*", thuật toán bắt đầu với mẫu rỗng tức là mẫu với độ dài=0. Mẫu mở rộng bằng cách thêm vào liên tục các từ của $a = v_a$ cho đến khi hệ số phù hợp của mẫu không nhỏ hơn giá trị cố định *s*. Nếu mẫu T hiện tại gồm có *i*-1 biến và sau đó từ thứ *i* được chọn như sau:

Tìm trong các thuộc tính không xuất hiện trong mẫu T với một thuộc tính *a* và *a* phù hợp với giá trị v_a giống như độ phù hợp của mẫu mới $T \cup (a=v_a)$ là cực đại. Việc xây dựng mẫu có thể được thực hiện một cách hiệu quả như sau:

Đặt T là mẫu với *i*-1 biến và $\mathcal{A}_{i-1} = (U_{i-1}, A_{i-1})$ với U_{i-1} là tập các đối tượng thoả mãn trong T, A_{i-1} bao gồm tất cả các thuộc tính từ A không xuất hiện trong mẫu. Thuật toán sắp xếp các đối tượng trong U_{i-1} theo giá trị của thuộc tính. Giữa các giá trị đã được sắp xếp của tất cả các thuộc tính nó chọn thuộc tính *a* và giá trị *v* với hệ số phù hợp cực đại $fitness_{\mathcal{A}_{i-1}}(a = v)$.

Thuật toán cho phép xây dựng mẫu lớn một cách hiệu quả nhưng nó chỉ sinh ra được một mẫu. Các tác giả đã giới thiệu một thuật toán cải tiến của thuật toán MaxI cho phép tìm được nhiều hơn một mẫu tốt. Thay vì chọn từ với sự phù hợp lớn nhất chúng ta sẽ quan tâm đến tất cả các từ được tạo trong bước 2.a và chọn ngẫu nhiên một từ trong số đó theo xác suất chắc chắn. Sau đó từ được chọn $a = v_a$ sẽ được thêm vào mẫu với xác suất:

$$P(a = v_a) = \frac{n_{\mathcal{A}}(a, v_a)}{\sum_{v \in V_a} n_{\mathcal{A}}(a, v)}$$

Thuật toán cải tiến MaxI như sau:

Algorithm (Max II)
T = ∅;
while (*length*(T) < *m* **and** $fitness_{\mathcal{A}}$(T) < *s* **do**
for *a* ∈ A
Sắp xếp các đối tượng từ U đối với giá trị của *a*;
Xác định giá trị v_a mà $n_{\mathcal{A}}(a,v_a) = \max_{v \in V_a}\{n_{\mathcal{A}}(a,v)\}$;
endfor
Chọn ngẫu nhiên từ $a = v_a$ với xác suất

$$P(a = v_a) = \frac{n_{\mathcal{A}}(a,v_a)}{\sum_{v \in V_a} n_{\mathcal{A}}(a,v)}$$

U = tập các đối tượng từ U phù hợp mẫu $a = v_a$;
A = A\{*a*}; T = T ∪ { *a* = *v* };
endwhile

Cả hai thuật toán MaxI và MaxII đều có thời gian thực hiện là $O(m^2 nlogn)$ trong trường hợp xấu nhất.

c) Tìm kiếm mẫu sử dụng thuật toán di truyền.

Thuật toán di truyền là một lớp các siêu tìm kiếm theo kinh nghiệm dựa trên giải thuật di truyền (Thuyết tiến hoá). Thuật toán dựa trên một chuỗi các bước đơn giản sau đây:

Bước 1: Lấy một đối tượng x_0 như là một đối tượng cơ sở

Bước 2: Đặt ∂ là phép hoán vị của các thuộc tính.

Bước 3: Coi như a là tập các mẫu của form: $T_1 = (a_{\partial 1} = v_{\partial 1})$; $T_2 = (a_{\partial 1} = v_{\partial 1}) \wedge (a_{\partial 2} = v_{\partial 2})$, .., v_i biểu thị 1 giá trị *i*-th thuộc tính trên $x_{0.}$

Bước 4: Chọn mẫu tốt nhất giữa T_1,. . ., T_n. Đây là kết quả được sinh ra bởi phép hoán vị ∂.

Đây là phương pháp đánh giá kinh nghiệm đơn giản để sinh ra các mẫu tốt. Tuy nhiên, kết quả phụ thuộc vào đối tượng cơ sở x_0 và phép hoán vị ∂. Đối tượng x_0

được chọn ngẫu nhiên, ngược lại phép hoán vị tối ưu được sinh ra bởi giải thuật di truyền tiến hoá (order-based). Một hàm phù hợp của phép hoán vị ∂ tương ứng với giá trị của mẫu tốt nhất được sinh ra bởi ∂.

d) Các mẫu suy rộng

Với ý tưởng một mẫu có thể được mở rộng gọi là các *mẫu suy rộng*.

$$GT = (a_{i_1} = v_{i_1} \vee ... \vee a_{i_1} = v_{i_n}) \wedge ... \wedge (a_{j_k} = v_{j_1} \vee ... \vee a_{j_k} = v_{j_m}).$$

Sự khác biệt chính ở đây là thay vì một giá trị chúng ta có nhiều giá trị thế của GT. Chúng ta nói rằng một đối tượng x thoả mãn từ suy rộng $a = v_1 \vee ... \vee a = v_m$ nếu giá trị của a trên x thuộc vào tập $\{v_1, ... ,v_m\}$. Một đối tượng x thoả mãn mẫu suy rộng GT nếu nó thoả mãn tất cả các từ trong GT. Trường hợp mở rộng của ý tưởng này có thể thu được bởi mẫu với các từ không riêng rẽ.

$$a \in [v_{i_1}, v_{i_2}] \vee ... \vee a \in [v_{m_1}, v_{m_2}]$$

Đối với mẫu suy rộng GT có thể thay đổi độ dài của một từ trong GT bởi công thức sau:

$$l(a) = \begin{cases} 1/k & \text{nếu } a \text{ xuất hiện trong mẫu} \\ 0 & \text{trong các trường hợp khác} \end{cases}$$

Cho bất kỳ $a \in$ A, số k bằng số các từ khác nhau (*length*) của từ suy rộng a. Độ chất lượng của từ suy rộng a là tích số giữa $l(a)$ và số các đối tượng thoả mãn. Sử dụng chức năng l có thể dễ dàng sửa chữa sự phù hợp (*fitness*) và số các từ khác nhau (*length*) của mẫu suy rộng. Trong đó $fitness_{\mathbf{A}}$ (GT) của GT được hiểu là số các đối tượng thoả mãn GT và số các từ khác nhau của GT:

$$length(GT) = \sum_{a \in A} l(a)$$

Độ chất lượng của mẫu GT được tính là $fitness_{\mathbf{A}}$(GT) x *length*(GT).

Một trong những chiến lược đơn giản nhất là cải tiến thuật toán Max. Cho bất kỳ thuộc tính $a \in A$ thay vì tìm kiếm một giá trị phù hợp với số lượng tối đa các đối tượng được rút ra trong tập giá trị S_a thì độ chất lượng của từ mở rộng được định nghĩa bởi a và giá trị từ S_a là cực đại. Tập S_a được chọn từ lớp con tuần tự từ danh sách được sắp xếp tất cả các giá trị V_a được định nghĩa trên a. Tập con tuần tự S_a là tối ưu nếu độ đo chất lượng của từ $V\{a = v : v \in S_a\}$ là cực đại. Bắt đầu từ mẫu rỗng GT = ∅, giản đồ mô tả quá trình sinh GT như sau:

Bước 1: Cho bất kỳ thuộc tính $a \in A$ tính toán tối ưu tập S_a.

Bước 2: Chọn 1 thuộc tính a và tương ứng với tập giá trị S_a như vậy độ đo chất lượng của từ $p = V\{a = v : v \in S_a\}$ là cực đại.

Bước 3: Thêm từ p vào GT; Loại bỏ a trong A. Tính toán độ đo chất lượng của GT.

Bước 4: Lặp lại bước 1 đến 3 cho đến khi A rỗng.

Bước 5: Trong các mẫu được sinh ra chọn một mẫu tốt nhất chính là mẫu có độ đo chất lượng cực đại.

II.2.3. Mối liên hệ giữa mẫu và luật theo tiếp cận tập thô

Trong quá trình khám phá tri thức, một trong những mục tiêu chính của việc phân tích dữ liệu theo cách tiếp cận tập thô là tìm ra những mẫu hay luật từ dữ liệu (các dữ liệu này được biểu diễn dưới dạng hệ thông tin hay bảng quyết định). Bảng quyết định $\mathcal{A} = (U, A \cup \{d\})$ là một kiểu đặc biệt của hệ thông tin $\mathcal{A} = (U,A)$. Như vậy, luật quyết định là một kiểu đặc biệt của mẫu [3,5,6]. Một tập các mẫu giống như một tập luật trong trường hợp tập luật đó không chứa kết quả. Mẫu là kết quả của việc tính toán trên tập rút gọn khi người ta không quan tâm

đến thuộc tính quyết định. Luật quyết định phản ánh một quan hệ, hay một xác xuất có thể giữa tập thuộc tính điều kiện và tập thuộc tính quyết định.

