**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-------------------------------------

****

**LÊ THỊ HẰNG**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

**HÀ NỘI – 2017**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

-------------------------------------

****

**LÊ THỊ HẰNG**

**NGHIÊN CỨU ỨNG DỤNG KỸ THUẬT TẬP THÔ**

**TRONG PHÂN LỚP DỮ LIỆU**

**CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH**

**MÃ SỐ: 60.48.01.01**

**LUẬN VĂN THẠC SĨ KỸ THUẬT**

**NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. VŨ VĂN THỎA**

**HÀ NỘI – 2017**

LỜI CAM ĐOAN

# Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

# Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất cứ công trình nào.

|  |  |
| --- | --- |
|  | TÁC GIẢLê Thị Hằng |

# LỜI CẢM ƠN

Em xin được bày tỏ lòng kính trọng và biết ơn sâu sắc đến TS. Vũ Văn Thỏa, Thầy đã trực tiếp hướng dẫn và tận tình chỉ bảo em trong suốt thời gian làm luận văn tốt nghiệp. Ngoài những kiến thức Thầy truyền đạt, em còn học được ở Thầy một phong cách làm việc rất khoa học, nghiêm túc và đầy trách nhiệm.

Em xin chân thành cảm ơn toàn thể các thầy giáo, cô giáo Học viện Công nghệ Bưu chính Viễn thông đã dìu dắt, chỉ bảo tận tình cho em trong suốt thời gian học tập tại trường.

Cuối cùng, em xin gửi lời cảm ơn tới gia đình, bạn bè, đồng nghiệp đã luôn giúp đỡ và tạo điều kiện để em có thể hoàn thành tốt luận văn này.

Mặc dù em đã cố gắng hoàn thành luận văn trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót, kính mong nhận được góp ý của quý thầy cô và các bạn.

|  |  |
| --- | --- |
|  | *Hà Nội, tháng 06 năm 2017*  Học viên  Lê Thị Hằng |

# DANH MỤC THUẬT NGỮ

|  |  |
| --- | --- |
| **Thuật ngữ tiếng Anh** | **Thuật ngữ tiếng Việt** |
| Attribute Reduction | Rút gọn thuộc tính |
| Complete Decision Table | Bảng quyết định đầy đủ |
| Consistent Decision Table | Bảng quyết định nhất quán |
| Core | Tập lõi |
| Decision Rule | Luật quyết định |
| Decision Table | Bảng quyết định |
| Inconsistent Decision Table | Bảng quyết định không nhất quán |
| Indiscernibility Relation | Quan hệ không phân biệt được |
| Information System | Hệ thông tin |
| Lower Approximation | Xấp xỉ dưới |
| Rough Set | Tập thô |
| Upper Approximation | Xấp xỉ trên |

# 

# BẢNG CÁC KÝ HIỆU TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Ký hiệu, từ viết tắt** | **Diễn giải** |
|  | Hệ thông tin |
|  | Hệ quyết định |
|  | Số đối tượng |
|  | Số thuộc tính điều kiện trong hệ quyết định |
|  | Số thuộc tính trong hệ thông tin |
|  | Giá trị của đối tượng *u* tại thuộc tính *a* |
|  | Quan hệ không phân biệt |
|  | Lớp tương đương chứa *u* của quan hệ |
|  | Phân hoạch của  sinh bởi tập thuộc tính |
|  | xấp xỉ dưới của |
|  | xấp xỉ trên của |
|  | *B -* miền biên của *X* |
|  | miền dương của |
|  | Họ tất cả các tập rút gọn của hệ quyết định |
|  | Tập lõi của hệ quyết định |
|  | Tri thức sinh bởi tập thuộc tính *P* trong hệ thông tin |