Với mẫu người ta sử dụng các độ đo là độ phù hợp $fitness_{\mathcal{A}}(T)$ biểu thị số các đối tượng trong tập tổng thể phù hợp với mẫu T và độ chất lượng $quanlity_{\mathcal{A}}(T) = fitness_{\mathcal{A}}(T) \times length(T)$ (tích của độ phù hợp với số các từ khác nhau trong mẫu) biểu thị chất lượng của mẫu tìm được. Còn với luật, người ta sử dụng độ mạnh để biểu thị số các đối tượng thoả mãn bộ sinh luật và độ nhiễu để biểu thị độ mạnh của luật khi xử lý loại dữ liệu có nhiễu.

II.3. SO SÁNH LUẬT THEO TIẾP CẬN TẬP THÔ VÀ LUẬT KẾT HỢP

Việc khai phá luật kết hợp từ CSDL nhằm mục đích tìm ra mỗi quan hệ giữa các thuộc tính (các thuộc tính đó có thể hoàn toàn độc lập với nhau trong bảng dữ liệu). Kết quả đưa ra trong quá trình phân tích luật kết hợp là những luật kết hợp được biểu diễn dưới dạng ngôn ngữ tự nhiên hoặc một câu lệnh trong ngôn ngữ hỏi có cấu trúc như SQL. Biểu diễn các mẫu dữ liệu thành những luật dạng "nếu ... thì... " làm cho luật dễ hiểu và việc áp dụng chúng dễ dàng. Thêm vào đó luật kết hợp còn hỗ trợ việc tìm kiếm dữ liệu không trực tiếp, dữ liệu có kích thước thay đổi và đưa ra những luật với kết quả khá sáng sủa, rõ ràng và không làm mất thông tin. Các tính toán cần thiết để áp dụng phân tích luật kết hợp cũng khá đơn giản mặc dù số lượng tính toán tăng nhanh cùng với số lượng của các giao tác và số lượng các mục (item) khác nhau trong quá trình phân tích. Tuy nhiên quá trình khai phá luật kết hợp từ CSDL gặp phải một số vấn đề như sau:

- Độ phức tạp tính toán lại tỷ lệ theo hàm mũ đối với kích thước của bảng dữ liệu: Người ta đã đưa ra giải pháp để làm giảm độ phức tạp tính toán là giảm bớt số lượng các mục bằng cách sinh ra các lớp mục chung, nhưng phương pháp này rất có thể sẽ làm mất đi những luật quan trọng.

- Việc hỗ trợ các thuộc tính cũng bị giới hạn
- Khó khăn trong việc xác định chính xác số lượng các mục: Thông thường, vấn đề khó khăn nhất trong việc áp dụng luật kết hợp là xác định đúng đắn tập các mục để sử dụng cho việc phân tích. Bằng cách tổng quát hoá các mục thành các lớp thì có thể đảm bảo được tần xuất xuất hiện của các mục sử dụng để phân tích là như nhau mặc dù quá trình khái quát hoá này làm mất một số thông tin, các mục ảo có thể được thêm vào trong qua trình phân tích để lấy lại những thông tin tiềm ẩn trong các mục được tổng quát.
- Vấn đề đối với các mục ít xuất hiện trong cơ sở dữ liệu: Quá trình khai phá luật chỉ làm việc tốt nhất khi các mục có tần xuất xuất hiện gần giống nhau trong dữ liệu. Các mục ít xuất hiện, thường là trong một số ít giao tác sẽ bị xén bớt. Có thể điều chỉnh để các giá trị mục quan trọng được giữ lại bằng cách điều chỉnh ngưỡng của độ hỗ trợ tối thiểu.

Lý thuyết tập thô được phát triển bởi Pawlak cho phép suy dẫn ra các tập xấp xỉ của khái niệm. Nó cung cấp những công cụ toán học giúp rút gọn dữ liệu trong quá trình tìm kiếm mẫu dữ liệu ẩn và sinh luật. Nó có thể được sử dụng cho việc lựa chọn các đặc trưng, rút ra các đặc trưng, rút gọn dữ liệu, sinh luật quyết định và mẫu. Lý thuyết này được sử dụng trong việc phát hiện luật từ dữ liệu dạng bảng quyết định với những loại dữ liệu nhiễu, dữ liệu liên tục (được rời rạc hoá), dữ liệu không hoàn hảo nhằm biểu thị mối quan hệ giữa thuộc tính điều kiện và thuộc tính quyết định. Việc sử dụng tri thức nền một cách tự nhiên trong chọn luật cũng giảm bớt được số thuộc tính cần xem xét để tạo luật một cách hiệu quả. Cách tiếp cận tập thô đã được chứng minh là một công cụ rất hữu ích để giải quyết các vấn đề trong việc phân tích quyết định thông thường là phân tích những quyết định đa mục tiêu.

Trong quá trình khai phá luật kết hợp người ta sử dụng các bảng biểu để biểu diễn dữ liệu còn trong tập thô người ta sử dụng hệ thông tin (bảng quyết định) để biểu diễn dữ liệu. Trong khai phá luật theo cách tiếp cận thông thường người ta sử dụng độ tin cậy để biểu thị sự phù hợp của các đối tượng đối với luật được phát hiện thì trong khai phá luật theo tiếp cận tập thô người ta sử dụng độ mạnh để biểu thị số các trường hợp mà luật phát hiện bao phủ.

II.4. KẾT LUẬN CHƯƠNG II

Trong chương này luận văn trình bày về quá trình khám phá luật theo cách tiếp cận truyền thống theo ý tưởng của Rakesh Agrawal (mục II.1), và phát hiện luật, mẫu từ dữ liệu theo tiếp cận tập thô, trong đó đưa ra quá trình khám phá luật từ bảng quyết định (mục II.2.1) và quá trình khám phá mẫu từ bảng quyết định (mục II.2.2). Từ đó đưa ra mỗi liên hệ giữa mẫu và luật trong lý thuyết tập thô.

Mục tiêu của chúng tôi trong chương này là tìm ra một số nhận xét đối sánh luật kết hợp theo thông thường và luật kết hợp cận tập thô (mục II.3) trong đó chú trọng đến việc đưa ra những so sánh ở mức khái niệm của việc khám phá luật từ dữ liệu theo hai cách tiếp cận. Tuy đây là hai cách tiếp cận khác nhau nhưng chúng đều dựa trên một mục tiêu cơ bản đó là tìm ra mối quan hệ giữa các thuộc tính trong bảng dữ liệu.

CHƯƠNG 3. ỨNG DỤNG CỦA MẪU VÀ THỬNGHIỆM QUÁ TRÌNH KHÁM PHÁ LUẬT THEO TIẾP CẬN TẬP THÔ

III.1. ỨNG DỤNG MẪU

III.1.1. Mẫu và quá trình phân loại ban đầu

Mẫu quyết định hữu ích trong quá trình phân lớp ban đầu nhanh các đối tượng mới. Nếu một đối tượng phù hợp với một trong số các mẫu đã được sinh ra cho lớp quyết định *C*, ta có thể cho rằng đối tượng đó phù hợp với lớp *C*. Ví dụ sau đây [5] thể hiện rằng trong nhiều trường hợp thông tin ẩn trong các mẫu là đủ cho sự phân lớp.

Cơ sở dữ liệu thử nghiệm: Dữ liệu ảnh từ vệ tinh (gồm có 4435 đối tượng dùng cho việc huấn luyện, 2000 đối tượng dùng cho việc kiểm tra, mỗi đối tượng được mô tả bởi 36 thuộc tính). Thời gian huấn luyện là: 1203 giây, sự phân lớp các đối tượng kiểm tra được thực hiện trong 12 giây, kết quả như sau:

- 37% số các đối tượng kiểm tra được phân loại đúng
- 6% số các đối tượng bị phân loại sai
- 2% số đối tượng được phân vào nhiều hơn một lớp quyết định
- 52% số đối tượng không được phân loại
- 99.97% các đối tượng huấn luyện được phân loại đúng.

Do tỉ lệ các đối tượng không phân loại được cao nên kĩ thuật này không được sử dụng để phân chia lớp. Tuy nhiên, đối với các đối tượng đã được huấn luyện thì tỉ lệ nhận biết được các đối tượng là cao và thời gian huấn luyện ngắn (so sánh với các hệ chuyên gia khác) do đó kĩ thuật này thường được sử dụng kết hợp với các kỹ thuật khác. Lý do gây nên việc kỹ thuật này có tỷ lệ các đối tượng không phân loại được cao liên quan đến chất lượng mẫu. Để việc phân loại các đối tượng

mềm dẻo hơn, người ta đưa ra một ý tưởng mới về độ đo tương tự của đối tượng đối với một mẫu. Độ đo tương tự của giá trị thuộc tính là một hàm d(v_i, v_j), nhận các giá trị giữa 0 và 1 (1 - giá trị bằng hay gần bằng, 0 - giá trị hoàn toàn khác). Ví dụ

$$d(v_1, v_2) = \frac{|v_1 - v_2|}{|v_{\max} - v_{\min}|}$$

với v_{max} và v_{min} là các giá trị cực đại và cực tiểu của thuộc tính. Hàm biểu thị giá trị tương tự có thể có các dạng phức tạp hơn (số mũ, rời rạc, không hoàn chỉnh) và có thể khác nhau cho mỗi thuộc tính.

Giả sử độ đo số tương tự d_i: V_i x $V_i \rightarrow$ [0,1] xác định trên các giá trị của tất cả các thuộc tính a_i. Đặt *D(x,T)* là độ đo tương tự của một đối tượng *x* cho một mẫu T, thì *D(x,T)* được xác định như sau:

$$D(x,T) = \prod_{i:v_i \neq "*"} d_i(a_i(x), v_i)^{p_i}$$

với v_i là giá trị của thuộc tính thứ *i* trong mẫu T, và p_i là tham số chính xác kết hợp với giá trị v_i của thuộc tính a_i trong mẫu T.

Độ đo tương tự *D* nhận giá trị từ [0,1], với một đối tượng mới *x*, ta có thể tính toán giá trị *D(x,T)* cho bất kỳ mẫu nào trong tập bao phủ, sau đó tìm mẫu gần nhất và lớp quyết định kết hợp với nó. Đối tượng mới *x* được phân loại thuộc về lớp quyết định này.

Ý tưởng của phương pháp tìm độ đo tương tự của một đối tượng đối với một mẫu rất hữu ích khi mô tả các đối tượng không hoàn hảo (khi giá trị của một vài thuộc tính của đối tượng đó bị thiếu). Tỷ lệ tương tự của các trường trống và giá trị các thuộc tính trong mẫu có thể được đặt là hằng số hoặc phụ thuộc vào phân bổ xác suất của các giá trị trong CSDL huấn luyện [9].

III.1.2. Mô tả các lớp quyết định

Giả sử có $\mathcal{A}$ = (U, A ∪ {d}), với $d \notin$ A là thuộc tính quyết định, ta xem xét sự mô tả lớp quyết định thứ i bởi tập các luật quyết định (thuật toán quyết định trong lớp này).

Khả năng để tìm kiếm tập mẫu bao phủ lớp quyết định mà phần lớn các đối tượng trong lớp phù hợp với một trong các mẫu trong khi có ít nhất các đối tượng từ các lớp khác có thể phù hợp với các mẫu đó. Thuật toán sinh mẫu có thể được làm thích nghi cho một kiểu mẫu mới: Người ta có thể thay đổi công thức tính sự phù hợp mẫu (phần II.2.2.2). Các bước như sau:

Bước 1: Đưa ra một tập các mẫu

Bước 2: Đưa các mẫu thu được từ bước 1 vào nhóm và ghép vào quá trình hoạt động của việc mở rộng và/hoặc thu nhỏ nhóm. Nhóm đưa ra được thực hiện sau khi chọn mẫu. Trong bước này tiến hành các bước nhỏ sau:

(i) Hai mẫu bao phủ các đối tượng gần giống nhau trong lớp và tách biệt nhau nên được chia ra thành hai nhóm khác nhau sử dụng các thủ tục nhóm.

(ii) Họ các phần giao của các mẫu khác nhau trong một nhóm nên không bao hàm "Close" trong việc phân hoạch của lớp quyết định thành một nhóm các tập thành phần. Nhóm các mẫu nhận được là kết quả của các thủ tục này. Các lớp bao phủ xấp xỉ khác nhau của lớp quyết định xây dựng bởi việc mở rộng các nhóm này. Các nhóm đưa ra được thực hiện tiếp tục như một tiền xử lý cho việc xây dựng. Quá trình được tiếp tục cho đến khi mô tả của lớp quyết định với chất lượng thích đáng được hình thành. Trong các trường hợp khác, việc xây dựng được đánh giá là chưa thành công và sẽ được làm lại từ một vài mức trước đó bởi nhóm khác hoặc chiến lược xây dựng khác. Toán tử suy rộng có thể không hiểu được trong trường hợp đơn giản nhất ví dụ như hợp của các đối tượng thoả mãn một trong các mẫu.

Lặp lại bước 2 cho đến khi độ đo chất lượng rút ra từ thuật toán quyết định là đủ tốt.

Bước 3: Nếu độ đo chất lượng của thuật toán chưa thoả mãn thì lặp lại bước một hoặc chúng ta có thể sử dụng thuật toán như việc xác định xấp xỉ của lớp quyết định thứ *i*.

Chất lượng của thuật toán quyết định rút ra bởi phương pháp này phụ thuộc vào việc nó phù hợp như thế nào với lớp quyết định và sự phức tạp của nó. Người ta nhắm tới việc sản sinh ra các luật với mô tả đơn giản nhất có thể.

III.1.3. Mẫu và bài toán phân tách bảng dữ liệu lớn

Ý tưởng chính của phương pháp này là tìm ra phương pháp phân chia các bảng dữ liệu lớn thành các bảng con có kích thước có thể thực hiện được. Điều đó có nghĩa là các bảng con không nên có kích thước quá lớn và phải được phân tích bởi thuật toán đang tồn tại. Đồng thời, các bảng đó không nên quá nhỏ để đảm bảo chắc chắn rằng các luật quyết định rút ra từ chúng là đủ tổng quát. Trong quá trình phân tách ta cố gắng giảm tối thiểu số các bảng con được sinh ra. Thêm vào đó, các bảng được sinh ra nên có kích thước tương đối đều nhau.

a) Phân tách cây nhị phân

Giả sử có $\mathcal{A} = (U, A \cup \{d\})$, với $d \notin A$ là thuộc tính quyết định, các bước thực hiện phân tích bảng dữ liệu $\mathcal{A}$ tiến hành tuần tự như sau:

Bước 1: Tìm một mẫu T tốt nhất trong $\mathcal{A}$

Bước 2: Chia $\mathcal{A}$ thành hai bảng con: $\mathcal{A}(T)$ chứa tất cả các đối tượng thoả mãn T, và $\mathcal{A}(\neg T) = \mathcal{A} - \mathcal{A}(T)$.

Bước 3: Nếu đã thu được bảng con có kích thước đạt yêu cầu thì dừng lại, nếu không thì lặp lại bước 1 đến 3 cho tất cả các bảng con có kích thước lớn mới thu được.

Bước 4: Tìm luật quyết định cho các bảng con mới thu được.

Thuật toán sinh ra một cây nhị phân của các bảng con, với tập luật quyết định tương đương với mỗi bảng con là các lá của cây nhị phân.

b) Phân tách bởi tập con bao phủ tối thiểu

Ý tưởng của phương pháp này là việc phân chia bảng lớn bởi một số tập tối ưu các bảng con bao phủ toàn bộ (hoặc là phần dữ liệu chính) của bảng dữ liệu cũ. Tập tối ưu bao phủ có thể được xác định bởi một số chiến lược khác nhau. Tuy nhiên trong phần này ta chỉ quan tâm đến việc xác định tập tối ưu bao phủ bởi các yếu tố tối thiểu.

Xem xét tất cả các đối tượng và xác định một số mẫu tốt nhất (mẫu phù hợp với các đối tượng này và có độ chất lượng cực đại). Bất kỳ đối tượng $u \in U$ nào cũng có thể được coi như một *bộ sinh* ra các bảng con của các đối tượng tương tự với u và bao phủ u. Đối tượng này được gọi là một *bộ sinh đại diện* nếu nó tương tự với nhiều đối tượng khác. Người ta có thể sử dụng đối tượng với độ đo tương tự để phân loại các bộ sinh đại diện. Quá trình tìm kiếm cho tập bao phủ tối ưu của một bảng cho trước được tiến hành như sau:

Bước 1: Chọn bộ sinh đại diện $u \in U$ và xây dựng mẫu tốt T_u phù hợp với u. Gọi U_1 là bảng con phù hợp với mẫu T_u

Bước 2: Loại bỏ U_1 khỏi U, lặp lại bước 1 với các đối tượng còn lại cho đến khi U là tập rỗng.

Tập các bảng con được sinh ra bởi thuật toán trên tạo thành một tập con tối thiểu bao phủ bảng dữ liệu ban đầu.

III.1.4. Mẫu và bài toán phân lớp

a) Phân lớp sử dụng cây nhị phân phân tách

Giả sử ta có cây nhị phân được tạo trong quá trình phân tích cây nhị phân-BDT (phần III.1.3). Đặt x là một đối tượng mới và $\mathcal{A}$(T) là một bảng con chứa tất cả các đối tượng phù hợp T, việc đánh giá x xuất phát từ gốc của cây như sau:

Bước 1: Nếu x phù hợp mẫu T đã tìm được trong $\mathcal{A}$ thì chuyển xuống cây con có cùng tầng với $\mathcal{A}$(T) nếu không thì đi đến cây con có cùng tầng với $\mathcal{A}$(¬T).

Bước 2: Nếu x là lá của cây thì chuyển xuống bước 3 ngược lại thì lặp lại bước 1 đến 2 thay thế tương ứng $\mathcal{A}$(T) hoặc $\mathcal{A}$(¬T) cho $\mathcal{A}$.