# DANH MỤC BẢNG

# DANH MỤC HÌNH

MỞ ĐẦU

Những năm gần đây, cùng với sự phát triển của Công nghệ thông tin, con người tạo ra nhiều dữ liệu nghiệp vụ, các tập dữ liệu được tích lũy có kích thước ngày càng lớn, và có thể chứa nhiều thông tin ẩn dạng những quy luật chưa được khám phá. Chính vì vậy, một nhu cầu đặt ra là cần tìm cách trích rút từ tập dữ liệu đó các luật để phân lớp dữ liệu hay dự đoán những xu hướng dữ liệu tương lai. Có rất nhiều thuật toán khai phá tri thức bằng cách phân lớp và rời rạc dữ liệu như: sử dụng cây quyết định, phương pháp thống kê, các mạng nơron, thuật toán di truyền,... Gần đây, lý thuyết tập thô được nhiều nhóm nghiên cứu trong lĩnh vực khai phá tri thức và triển khai ứng dụng trong thực tế. Lý thuyết tập thô được xây dựng trên nền tảng 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, và đặc biệt hữu ích trong các bài toán phải xử lý các dữ liệu mơ hồ, không chắc chắn. Các mối quan hệ giữa các dữ liệu trong mô hình này được biểu diễn qua mối quan hệ “không phân biệt được”; các tập dữ liệu là mơ hồ, không chắc chắn được biểu diễn thông qua tập xấp xỉ trên và xấp xỉ dưới của nó. Nhờ vào những điều này mà dữ liệu có thể phân tích và xử lý bằng những công cụ toán học...Cụ thể trong lý thuyết tập thô dữ liệu được biểu diễn thông qua hệ thông tin hay bảng quyết định. Trong thực tế, với những bảng dữ liệu lớn với dữ liệu không hoàn hảo, dư thừa, liên tục hoặc biểu diễn dưới dạng các ký hiệu, lý thuyết tập thô cho phép khai phá tri thức trong những cơ sở dữ liệu như thế này nhằm phát hiện những tri thức tiềm ẩn từ những khối dữ liệu “thô” này. Tri thức tìm được được thể hiện dưới dạng các luật, các mẫu. Sau khi tìm được những quy luật chung nhất để biểu diễn dữ liệu, người ta có thể tính toán độ mạnh và độ phụ thuộc giữa các thuộc tính trong hệ thông tin.

Xuất phát từ những vấn đề trên, học viên xin lựa chọn đề tài: “Nghiên cứu ứng dụng kỹ thuật tập thô trong phân lớp dữ liệu” làm luận văn tốt nghiệp cao học.

Ý tưởng chính của lý thuyết tập thô lần đầu tiên được nhà toán học Ba Lan Z.Pawlak đề xuất. Bài báo kinh điển nổi tiếng của ông là Rough Sets, xuất bản vào năm 1982. Tiếp theo sau đó, các nhà khoa học khắp nơi trên thế giới đóng góp nhiều thành tựu quan trọng nhằm phát triển lý thuyết tập thô và ứng dụng. Một số bài toán trong lĩnh vực như ngân hàng, tài chính, y học, … đã được giải quyết thành công nhờ công cụ tập thô.

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

- 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 cơ sở dữ liệu.  
- Sử dụng được tri thức kinh nghiệm.

- Nhận ra được những 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 rút gọn.

- Cách giải thích rõ ràng và dễ hiểu.

Kiến thức cơ sở của lý thuyết tập thô cổ điển là các phép toán xấp xỉ trên, xấp xỉ dưới dựa trên phân hoạch được tạo bởi một quan hệ tương đương. Trên cơ sở đó, các kỹ thuật tập thô tập trung vào hai hướng chính:

(1) Nghiên cứu các thuật toán để rút gọn tập các thuộc tính nhằm hạn chế số chiều của không gian dữ liệu.

(2) Nghiên cứu các thuật toán sinh luật quyết định nhằm phân lớp dữ liệu để làm giảm độ phức tạp của không gian dữ liệu.

Trong luận văn này, học viên sẽ tập trung nghiên cứu hướng tiếp cận tập thô thứ 2 và ứng dụng trong phân lớp dữ liệu.

Ngoài phần mở đầu, kết luận, tài liệu tham khảo, luận văn bao gồm ba chương chính như sau:

**Chương 1: Tổng quan về tập thô và các vấn đề liên quan**

Chương này của luận văn trình bày các khái niệm cơ bản về hệ thông tin và các vấn đề liên quan, xấp xỉ trên, xấp xỉ dưới và khái niệm tập thô. Các khái niệm về ma trận phân biệt được và hàm phân biệt được, hệ quyết định và luật quyết định cũng được đề cập trong nội dung của chương.

**Chương 2: Nghiên cứu kỹ thuật sinh luật quyết định dựa trên tập thô**

Chương này của luận văn tập trung kỹ thuật tập thô thường được sử dụng trong xử lý dữ liệu là các thuật toán sinh luật quyết định dựa trên tập thô.

**Chương 3: Phân lớp dữ liệu dựa trên tập thô**

Nội dung chính của chương này là khảo sát ứng dụng tập thô trong phân lớp dữ liệu. Trên cơ sở đó, luận văn xây dựng ứng dụng tập thô hỗ trợ hệ thống phát hiện xâm nhập. Trong chương này luận văn cũng trình bày các kết quả thử nghiệm trên bộ dữ liệu cụ thể.

**Chương 1: TỔNG QUAN VỀ TẬP THÔ**

**VÀ CÁC VẤN ĐỀ LIÊN QUAN**

*Chương này của luận văn trình bày một số khái niệm cơ bản về lý thuyết tập thô và khảo sát các vấn đề liên quan. Nội dung chương này sẽ bao gồm các kiến thức nền tảng để nghiên cứu phương pháp sinh luật quyết định sẽ được nghiên cứu trong các chương tiếp theo*.

* 1. **Hệ thông tin và các vấn đề liên quan**
     1. ***Hệ thông tin***

Trong hầu hết các hệ quản trị cơ sở dữ liệu, một tập dữ liệu thường được biểu diễn dưới dạng bảng. Trong đó, mỗi hàng của bảng biểu diễn thông tin ứng với một đối tượng, còn mỗi cột biểu diễn một thuộc tính có thể đo được của một đối tượng. Bảng này được gọi là một hệ thông tin.

**Định nghĩa 1.1:** Hệ thông tin là một cặp IS = (U, A). Trong đó:

U là một tập hữu hạn khác rỗng các đối tượng (*tập vũ trụ* hay *tập phổ dụng*).

A là một tập hữu hạn khác rỗng các thuộc tính.

Với mỗi u U và a A ta ký hiệu u(a) là giá trị của đối tượng u tại thuộc tính a. Nếu gọi Va là tập tất cả các giá trị của thuộc tính a thì u(a) Va với mọi u U. Nếu B = {b1, b2, . . ,bk} A là một tập các thuộc tính, ta ký hiệu bộ các giá trị u(bi) bởi u(B). Như vậy, nếu u và v là hai đối tượng, thì ta sẽ viết u(B) = v(B) nếu u(bi) = v(bi) với mọi i = 1,2,…,k.

### Ví dụ 1: Bảng sau đây biểu diễn một hệ thông tin của 6 đối tượng với 3 thuộc tính

**Bảng 1.1: Hệ thông tin**

|  |  |  |  |
| --- | --- | --- | --- |
| U | **Đau đầu** | **Đau cơ** | **Thân nhiệt** |
| u1 | Không | Có | Cao |
| u2 | Có | Không | Cao |
| u3 | Có | Có | Rất cao |
| u4 | Không | Có | Bình thường |
| u5 | Có | Không | Cao |
| u6 | Không | Có | Rất cao |

Ta có một hệ thông tin: IS = ( U,A).

U **= {** u1, u2, u3, u4, u5, u6 };

A = { Đau đầu, Đau cơ, Thân nhiệt };

VĐau đầu = { Không, Có};

VĐau cơ = { Có, Không };

VThân nhiệt = { Bình thường, Cao, Rất cao}.

* + 1. ***Quan hệ không phân biệt được***

Một trong những đặc điểm cơ bản của lý thuyết tập thô là dùng để lưu giữ và xử lý các dữ liệu không phân biệt được. Trong một hệ thông tin theo định nghĩa trên cũng có thể có những đối tượng không phân biệt được. Trước tiên ta nhắc lại định nghĩa quan hệ tương đương như sau:

**Định nghĩa 1.2**: Một quan hệ hai ngôi (quan hệ nhị phân) R ⊆ U x U trên U là một quan hệ tương đương khi nó có cả 3 tính chất:

- Phản xạ: Mọi đối tượng đều quan hệ với chính nó.

- Đối xứng: Nếu xRy thì yRx.

- Bắc cầu: Nếu xRy và yRz thì xRz.

Quan hệ tương đương R sẽ chia tập các đối tượng U thành các lớp tương đương. Lớp tương đương của phần tử x∈U, ký hiệu là [x]R, chứa tất cả các đối tượng y mà xRy.

Bây giờ bắt đầu định nghĩa một quan hệ tương đương trên hệ thông tin. Quan hệ này sau này được sử dụng để biểu diễn những thông tin không phân biệt được.

**Định nghĩa 1.3**: Cho tập con các thuộc tính B ⊂ A trong hệ thống thông tin ( U,A ). Quan hệ B-không phân biệt được (ký hiệu là INDA(B)), được định nghĩa như sau: INDA(B) = {(x,x’) ∈ U2 | ∀a∈B, a(x)=a(x’)}

Khi đó INDA(B) là một quan hệ tương đương trên U.

Lớp tương đương chứa x của quan hệ không phân biệt được trên B được ký hiệu là [x]B.

Hai đối tượng x, x’, mà (x, x’) ∈ INDA(B) được gọi là không phân biệt được bởi các thuộc tính trong B.

Khi xét trên một hệ thông tin xác định ta sẽ viết IND(B) thay cho INDA(B).

**Ví dụ 2**: Xét *B*ả*ng 1.1* để minh họa cho quan hệ không phân biệt được. Nhận  
thấy các bệnh nhân *u2*, *u3*, *u5* không phân biệt được với thuộc tính “Đ*au* đầ*u*”,  
bệnh nhân *u2*, *u5* không phân biệt được với thuộc tính “Đ*au* đầ*u*”, “Đ*au c*ơ”…  
 *IND*({Đ*au* đầ*u*}) = {{*u2, u3*, *u5*}, {*u1*, *u4*, *u6*}}

*IND*({Đ*au* đầ*u*, Đ*au c*ơ}) = {{*u1*, *u4*, *u6*}, {*u2*, *u5*}, {*u3*}}

*IND*({Đ*au* đầ*u*, Đ*au c*ơ, *Thân nhi*ệ*t*}) = {{*u1*}, {*u2*, *u5*}, {*u3*}, {*u4*}, {*u6*}}

**1.2 Khái niệm tập thô**

***1.2.1 Xấp xỉ trên và xấp xỉ dưới***

Để biểu diễn một tập hợp bằng tri thức được cho là xác định bởi một tập thuộc tính, người ta định nghĩa hai phép xấp xỉ: xấp xỉ trên và xấp xỉ dưới.