Bước 3: Gắn các luật quyết định đã được tính toán vào bảng con đã được gắn với lá để phân loại x.

b) Trường hợp phân lớp sử dụng tập bao phủ tối thiểu

Một cách tiếp cận khác cho việc phân lớp đối tượng mới dựa trên bảng con bao phủ miền, chúng ta biết rằng tất cả các bảng con từ một tập bao phủ đều gắn với một mẫu phù hợp với nó. Giả sử rằng {T_1, T_2, ..., T_m} là một tập các mẫu được xác định bởi tập bao phủ, thì đối tượng x có thể được phân loại theo các bước như sau:

Bước 1: Sử dụng các phương pháp tốt đã biết (phát hiện luật từ bảng phân bố tổng quát, rời rạc hoá dữ liệu) để sinh ra các luật quyết định cho bất kỳ một bảng con nào từ tập bao phủ.

Bước 2: Phân loại x thành các bảng con thích hợp phù hợp với mẫu từ {T_1, T_2, ..., T_m}.

Bước 3: Sử dụng luật quyết định của bảng con tìm được trong bước 2 để phân loại x.

III.2. THỬ NGHIỆM QUÁ TRÌNH KHÁM PHÁ LUẬT THEO TIẾP CẬN TẬP THÔ TRÊN BÀI TOÁN QUẢN LÝ THÔNG TIN KHÁCH XUẤT NHẬP CẢNH QUA CỬA KHẨU

III.2.1. Bài toán quản lý thông tin khách xuất nhập cảnh qua cửa khẩu

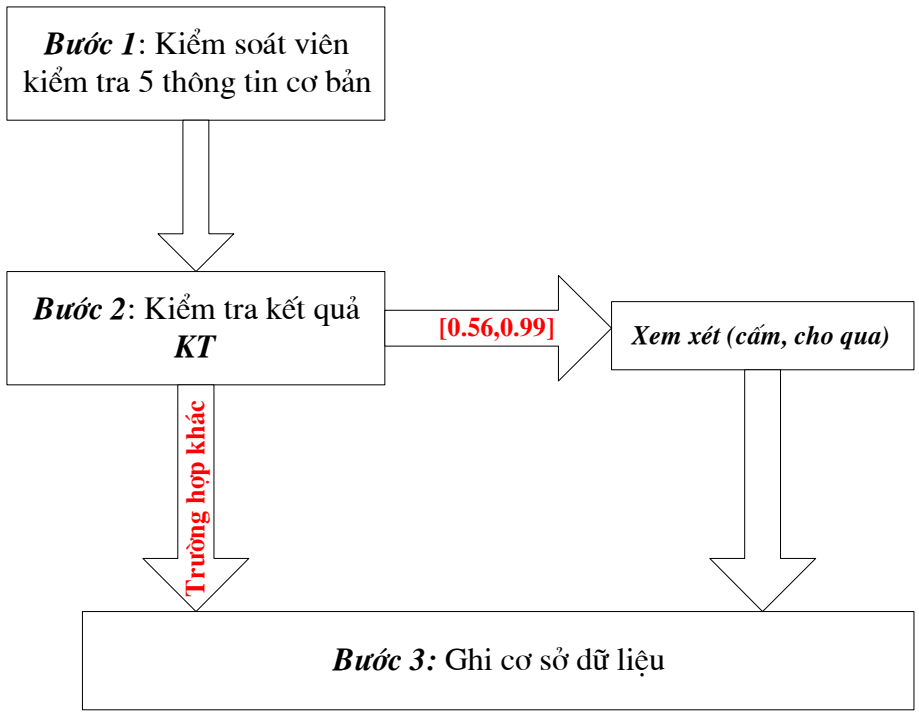
III.2.1.1. Mô tả bài toán XNC

Một số thuật ngữ sử dụng trong việc mô tả bài toán

TT	*Các thuật ngữ*	*Môtả*
1.	Khách Xuất nhập cảnh	Người Việt Nam, người nước ngoài, Việt kiều cần xuất cảnh ra nước ngoài hoặc nhập cảnh vào Việt Nam
2.	Kiểm soát viên	Chiến sĩ công an tại cửa khẩu làm nhiệm vụ kiểm soát việc xuất, nhập cảnh của khách Xuất nhập cảnh
3.	Đối tượng cấm xuất nhập cảnh	Những đối tượng đang bị nhà nước Việt Nam không cho phép nhập cảnh vào Việt Nam hoặc xuất cảnh ra nước ngoài.
4.	5 thông tin cơ bản	Họ và tên, giới tính, ngày sinh, số hộ chiếu, quốc tịch hiện nay.

Bài toán quản lý thông tin xuất nhập cảnh tại cửa khẩu quốc tế Nội Bài được đặt ra với yêu cầu cụ thể như sau: Xây dựng hệ thống quản lý thông tin về *khách xuất nhập cảnh* qua cửa khẩu quốc tế Nội Bài; Đối với mỗi khách xuất nhập cảnh khi làm thủ tục xuất, nhập cảnh qua của khẩu đều phải qua một khâu kiểm tra của *kiểm soát viên* để quyết định người đó có được phép xuất, nhập cảnh qua của khẩu Việt Nam hay không.

Việc kiểm tra đó được tiến hành theo các bước như sau:



Sơ đồ mô tả bài toán quản lý thông tin khách xuất nhập cảnh
tại cửa khẩu Nội Bài

- ***Bước 1***: Kiểm soát viên sẽ sử dụng một phần mềm máy tính để đối chiếu *5 thông tin cơ bản* trong hộ chiếu của khách xuất nhập cảnh với 5 thông tin cơ bản của các *đối tượng cấm xuất nhập* cảnh Việt Nam.
- ***Bước 2***: Kết quả của quá trình kiểm tra trả về một giá trị ***KT*** kiểu số là tỷ lệ trùng lặp 5 thông tin cơ bản của khách xuất nhập cảnh với 5 thông tin cơ bản của đối tượng cấm xuất nhập cảnh.
  + Nếu KT =1 thì khách xuất nhập cảnh đó bị cấm hoàn toàn
  + Nếu KT=[0.56,0.99] thì khách xuất nhập cảnh đó bị đưa vào diện nghi ngờ. Trong trường hợp này, kiểm soát viên cần sử dụng nghiệp vụ an ninh để quyết định đối tượng bị cấm hay cho qua.
  + Trường hợp còn lại khách được phép xuất nhập cảnh qua cửa khẩu.

- ***Bước 3***: Ghi nhận thông tin và kết quả xử lý của mỗi khách xuất nhập cảnh vào cơ sở dữ liệu.

III.2.1.2. Tập thô trong bài toán quản lý thông tin Xuất Nhập cảnh

Trong thực tế, cơ sở dữ liệu lưu trữ thông tin về khách xuất nhập cảnh được lưu trữ và mô tả dưới dạng một bảng quyết định (**bảng XNCIII.2.1.2** trong phụ lục) bao gồm nhiều thuộc tính điều kiện mô tả về khách xuất nhập cảnh (ví dụ như: Họ tên, ngày sinh, giới tính, số hộ chiếu, quốc tịch hiện nay, nơi sinh, tôn giáo, nghề nghiệp, xuất/nhập cảnh đến nước nào...) và một thuộc tính quyết định là kết quả kiểm tra đối chiếu khách xuất nhập cảnh đó được phép hay không được phép xuất/nhập cảnh qua cửa khẩu. Như vậy khi xem xét các thuộc tính mô tả về một khách xuất nhập cảnh (quốc tịch hiện nay, nơi sinh, tôn giáo, nghề nghiệp, xuất/nhập cảnh đến nước nào...) rất có thể ta sẽ thấy các thông tin này giống hệt nhau nhưng lại có kết quả kiểm tra đối chiếu khác nhau (đây là trường hợp không phân biệt được). Bài toán đặt ra là tìm ra mối quan hệ tiềm ẩn giữa các thuộc tính điều kiện và thuộc tính quyết định trong bảng quyết định này.

III.2.2. Đề xuất giải quyết tập thô trong bài toán

Trong phần này của luận văn, chúng tôi tập trung giải quyết vấn đề tập thô trong bài toán quản lý thông tin khách xuất nhập cảnh qua cửa khẩu nhằm tìm ra các luật kết hợp theo tiếp cận tập thô để biểu diễn mối quan hệ giữa các thông tin mô tả về khách xuất nhập cảnh. Ngoài ra, chúng tôi đề xuất một số phương hướng ứng dụng các kết quả tìm được trong bài toán thực tế.

III.2.2.1. Mô tả dữ liệu

a) Cấu trúc và dữ liệu mô phỏng thông tin khách xuất nhập cảnh sử dụng trong bài toán.

Cấu trúc bảng dữ liệu XNC

STT	Tên trường	Mô tả	Kiểu dữ liệu
1	HO_TEN	Họ tên khách xuất nhập cảnh	VARCHAR2(80)
2	SO_HC	Số hộ chiếu	VARCHAR(15)
3	NGAY_SINH	Ngày sinh	DATE
4	GIOI_TINH	Giới tính	VARCHAR2(5)
5	NOI_SINH	Thông tin nơi sinh của khách xuất nhập cảnh	VARCHAR2(60)
6	QT_HNAY	Quốc tịch hiện nay	NUMBER(4)
	TON_GIAO	Tôn giáo	VARCHAR2(30)
	NGHE_NGHIEP	Nghề nghiệp	VARCHAR2(40)
	DEN_TOI	Xuất nhập cảnh đến nước nào	NUMBER(4)
	XEM_XET	Xem xét xem khách có được phép xuất nhập cảnh hay không	NUMBER(1)

Trong bảng thông tin lưu trữ thông tin về khách xuất nhập cảnh. Các thông tin mô tả về một khách được lưu trữ bằng một bản ghi với nhiều thuộc tính trong bảng quyết định. Các thuộc tính trong mỗi bản ghi có đặc thù và độ quan trọng khác nhau. Chúng tôi chọn ra các thuộc tính mô tả nơi sinh, quốc tịch, tôn giáo, nghề nghiệp, xuất/nhập cảnh đến nước nào của khách xuất nhập cảnh để tìm quy luật. Vì những thuộc tính này mang thông tin đặc trưng về một con người.