Cho một hệ thông tin IS = ( U, A), với mỗi tập con X U và B A, ký hiệu R = IND(B) ta có hai tập con sau:

(X) = {u U | [u]B X};

(X) = {u U | [u]B X }

(X), (X) lần lượt gọi là R-xấp xỉ dưới và R-xấp xỉ trên của tập X

Tập (X) bao gồm tất cả các phần tử U có khả năng được phân loại vào những phần tử thuộc X ứng với quan hệ R.

Từ hai tập xấp xỉ người ta định nghĩa các tập:

BNB(X) = (X) - (X): B- miền biên của X.

POSB(X) = (X): B- vùng dương của X.

NEGB(X) = U - (X): B- vùng âm của X.

Ký hiệu tập thương của IND(B) trên U là U/B, các xấp xỉ trên và dưới của X có thể viết lại:

(X) = {W U/B | W X }

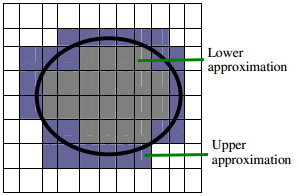
(X) = {W U/B | W X }

Trong trường hợp BNB(X) , X được gọi là *tập thô*, ngược lại X được gọi là *tập rõ.*

Đối với một hệ thông tin IS = ( U,A), B, D A, ký hiệu R = IND(B), người ta gọi B-miền khẳng định dương của D là tập được xác định như sau:

POSB(D) = (V))

Rõ ràng POSB(D) là tập tất cả các đối tượng của u sao cho với mọi v U mà u(B) = v(B) ta đều có u(D) = v(D).



Hình 1.1: Minh họa xấp xỉ dưới và xấp xỉ trên của tập thô

***1.2.2 Các tính chất của xấp xỉ***

Cho một hệ thông tin IS = (U, A), X, Y U và B A, đặt R = IND(B) khi đó:

(1L) (U) = U

(1H) (U) = U

(2L) () =

(2H) () =

(3L) (X) X

(3H) (X) X

(4L) (X Y) = (X) (Y)

(4H) (X Y) = (X) (Y)

(5L) (X) = (X)

(5H) (X) = (X)

(6L) (U – X) = U - X

(6H) (U – X) = U - X

(7L) X Y =>(X) (Y)

(7H) X Y =>(X) (Y)

(8L) (U - (X)) = U - (X)

(8H) (U - (X)) = U - (X)

(9L) K U/R, (K) = K

(9H) K U/R, (K) = K

Tính chất (3L), (4L) và (8L) là những tính chất đặc trưng cho phép xấp xỉ dưới, điều đó có nghĩa là những tính chất khác của phép xấp xỉ dưới có thể suy dẫn từ ba tính chất này. Tương tự với (3H), (4H) và (8H) là những tính chất đặc trưng của phép xấp xỉ trên.

Sự chính xác của tập xấp xỉ X đối với phân hoạch trên B là giá trị sau:

αR(X) = =

Trong đó Card(X) = |X| là lực lượng (số phần tử) của tập X. Rõ ràng 0 αR  1, nếu αR(X) = 1, ta nói X là chính xác đối với R còn αR(X) <1, X được gọi là tập thô với R.

1.2.3 Thuật toán tìm tập xấp xỉ dưới và xấp xỉ trên

a, Tìm xấp xỉ dưới

**Input:**

* Tập các đối tượng X
* Tập thuộc tính P

**Output:**

* Tập xấp xỉ dưới của X theo P

**Các biến cục bộ:**

* strRowx: Đối tượng nằm trong U
* arrXXD: tập xấp xỉ dưới của các đối tượng X theo tập thuộc tính P
* txtloptuongduong: Tập lớp tương đương

**Thuật toán:**

{

arrXXD = rỗng;

While (strRowX chạy qua tất cả các đối tượng của U)

{

If (txtloptuongduong ⊆ X)

arrXXD=arrXXD+strRowX;

}

Return arrXXD;

}

***b, Tìm tập xấp xỉ trên***

**Input:**

* Tập các đối tượng X
* Tập thuộc tính P

**Output:**

* Tập xấp xỉ trên của X theo P

**Các biến cục bộ:**

* strRowx: Đối tượng nằm trong U
* arrXXT: tập xấp xỉ trên của các đối tượng X theo tập thuộc tính P
* txtloptuongduong: Tập lớp tương đương

**Thuật toán:**

{

arrXXT = rỗng;

While (strRowX chạy qua tất cả các đối tượng của U)