Dữ liệu mô phỏng trong bảng XNC

Nơi sinh	*Quốc tịch*	*Tôn giáo*	*Nghề nghiệp*	*đến tới*	*Xem xét*
"DL"	54	"khong"	"Cong nhan"	106	0
"CHINA"	52	"khong"	"Cong nhan"	101	1
"TW"	54	"cao dai"	"Cong nhan"	101	1
"Yen Thanh, NA"	54	"khong"	"Cong nhan"	101	1
"DL"	54	"cao dai"	"Cong nhan"	105	1

"TW"	54	"cao dai"	"Cong nhan"	103	1
"CHINA"	51	"cao dai"	"Cong nhan"	103	0
"CHINA"	51	"cao dai"	"Cong nhan"	103	0
"VN"	54	"khong"	"Cong nhan"	103	1
"KR"	54	"khong"	"Cong nhan"	103	1
"HAI PHONG"	54	"cao dai"	"Cong nhan"	101	1
"SA DEC"	54	"khong"	"Cong nhan"	103	1
"HAI HUNG"	52	"khong"	"Cong nhan"	101	1
"TQ"	54	"khong"	"Cong nhan"	101	1
"DL"	54	"khong"	"Cong nhan"	101	1
“CHINA"	45	"khong"	"Cong nhan"	101	1
"DL"	224	"Dao Phat"	"Giam muc"	260	0
"NHAT"	145	"Dao Phat"	"Giam muc"	260	0
"NHAT"	145	"Dao Phat"	"Giam muc"	260	1
"TW"	224	"Dao Phat"	"Giam muc"	260	1
"DL"	224	"Dao Phat"	“Giam muc"	260	1
"Q.BINH"	48	"Dao Hoa hao"	"Cong nhan"	260	1
USA	54	"Thien chua giao"	"Kĩ sư"	260	1
CHN	79	“Phat”	"Kĩ sư"	260	0

b) Định nghĩa tập dữ liệu biểu diễn trường XEM_XET (xem khách XNC thuộc diện được phép hay không được phép xuất/nhập cảnh)

XEM_XET	*Giá trị*
1	Cấm không được phép xuất hoặc nhập cảnh qua cửa khẩu
0	Được phép xuất nhập cảnh qua cửa khẩu

c) Định nghĩa tập dữ liệu tên các quốc gia biểu diễn trường dữ liệu QT_HNAY (Quốc tịch hiện nay), DEN_TOI (nhập, xuất cảnh đến nước nào) của khách xuất nhập cảnh (**Bảng QUOCGIA** trong phụ lục).

III.2.2.2. Quá trình phát hiện luật

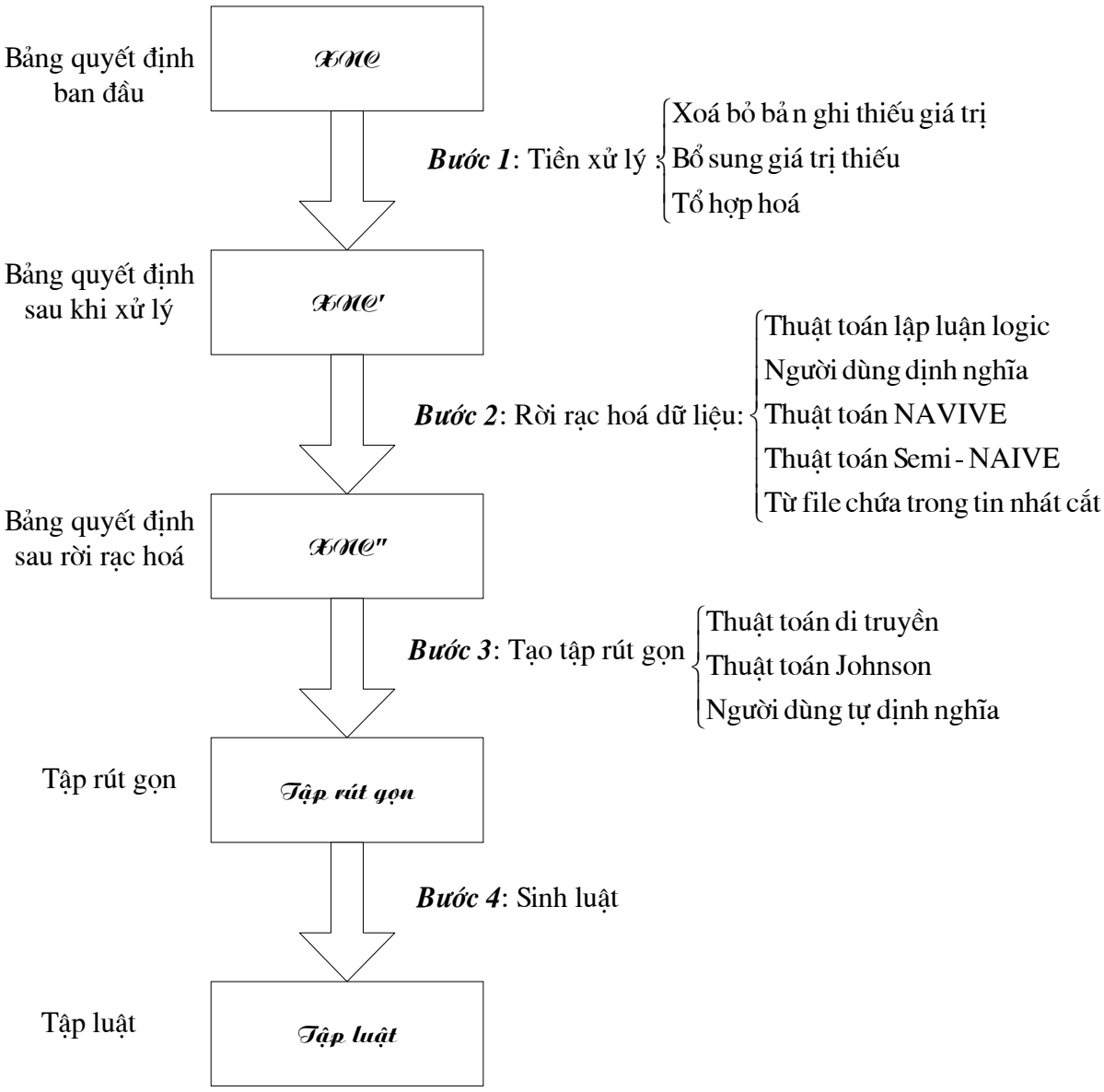
Bảng quyết định $\mathcal{DT} = (U, A \cup \{d\})$ với U là tập các khách xuất nhập cảnh, A là tập các thuộc tính điều kiện bao gồm NOI_SINH (Nơi sinh), QT_HNAY (Quốc tịch), TON_GIAO (Tôn giáo), NGHE_NGHIEP (Nghề nghiệp), DEN_TOI (Xuất/nhập cảnh đến nước nào) và thuộc tính quyết định XEM_XET (Kết quả đối chiếu khách xuất nhập cảnh được phép hay không được phép xuất/nhập cảnh). Quá trình phát hiện luật sẽ sử dụng bộ công cụ (ROSETTA - Rough sets Toolkit for Analysis of Data) [3] để thử nghiệm trên bảng quyết định với dữ liệu bao gồm 1000 bản ghi. Bộ công cụ ROSETTA do Aleksander Øhrn và cộng sự là nhóm nghiên cứu tri thức thuộc khoa Khoa học máy tính và thông tin của trường đại học Norwegian, Trondheim, Na-uy cùng nhóm Logic thuộc ĐHTH Warsaw, Ba-lan xây dựng. Đây là một bộ phần mềm gồm có các hàm và thư viện được cài đặt trên ngôn ngữ C^{++} hỗ trợ việc phân tích dữ liệu và khai phá tri thức theo tiếp cận tập thô. Các hàm và thư viện cài đặt các thuật toán sử dụng trong quá trình khám phá luật ví dụ: thuật toán lập luận logic, thuật toán NAIVE, thuật toán Semi - NAIVE (sử dụng trong việc rời rạc hoá dữ liệu); Thuật toán di truyền, thuật toán Johnson (sử dụng trong việc tìm tập rút gọn)...