{

If (txtloptuongduong ∩ X=∅)

arrXXT=arrXXT+strRowX;

}

Return arrXXT; }

***1.2.4 Khái niệm tập thô***

Khái niệm cơ bản của lý thuyết tập thô là xấp xỉ dưới và trên của một tập, sự xấp xỉ của không gian là hình thức phân loại tri thức liên quan đến miền quan tâm. Tập con được tạo ra bởi xấp xỉ dưới mô tả bởi các đối tượng là những thành phần chắc chắn của một tập, trong khi xấp xỉ trên được đặc trưng bởi các đối tượng có khả năng thuộc tập quan tâm. Mỗi tập con xác định thông qua xấp xỉ dưới và xấp xỉ trên được gọi là tập thô.

Tập thô là một bộ <(X), (X) >, trong đó (X) là xấp xỉ dưới và (X) là xấp xỉ trên. Độ chính xác thô của việc biểu diễn bởi X được cho bởi (Pawlak 1991):

0 ≤ α**B**(X) = (X)/ (X) ≤1

Nếu α**B**(X) = 1 thì X là tập cổ điển, ngược lại nếu α**B**(X) < 1 thì X là tập thô.

* 1. **Ma trận phân biệt được và hàm phân biệt được**
     1. ***Ma trận phân biệt được***

Xét hệ thông tin IS = (U, A), giả sử B A . Ta định nghĩa ma trận B là phân biệt được, kí hiệu M(B), là ma trận đối xứng cấp n×n mà mỗi phần tử của nó được xác định như sau :

cij = { a B: a(xi) ≠ a(xj)}, i,j =

Do vậy phần tử cij là tập tất cả các thuộc tính phân biệt được các đối tượng xi và xj.

* + 1. ***Hàm phân biệt được***

Tất cả các rút gọn của một hệ thông tin có thể tìm được thông qua hàm phân biệt được. Với hệ thông tin IS = (U, A), B A, có ma trận phân biệt M(B) = cij với (cij) = {a∈B: a(xi) ≠ a(xj)} và i,j = . Hàm phân biệt fs là một hàm Boolean của m biến Boolean a\*1, a\*2, …,a\*m (ứng với các thuộc tính a1, a2, …, am) được xây dựng dưới dạng chuẩn tắc tuyển như sau:

fs(a\*1, a\*2, …,a\*m) = ∧ { ∨ cij | 1 ≤ j ≤ i ≤ n, cij ≠ ∅}

Trong đó: c\*ij = {a\* | a ∈ cij}

Tập các đơn thức của fs xác định tập rút gọn của S.

**Ví dụ 3:** Xét hệ thông tin IS = (U, {a, b, c, d, e}) như bảng 1.2

**Bảng 1.2: Miêu tả hệ thông tin**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| U | a | b | c | d | e |
| x1 | 1 | 0 | 2 | 1 | 0 |
| x2 | 0 | 0 | 1 | 2 | 1 |
| x3 | 2 | 0 | 2 | 1 | 0 |
| x4 | 0 | 0 | 2 | 2 | 2 |
| x5 | 1 | 1 | 2 | 1 | 0 |

Ma trận phân biệt miêu tả trong bảng 1.3

**Bảng 1.3: Miêu tả ma trận phân biệt được**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | x1 | x2 | x3 | x4 | x5 |
| x1 |  |  |  |  |  |
| x2 | a, c, d, e |  |  |  |  |
| x3 | a | a, c, d, e |  |  |  |
| x4 | a, d, e | c, e | a, d, e |  |  |
| x5 | B | a, b, c, d, e | a, b | a, b, d, e |  |

Hàm phân biệt của hệ thông tin IS thu được từ ma trận phân biệt ở bảng 1.3 là:

fs(a,b,c,d,e)=(a∨c∨d∨e)∧a∧(a∨d∨e)∧b∧(a∨c∨d∨e)∧(c∨e)∧(a∨b∨c∨d∨e)∧ (a∨d∨e)∧(a∨b)∧(a∨b∨d∨e)

Áp dụng luật hút (luật nuốt):

x ∧ (x ∨ y) = x

x ∨ (x ∧ y) = x

Hàm phân biệt được đơn giản:

fs(a,b,c,d,e) = a∧(a∨d∨e)∧b∧(c∨e)∧ (a∨d∨e)∧(a∨b)

fs(a,b,c,d,e) = a ∧ b ∧ (c ∨ e)

Áp dụng luật phân phối ta được:

fs(a,b,c,d,e) = (a ∧ b ∧ c) ∨ (a ∧ b ∧ e)

Từ biểu thức trên, để bảo toàn khả năng phân loại của tập thuộc tính ban đầu, ta cần sử dụng tập thuộc tính {a, b, c} hoặc {a, b, e}. Đây là hai thuộc tính rút gọn hoàn toàn của hệ thông tin.