Các bước thực hiện quá trình phát hiện luật kết hợp theo tiếp cận tập thô trên bảng dữ liệu xuất nhập cảnh được tiến hành như sau:

- ***Bư**ớc **1***: Tiền xử lý bảng quyết định

Thông thường từ một cơ sở dữ liệu rất có thể chứa những thông tin không hoàn chỉnh. Vì vậy cần có một bước làm sạch dữ liệu để biến bảng quyết định ban đầu thành bảng quyết định có đầy đủ giá trị của tất cả các thuộc tính. Một số phương pháp làm sạch dữ liệu có thể làm thay đổi cả tập đối tượng hay tập thuộc tính, cũng có những phương pháp bổ sung thêm giá trị cho những thuộc tính có giá trị thiếu. Có thể kể ra một số cách làm sạch dữ liệu trong bộ Toolkit như sau:

+ Xoá bỏ những bản ghi thiếu giá trị của các thuộc tính.
+ Bổ sung giá trị vào những bản ghi có thuộc tính có giá trị thiếu
+ Tổ hợp hoá dữ liệu: Mở rộng mỗi giá trị thiếu cho mỗi bản ghi (đối tượng) thành tập các giá trị có thể. Một đối tượng được mở rộng thành vài đối tượng bao phủ tất cả các trường hợp có thể xảy ra (tổ hợp giá của các giá trị thiếu của đối tượng)

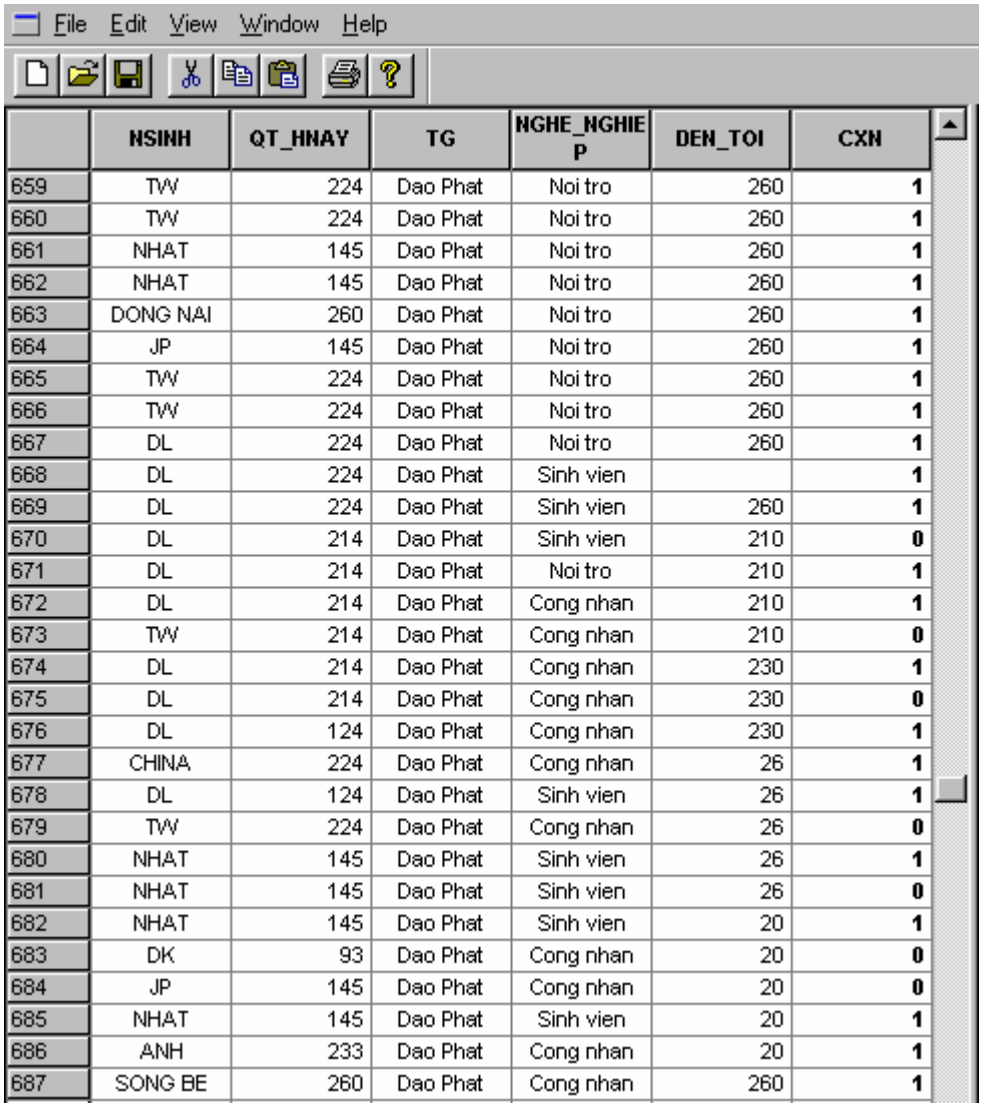


Sơ đồ mô tả quá trình sinh luật từ bảng quyết định XNC

Trong bài toán kiểm soát thông tin xuất nhập cảnh chúng tôi chọn phương pháp bổ sung giá trị vào những bản ghi có thuộc tính có giá trị thiếu. Với thuộc tính có giá trị kiểu xâu thì giá trị thiếu sẽ được thay thế bằng giá trị xuất hiện nhiều nhất trong tập giá trị của thuộc tính đó, với thuộc tính giá trị kiểu số thì

thuộc tính không hoàn hảo sẽ được thay thế bằng giá trị trung bình của tất cả tập giá trị của thuộc tính đó.

Bảng quyết định ban đầu giá trị ở thuộc tính DEN_TOI trên bản ghi số 668 bị thiếu giá trị.



	NSINH	QT_HNAY	TG	NGHE_NGHIEP	DEN_TOI	CXN
659	TW	224	Dao Phat	Noi tro	260	1
660	TW	224	Dao Phat	Noi tro	260	1
661	NHAT	145	Dao Phat	Noi tro	260	1
662	NHAT	145	Dao Phat	Noi tro	260	1
663	DONG NAI	260	Dao Phat	Noi tro	260	1
664	JP	145	Dao Phat	Noi tro	260	1
665	TW	224	Dao Phat	Noi tro	260	1
666	TW	224	Dao Phat	Noi tro	260	1
667	DL	224	Dao Phat	Noi tro	260	1
668	DL	224	Dao Phat	Sinh vien		1
669	DL	224	Dao Phat	Sinh vien	260	1
670	DL	214	Dao Phat	Sinh vien	210	0
671	DL	214	Dao Phat	Noi tro	210	1
672	DL	214	Dao Phat	Cong nhan	210	1
673	TW	214	Dao Phat	Cong nhan	210	0
674	DL	214	Dao Phat	Cong nhan	230	1
675	DL	214	Dao Phat	Cong nhan	230	0
676	DL	124	Dao Phat	Cong nhan	230	1
677	CHINA	224	Dao Phat	Cong nhan	26	1
678	DL	124	Dao Phat	Sinh vien	26	1
679	TW	224	Dao Phat	Cong nhan	26	0
680	NHAT	145	Dao Phat	Sinh vien	26	1
681	NHAT	145	Dao Phat	Sinh vien	26	0
682	NHAT	145	Dao Phat	Sinh vien	20	1
683	DK	93	Dao Phat	Cong nhan	20	0
684	JP	145	Dao Phat	Cong nhan	20	0
685	NHAT	145	Dao Phat	Sinh vien	20	1
686	ANH	233	Dao Phat	Cong nhan	20	1
687	SONG BE	260	Dao Phat	Cong nhan	260	1

Bảng quyết định đầy đủ sau khi bổ sung dữ liệu

	NSINH	QT_HNAY	TG	NGHE_NGHIEP	DEN_TOI	CXN
659	TW	224	Dao Phat	Noi tro	260	1
660	TW	224	Dao Phat	Noi tro	260	1
661	NHAT	145	Dao Phat	Noi tro	260	1
662	NHAT	145	Dao Phat	Noi tro	260	1
663	DONG NAI	260	Dao Phat	Noi tro	260	1
664	JP	145	Dao Phat	Noi tro	260	1
665	TW	224	Dao Phat	Noi tro	260	1
666	TW	224	Dao Phat	Noi tro	260	1
667	DL	224	Dao Phat	Noi tro	260	1
668	DL	224	Dao Phat	Sinh vien	260	1
669	DL	224	Dao Phat	Sinh vien	260	1
670	DL	214	Dao Phat	Sinh vien	210	0
671	DL	214	Dao Phat	Noi tro	210	1
672	DL	214	Dao Phat	Cong nhan	210	1
673	TW	214	Dao Phat	Cong nhan	210	0
674	DL	214	Dao Phat	Cong nhan	230	1
675	DL	214	Dao Phat	Cong nhan	230	0
676	DL	124	Dao Phat	Cong nhan	230	1
677	CHINA	224	Dao Phat	Cong nhan	26	1
678	DL	124	Dao Phat	Sinh vien	26	1
679	TW	224	Dao Phat	Cong nhan	26	0
680	NHAT	145	Dao Phat	Sinh vien	26	1
681	NHAT	145	Dao Phat	Sinh vien	26	0
682	NHAT	145	Dao Phat	Sinh vien	20	1
683	DK	93	Dao Phat	Cong nhan	20	0
684	JP	145	Dao Phat	Cong nhan	20	0
685	NHAT	145	Dao Phat	Sinh vien	20	1
686	ANH	233	Dao Phat	Cong nhan	20	1
687	SONG BE	260	Dao Phat	Cong nhan	260	1

- ***Bước 2*:** Rời rạc hoá dữ liệu

  Mỗi phương pháp xử lý khác nhau có thể cho ra kết quả khác nhau, có thể kể ra một số phương pháp rời rạc hoá trong bộ Toolkit như sau:

  + Sử dụng thuật toán lập luận logic
  + Rời rạc hoá theo cách người sử dụng tự định nghĩa
  + Sử dụng thuật toán Naive
  + Sử dụng thuật toán Semi-naive
  + Từ file chứa thông tin về các nhát cắt

Trong bước này chúng tôi chọn phương pháp sử dụng thuật toán lập luận logic theo tiếp cận tập thô để rời rạc hoá dữ liệu. Quá trình rời rạc hoá sẽ phân chia tập giá trị của các thuộc tính điều kiện thành các khoảng.

Bảng quyết định sau khi được rời rạc hoá như sau:

File Edit View Window Help

	NSINH	QT_HNAY	TG	NGHE_NGHIEP	DEN_TOI	CXN
536	MI THO	[36, 65)	Dao Hoa hao	Giao vien	[257, 261)	1
537	Thanh Hoa	[239, 262)	Dao Thien Ch	Tho han	[228, 257)	1
538	Ha noi	[239, 262)	Dao Thien Ch	Tho han	[23, 56)	1
539	M/HAI	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
540	HUE	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
541	LONG AN	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
542	THUA THIEN	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
543	CH	[74, 92)	Dao Thien Ch	Tho han	[257, 261)	1
544	C THO	[36, 65)	Dao Thien Ch	Tho han	[261, 262)	0
545	VN	[92, 96)	Dao Thien Ch	Tho han	[257, 261)	1
546	SAI GON	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
547	Ha Noi	[36, 65)	Dao Thien Ch	Tho han	[23, 56)	1
548	HCM	[*, 21)	Dao Thien Ch	Tho han	[257, 261)	1
549	GIA DINH	[36, 65)	Dao Thien Ch	Tho han	[23, 56)	1
550	HUE	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
551	TW	[223, 225)	Dao Thien Ch	Tho han	[257, 261)	1
552	hn	[239, 262)	Dao Thien Ch	Tho han	[23, 56)	1
553	TIEN GIANG	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
554	DL	[223, 225)	Dao Thien Ch	Tho han	[257, 261)	1
555	DL	[223, 225)	Dao Thien Ch	Tho han	[257, 261)	1
556	DL	[223, 225)	Dao Thien Ch	Tho han	[257, 261)	1
557	Hue	[239, 262)	Dao Thien Ch	Tho han	[228, 257)	1
558	Ha Noi	[36, 65)	Dao Thien Ch	Tho han	[23, 56)	1
559	TP HCM	[74, 92)	Dao Thien Ch	Tho han	[23, 56)	1
560	SAI GON	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1
561	JP	[144, 155)	Dao Thien Ch	Tho han	[257, 261)	1
562	HKG	[228, 234)	Dao Thien Ch	Tho han	[257, 261)	1
563	SAIGON	[239, 262)	Dao Thien Ch	Tho han	[257, 261)	1

- ***Bước 3:*** Tạo tập rút gọn

  Các phương pháp tính toán tập rút gọn hay tập xấp xỉ từ bảng quyết định trong bộ Toolkit là:

  + Sử dụng thuật toán di truyền

+ Sử dụng thuật toán Johnson
+ Do người sử dụng tự định nghĩa

Trong bước này chúng tôi sử dụng thuật toán di truyền để tạo tập rút gọn. Kết quả tập rút được thể hiện như sau:

	Reduct
1	{NSINH, QT_HNAY, DEN_TOI}
2	{NSINH, DEN_TOI}
3	{NSINH, TG, DEN_TOI}
4	{TG, DEN_TOI}
5	{QT_HNAY, DEN_TOI}
6	{NSINH, TG, NGHE_NGHIEP}
7	{NSINH, QT_HNAY, TG}
8	{NSINH, TG}
9	{NSINH, QT_HNAY}
10	{NSINH, QT_HNAY, NGHE_NGHIEP}
11	{NSINH}
12	{NSINH, NGHE_NGHIEP, DEN_TOI}
13	{NSINH, DEN_TOI, CXN}
14	{NSINH, NGHE_NGHIEP}
15	{NGHE_NGHIEP, DEN_TOI}
16	{NSINH, QT_HNAY, CXN}
17	{NSINH, TG, NGHE_NGHIEP, CXN}
18	{TG, CXN}
19	{QT_HNAY, TG}
20	{TG, NGHE_NGHIEP}
21	{CXN}
22	{DEN_TOI, CXN}
23	{NSINH, QT_HNAY, NGHE_NGHIEP, CXN}
24	{NGHE_NGHIEP}
25	{TG, DEN_TOI, CXN}
26	{QT_HNAY, TG, DEN_TOI}
27	{NGHE_NGHIEP, DEN_TOI, CXN}
28	{QT_HNAY, NGHE_NGHIEP, DEN_TOI}
29	{QT_HNAY, TG, NGHE_NGHIEP}

- ***Bước 4:*** Sinh luật

Sinh ra các luật kết hợp từ tập rút gọn. Kết quả tập luật sinh ra thể hiện như sau:

	Rule
111	NSINH(DL) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
112	NSINH(SA DEC) AND QT_HNAY([36, 65)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
113	NSINH(THAI) AND QT_HNAY([226, 228)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
114	NSINH(CHINA) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
115	NSINH(TH) AND QT_HNAY([226, 228)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
116	NSINH(CUU LONG) AND QT_HNAY([36, 65)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1
117	NSINH(DONG NAI) AND QT_HNAY([239, 262)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN
118	NSINH(TW) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
119	NSINH(TW) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([131, 142)) AND CXN(1) => CXN(1)
120	NSINH(THAI) AND QT_HNAY([226, 228)) AND TG(cao dai) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([131, 142)) AND CXN(1) => CXN(1)
121	NSINH(THAI) AND QT_HNAY([226, 228)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([131, 142)) AND CXN(1) => CXN(1)
122	NSINH(DL) AND QT_HNAY([223, 225)) AND TG(cao dai) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
123	NSINH(DL) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(0) => CXN(0)
124	NSINH(TW) AND QT_HNAY([223, 225)) AND TG(cao dai) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
125	NSINH(JP) AND QT_HNAY([144, 155)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
126	NSINH(HUE) AND QT_HNAY([36, 65)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(0) => CXN(0)
127	NSINH(TIEN GIANG) AND QT_HNAY([36, 65)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(
128	NSINH(NINH THUAN) AND QT_HNAY([36, 65)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(
129	NSINH(MALAY) AND QT_HNAY([155, 211)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
130	NSINH(CANADA) AND QT_HNAY([73, 74)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
131	NSINH(SAI GON) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1
132	NSINH(TQ) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
133	NSINH(DL) AND QT_HNAY([223, 225)) AND TG(khong) AND NGHE_NGHIEP(Linh muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXN(1)
134	NSINH(NHAT) AND QT_HNAY([144, 155)) AND TG(khong) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(0) => CXN(0)
135	NSINH(SAI GON) AND QT_HNAY([239, 262)) AND TG(Dao Phat) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([128, 131)) AND CXN(1) => CXI
136	NSINH(SAI GON) AND QT_HNAY([239, 262)) AND TG(Dao Phat) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([154, 157)) AND CXN(1) => CXI
137	NSINH(SAI GON) AND QT_HNAY([36, 65)) AND TG(Dao Phat) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([154, 157)) AND CXN(1) => CXN(
138	NSINH(NHAT) AND QT_HNAY([144, 155)) AND TG(Dao Phat) AND NGHE_NGHIEP(Su) AND DEN_TOI([154, 157)) AND CXN(1) => CXN(1)
139	NSINH(JP) AND QT_HNAY([144, 155)) AND TG(Dao Phat) AND NGHE_NGHIEP(Su) AND DEN_TOI([154, 157)) AND CXN(1) => CXN(1)
140	NSINH(TW) AND QT_HNAY([223, 225)) AND TG(Dao Phat) AND NGHE_NGHIEP(Giam muc) AND DEN_TOI([13, 17)) AND CXN(0) => CXN(0)

III.2.2.3. Đề xuất ứng dụng luật kết hợp tìm được trong bài toán thực tế

Dựa trên kết quả là tập luật kết hợp tìm được từ cơ sở dữ liệu khách xuất nhập cảnh chúng ta có thể xây dựng một công cụ hỗ trợ giúp kiểm soát viên đưa ra những quyết định về việc cho phép khách xuất/nhập cảnh qua cửa khẩu trong công tác hàng ngày (gọi là hệ hỗ trợ quyết định xuất nhập cảnh).