**1.4 Hệ quyết định và luật quyết định**

***1.4.1 Hệ quyết định***

Một trường hợp đặc biệt của hệ thông tin gọi là hệ quyết định nếu tập thuộc tính A được phân thành hai tập rời nhau C và D, trong đó C là tập các thuộc tính điều kiện, D là tập các thuộc tính quyết định sao cho C ∩ D = , C∪D = A. Hệ quyết định được ký hiệu là: DS = (U, C ∪ D).

**Ví dụ 4**: Hệ thông tin IS = (U, A) biểu diễn cơ sở tri thức về việc đảm bảo các giấy tờ để được lái xe tham gia giao thông được thể hiện trong bảng sau là một hệ quyết định DS = (U, C ∪ D). Trong đó:

U = {x1,x2, …, x6};

A = {Đăng kí, Bằng lái, Bảo hiểm, Lái xe};

Tập thuộc tính điều kiện C = {Đăng kí , Bằng lái, Bảo hiểm };

Tập thuộc tính quyết định D = {Lái xe};

Bảng 1.4: Hệ quyết định

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Đăng kí | Bằng lái | Bảo hiểm | Lái xe |
| x1 | Không | Có | Có | Có |
| x2 | Có | Không | Có | Có |
| x3 | Có | Có | Không | Có |
| x4 | Không | Có | Không | Không |
| x5 | Có | Không | Có | Không |
| x6 | Không | Có | Có | Có |

Cho một hệ quyết định DS = (U, C D), giả sử U/C = {X1, X2, …, Xm} và U/D = {Y1, Y2, …, Yn}. Một lớp Xi U/C được gọi là nhất quán nếu u(d) = v(d), u, v Xi, d D. Lúc này có thể viết u(D) = v(D) = Xi(D). Một lớp Yj U/D được gọi là nhất quán nếu u(a) = v(a), u, v Yj, a C.

Một hệ quyết định DS = (U, C D) là nhất quán nếu mọi lớp Xi U/C là nhất quán, ngược lại DS được gọi là không nhất quán.

***1.4.2 Luật quyết định***

Cho hệ quyết định DS = (U, C D), giả sử U/C = {X1, X2, …, Xm} và U/D = {Y1, Y2, …, Yn} là các phân hoạch được sinh bởi C, D. Với Xi U/C, Yj U/D và Xi ∩ Yj = , ký hiệu  và  lần lượt là các mô tả của các lớp tương đương  và  trong bảng quyết định DS.

Một luật quyết định đơn có dạng Zij: des(Xi) → des(Yj).

***1.4.3 Các độ đo đánh giá luật quyết định***

Một nguyên tắc quyết định là một biểu hiện dưới hình thức Φ Ψ, hay " nếu Φ thì Ψ", với Φ và Ψ là các công thức logic được gọi là điều kiện và quyết định của các quy tắc tương ứng.

Trong đó | Φ |biểu thị tập hợp của tất cả các đối tượng của U, có giá trị Φ.

Nếu Φ Ψ là một luật quyết định thì supp(Φ,Ψ) = card (|Φ ^ Ψ|) sẽ được gọi là độ hỗ trợ của các luật quyết định và

(Φ,Ψ) =

sẽ được gọi là độ mạnh của luật quyết định.

Với mỗi quy tắc tạo ra luật quyết định Φ Ψ chúng ta có độ chính xác cer(Φ,Ψ) = và một độ bao phủ cov(Φ,Ψ) =

* 1. **Ứng dụng của tập thô**

Lý thuyết tập thô cung cấp phương pháp có hiệu quả được áp dụng trong nhiều ngành của trí tuệ nhân tạo, một trong những ưu điểm của lý thuyết tập thô là chương trình triển khai thực hiện phương pháp này dễ dàng có thể chạy trên các máy tính song song, nhưng vẫn còn một số vấn đề cần giải quyết.

Gần đây, rất nhiều nghiên cứu đã được thực hiện trong tập thô kết hợp với các phương pháp trí tuệ nhân tạo như logic mờ, mạng nơtron, hệ chuyên gia và một số kết quả quan trọng đã được tìm thấy. Lý thuyết tập thô cho phép mô tả đặc tính của một tập các đối tượng trong nhóm các giá trị của thuộc tính; tìm ra toàn bộ hoặc một phần phụ thuộc giữa các thuộc tính; giảm thuộc tính thừa; tìm thấy các thuộc tính có ý nghĩa và sinh ra các luật quyết định.

Các ứng dụng của tập thô đã giải quyết những vấn đề phức tạp, và do đó đã tạo nên sự hấp dẫn cho các nhà nghiên cứu trong những năm gần đây, đã được áp dụng thành công trong một số lĩnh vực đầy thách thức như phương pháp tính toán mềm,…

* 1. **Kết luận chương 1**

Chương này của luận văn đã trình bày một số khái niệm cơ bản về lý thuyết tập thô, các khái niệm cơ bản về hệ thông tin và các vấn đề liên quan, các xấp xỉ trên, xấp xỉ dưới, ma trận và hàm phân biệt được, hệ quyết định và luật quyết định. Chương này sẽ bao gồm các kiến thức nền tảng để nghiên cứu các phương pháp rút gọn thuộc tính và sinh luật quyết định sẽ được nghiên cứu trong các chương tiếp theo.

**CHƯƠNG 2: NGHIÊN CỨU KỸ THUẬT SINH LUẬT QUYẾT ĐỊNH DỰA TRÊN TẬP THÔ**

*Trong chương này, luận văn sẽ khảo sát một số kỹ thuật sinh luật quyết định dựa trên lý thuyết tập thô.*

**2.1 Kỹ thuật sinh luật quyết định dựa trên tập thô**

Cho tới nay, đã có rất nhiều công trình nghiên cứu của các nhà nghiên cứu đã đề cập đến kỹ thuật sinh luật quyết định dựa trên tập thô. Tuy nhiên, trong phạm vi của luận văn này, học viên chỉ tập trung khảo sát hai thuật toán sau đây:

### (1) Thuật toán sinh luật quyết định dựa trên tập rút gọn của hệ quyết định [?].

**(2) Thuật toán sinh luật quyết định có bổ sung thêm thuộc tính theo yêu cầu của người dùng[?].**

### *2.1.1 Thuật toán sinh luật quyết định dựa trên tập rút gọn của hệ quyết định*

***2.1.1.1 Các khái niệm***

Cho hệ quyết định DS = (U, C D), giả sử U/C = {X1, X2, …, Xm} và U/D = {Y1, Y2, …, Yn}. Với Xi U/C, Yj U/D và Xi ∩ Yj = . Thuật toán RuleExtract hiển thị các luật quyết định dạng Zij: des(Xi) → des(Yj) với độ chắc chắn (Zij) = Xi Yj| / |Xi| và độ hỗ trợ s(Zij) = Xi Yj| / |U | tương ứng.

***2.1.1.2 Mô tả thuật toán***

**Thuật toán 2.1 RuleExtract**

**Input**: Hệ quyết định *DS = (U, C∪D, V, f)*.

**Output**: Hiển thị danh sách các luật với độ chắc chắn và độ hỗ trợ .

1. Tính phân hoạch *U/C*;
2. For each *Xi U/C*
3. Begin
4. Tính *Xi/D*;
5. For each *Yj Xi/D*
6. Begin
7. Sinh luật *Zij: des(Xi) → des(Yj)*
8. Tính *(Zij) = Yj| / |Xi|;*
9. Tính *s(Zij) = Yj| / |U |;*
10. Hiển thị luật Zij, độ chắc chắn (Zij), độ hỗ trợ s(Zij);
11. End;
12. End;
13. Return.

***Ví dụ 2.1:***

Cho hệ thông tin với các thuộc tính là { x,y,z,w }, thuộc tính quyết định là D

**Bảng 2.1: Hệ thông tin với 10 đối tượng và 5 thuộc tính**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **x** | **y** | **z** | **w** | **D** |
| 1 | A | P | A | 3 | 1 |
| 2 | A | P | S | 1 | 1 |
| 3 | P | P | A | 1 | 1 |
| 4 | P | R | A | 3 | 2 |
| 5 | A | R | A | 2 | 2 |
| 6 | P | R | P | 3 | 3 |
| 7 | S | R | P | 3 | 3 |
| 8 | S | N | P | 3 | 3 |
| 9 | S | N | S | 2 | 2 |
| 10 | S | N | S | 2 | 1 |

Trong bảng 2.1, các giá trị thuộc tính của 2 đối tượng 9 và 10 đều giống nhau, nhưng giá trị của thuộc tính quyết định lại khác nhau.

Ta có :

(X) = {1,2,3,4,5,6,7,8,9,10}

(X) = {1,2,3,4,5,6,7,8}

→ Miền biên của X là: (X) - (X) = {9,10}

Ta có hệ thông tin mới

**Bảng 2.2: Hệ thông tin với 8 đối tượng và 5 thuộc tính**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **x** | **y** | **z** | **w** | **D** |
| 1 | A | P | 3 | A | 1 |
| 2 | A | P | 1 | S | 1 |
| 3 | P | P | 1 | A | 1 |
| 4 | P | R | 3 | A | 2 |
| 5 | A | R | 2 | A | 2 |
| 6 | P | R | 3 | P | 3 |
| 7 | S | R | 3 | P | 3 |
| 8 | S | N | 3 | P | 3 |

Chúng ta sẽ tạo ra các quy tắc dựa trên tập rút gọn và lõi trong bảng 2.7.