Trong thực tế khi kiểm soát viên gặp phải những trường hợp kết quả kiểm tra đối chiếu của khách xuất nhập cảnh KT=[0.56,0.99] (bước 2 mục III.2.1.1) khi đó kiểm soát viên sẽ phải sử dụng nghiệp vụ an ninh để giải quyết. Qua các lần khảo sát và làm việc thực tế tại trạm công an cửa khẩu Nội Bài, chúng tôi thấy đây là trường hợp kiểm soát viên rất hay gặp (20% trên tổng số khách xuất nhập cảnh khi làm thủ tục bị rơi vào trường hợp cần xem xét). Khi gặp phải những trường

hợp như vậy thường là rất mất thời gian để đưa ra quyết định (5->7 phút), thời gian để giải quyết một khách xuất nhập cảnh như vậy là quá lâu dẫn đến hiện tượng ùn tắc khách tại bục kiểm soát. Chúng tôi đề xuất sử dụng công cụ "**Hỗ trợ quyết định xuất nhập cảnh**" tại mỗi bục kiểm soát để kiểm soát viên sử dụng kèm với chương trình "**Quản lý thông tin khách xuất nhập cảnh**" nêu trên (hai hệ thống này có khả năng trao đổi dữ liệu với nhau). Ví dụ kiểm soát viên có thể sử dụng "Hệ hỗ trợ quyết định xuất nhập cảnh" và đặt ra câu hỏi dạng "Khách có nơi sinh là Sài gòn, quốc tịch hiện nay là Việt Nam, tôn giáo là Đạo thiên chúa, và xuất cảnh đến Mỹ" và kết quả nhận được có thể là khách xuất nhập cảnh với thông tin như vậy sẽ bị cấm không được phép xuất/nhập cảnh hoặc được phép xuất/nhập cảnh. Khi đó dựa vào kết quả trả lời từ công cụ "Hỗ trợ quyết định xuất nhập cảnh" và kinh nghiệm nghiệp vụ của mình, kiểm soát viên hoàn toàn có thể đưa ra quyết định nhanh chóng và như vậy sẽ làm giảm được thời gian xử lý một khách xuất nhập cảnh, lượng khách được giải toả nhanh. Bài toán quản lý thông tin xuất nhập cảnh (công tác thực tế của nghành công an cửa khẩu) được cải tiến rõ rệt.

III.3. KẾT LUẬN CHƯƠNG III

Dựa trên lý thuyết tập thô người ta đã xây dựng những công cụ toán học để phát hiện những mẫu, luật tiềm ẩn trong dữ liệu. Có nhiều ứng dụng được xây dựng từ những mẫu tìm được. Các mẫu tìm được có thể sử dụng để phân lớp, phân cụm, phân tách bảng dữ liệu lớn, mô tả các lớp quyết định (mục III.1).

Có nhiều ứng dụng đã được phát triển dựa trên lý thuyết tập thô trong nhiều lĩnh vực như [6]: Y tế (Hỗ trợ quyết định chữa bệnh, Chuẩn đoán bệnh viêm phổi ...); tài chính (Phân tích thói quen mua bán của khách hàng tại siêu thị, phân tích rủi ro trong kinh doanh ngân hàng ...); môi trường (Lập trình hệ thống cung cấp

nước sạch, Phân tích tính ổn định nhiệt độ ...); kỹ nghệ (Nhận dạng âm nhạc, tiếng nói, phân tích chữ viết ...); thông tin khoa học; phân tích quyết định; khoa học xã hội; sinh học; hoá học. Bộ công cụ ROSETTA [3] là một ví dụ về hệ phần mềm hỗ trợ giải quyết các bài toán trên. Bài toán quản lý thông tin khách xuất nhập cảnh được đưa vào thử nghiệm trên bộ công cụ này nhằm tìm ra một phương pháp giải quyết tính thô của bài toán. Nó tỏ ra khá hữu ích trong việc giải quyết những trường hợp không phân biệt được trong cơ sở dữ liệu.

KẾT LUẬN

Thông qua việc tìm hiểu nghiên cứu một số tài liệu khoa học về phát hiện tri thức, luận văn với đề tài "Khai phá luật theo tiếp cận tập thô" tập trung nghiên cứu về lý thuyết tập thô và ứng dụng từ đó đưa ra so sánh hình thức giữa hai cách tiếp cận (khai phá luật kết hợp theo cách tiếp cận truyền thống và khai phá luật theo tiếp cận tập thô). Trong luận văn chúng tôi cũng đề xuất một số ứng dụng của việc khai phá luật theo tiếp cận tập thô trong một bài toán cụ thể (bài toán Quản lý thông tin khách xuất nhập cảnh tại cửa khẩu Nội Bài) thông qua việc khảo sát và khai thác bộ công cụ ROSETTA do Aleksander Øhrn và cộng sự là nhóm nghiên cứu tri thức thuộc khoa Khoa học máy tính và thông tin của trường đại học Norwegian, Trondheim, Na-uy cùng nhóm Logic thuộc ĐHTH Warsaw, Ba-lan xây dựng. Luận văn đã thực hiện được những kết quả sau đây:

- Trình bày một cách tổng quan lý thuyết cơ bản về tập thô và các bước cơ bản quá trình khám phá luật theo cách tiếp cận tập thô, những ứng dụng từ mẫu và luật phát hiện được theo tiếp cận tập thô,
- Từ một số cơ sở lý thuyết: khái niệm về mẫu và luật, quá trình phát hiện mẫu và luật theo tiếp cận tập thô luận văn đã đưa ra được mối liên hệ giữa mẫu và luật để từ đó thấy được luật trong bảng quyết định là một trường hợp đặc biệt của mẫu (mục II.2.3).
- Khảo sát bài toán khám phá luật theo tập thô dựa trên một số bài toán mẫu trong bảng quyết định. Luận văn đưa ra một số nhận xét bước đầu đối sánh hình thức một số nội dung khám phá luật theo tiếp cận tập thô với khám phá luật kết hợp do Rakesh Agrawal, Tomasz Imielinski, Arun Swami đề xuất. Từ đấy, luận văn cho rằng thông qua các cách tiếp cận khác nhau song một số khái niệm cơ bản trong chúng có ý nghĩa tương đồng (mục II.3),

- Luận văn trình bày sơ bộ về bài toán quản lý thông tin khách xuất nhập cảnh tại cửa khẩu Nội Bài. Phân tích và chỉ ra tính chất thô của bài toán trong quá trình xử lý thông tin (mục III.2.1) để từ đó đưa ra mô hình thử nghiệm quá trình phát hiện luật dựa trên bộ công cụ ROSETTA.
- Luận văn đã đề xuất xây dựng bộ công cụ "**Hỗ trợ quyết định xuất nhập cảnh**" từ bộ luật tìm được theo tiếp cận tập thô của bài toán để giải quyết tính thô trong bài toán quản lý thông tin khách xuất nhập cảnh (mục III.2.2). Từ đó đề xuất việc kết hợp bài toán Quản lý thông tin khách xuất nhập cảnh với hệ công cụ Hỗ trợ quyết định xuất nhập cảnh nhằm cải thiện thời gian làm thủ tục cho khách xuất nhập cảnh của cán bộ công an cửa khẩu.

Lĩnh vực khám phá tri thức trong các cơ sở dữ liệu hiện đang được ứng dụng rộng rãi tại nhiều nước công nghiệp tiên tiến và là một trong những nội dung trọng tâm của công nghệ tri thức. Tiếp cận tập thô trong lĩnh vực này tỏ ra là một công cụ hữu hiệu.

Việc khai thác các công cụ (chẳng hạn, ROSETTA) đối với các bài toán thực tế cho thấy khả năng ứng dụng rộng rãi của nó trong nhiều lĩnh vực. Đây là một trong những hướng mà tác giả luận văn sẽ định hướng nghiên cứu và triển khai trong thời gian tới.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

[1] Hà Quang Thuỵ (1996). *Một số vấn đề về không gian xấp xỉ, tập thô đối với hệ thông tin*. Luận án Phó Tiến sĩ Khoa học Toán Lý. ĐHKHTN, 1996

Tài liệu tiếng Anh

[2]. R.Agrawal and R. Srikant (1993). *Fast algorithms for association rules in large databases*. In Proceedings of the 20th International Conference on Very Large Data Basese, pages 478-499.

[3]. Aleksander. *Discernibility and Rough Sets in Medicine: Tools and Applications* Knowledge Systems Group, Dept. of Computer and Information Science, Norwegian University of Science and Technology, Trondheim, Norway.

[4]. Ho Tu Bao (1996). *Introduction to Knowledge Discovery and Data mining*. Institute of Information Technology National Center for Natural Science and Technology.

[5]. Sinh Nguyen Hoa, Andrzej Skowron, Piotr Synak (1998). *Discovery of Data Patterns with Application to Decomposition and Classification Problems*.

[6]. Jan Komorowski, Zdzislaw Pawlak, Lech Polkowski, Andrzej Skowron (2000). *Rough sets: A tutorial*

[7]. Elena Marchiori. *Data Minning*. Free University Amsterdam Faculty of Sciences, Departement of Mathematics and Computer Science, Amsterdam, The Netherlands.

[8]. Quinlan, J.R. (1993) *C4.5: Programs for machine learning*. Morgan Kaufmann, San Mateo, CA

[9]. Andrzej Skowron, Ning Zong (2000). *Rough Sets in KDD.* Tutorial Notes.

[10]. Wojciech P. Ziarko (Ed., 1994). *Rough Sets, Fuzzy Sets and Knowledge Discovery.* Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93), Banff, Alberta, Canada, 12-15 October 1993. Springer-Verlag.