Tập rút gọn của bảng 2.7 là {x, z, w}, {x, y, w}, {y, z, w} và lõi là w. Ta không thể loại bỏ được thuộc tính w vì đây là thuộc tính quan trọng. Bằng cách sử dụng độ chắc chắn(α) của thuộc tính, chúng ta sẽ tìm một thuộc tính không thể thiếu của bảng. Độ chắc chắn của thuộc tính cho luật kết hợp x→ D là tỉ lệ có chứa số x D với x.

Với bảng 2.2, ta có thể tính độ chắc chắn của thuộc tính x, y, z như sau:

. *Độ chắc chắn của quy tắc cho các thuộc tính x*:

(x = A) → (D =1) Độ chắc chắn của quy tắc này là 66%.

(x = P) → (D =1) Độ chắc chắn của quy tắc này là 33%

(x = P) → (D =2) Độ chắc chắn của quy tắc này là 33%

(x = A) → (D =2) Độ chắc chắn của quy tắc này là 33%

(x = P) → (D =3) Độ chắc chắn của quy tắc này là 33%

(x = S) → (D =3) Độ chắc chắn của quy tắc này là 100%

*. Độ chắc chắn của quy tắc cho các thuộc tính y:*

(y = P) → (D =1) Độ chắc chắn của quy tắc này là 100%

(y = R) → (D =2) Độ chắc chắn của quy tắc này là 50%

(y = R) → (D =3) Độ chắc chắn của quy tắc này là 50%

(y = N) → (D =3) Độ chắc chắn của quy tắc này là 100%

*. Độ chắc chắn của quy tắc cho các thuộc tính z:*

(z = 3) → (D =1) Độ chắc chắn của quy tắc này là 25%

(z = 1) → (D =1) Độ chắc chắn của quy tắc này là 100%

(z = 3) → (D =2) Độ chắc chắn của quy tắc này là 25%

(z = 2) → (D =2) Độ chắc chắn của quy tắc này là 50%

(z = 3) → (D =3) Độ chắc chắn của quy tắc này là 100%

Vì vậy thuộc tính y là thuộc tính không thể thiếu vì độ chắc chắn của các quy tắc cho thuộc tính *y* là tối đa. Ta có tập rút gọn thuộc tính mới là {y, z} ứng với bảng 2.3

**Bảng 2.3: Hệ thông tin với 8 đối tượng và 3 thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **y** | **w** | **D** |
| 1 | P | A | 1 |
| 2 | P | S | 1 |
| 3 | P | A | 1 |
| 4 | R | A | 2 |
| 5 | R | A | 2 |
| 6 | R | P | 3 |
| 7 | R | P | 3 |
| 8 | N | P | 3 |

Do trong bảng 2.3 các giá trị của đối tượng 1 và 3, 4 và 5, 6 và 7 giống nhau nên ta có thể rút gọn thành bảng 2.4 mới:

**Bảng 2.4: Hệ thông tin với 5 đối tượng và 3 thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **y** | **w** | **D** |
| 1 | P | A | 1 |
| 2 | P | S | 1 |
| 3 | R | A | 2 |
| 4 | R | P | 3 |
| 5 | N | P | 3 |

Trong bảng 2.4, chúng ta thấy w = A có hai giá trị quyết định là 1 và 2. Nó có nghĩa là dựa trên thuộc tính w chúng ta không thể đưa ra quyết định chính xác, do đó giá trị của y không thể được loại bỏ. Tương tự như vậy, chúng ta thấy y = R có hai giá trị quyết định là 2 và 3 nên dựa trên thuộc tính y chúng ta cũng không thể đưa ra quyết định chính xác, do đó giá trị của w có thể không được loại bỏ. Bây giờ, bảng 2.4 trở thành:

**Bảng 2.5: Hệ thông tin**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **y** | **w** | **D** |
| 1 | P | \* | 1 |
| 2 | P | \* | 1 |
| 3 | R | A | 2 |
| 4 | \* | P | 3 |
| 5 | \* | P | 3 |

Bảng 2.5 cho thấy cốt lõi của mỗi ví dụ. Ta có thể giảm hơn nữa Bảng 2.5 bằng cách sáp nhập các hàng trùng lặp. Bây giờ chúng ta một lần nữa loại bỏ các hàng giống hệt nhau.

**Bảng 2.6: Hệ thông tin với 3 đối tượng và 3 thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **y** | **w** | **D** |
| 1 | P | \* | 1 |
| 2 | R | A | 2 |
| 3 | \* | P | 3 |

Bảng 2.6 không thể rút gọn được nữa nên ta có các luật như sau:

* + 1. IF y → P THEN D → 1
    2. IF y → R AND w → A THEN D → 2
    3. IF w → P THEN D → 3

***2.1.2* Thuật toán sinh luật quyết định có bổ sung thêm thuộc tính theo yêu cầu của người dùng[?].**

***2.1.2.1 Các khái niệm liên quan***

Trong phần này, chúng ta xem xét các vấn đề về việc cập nhật gần đúng của một tập con X của U về cách thêm và loại bỏ một thuộc tính tại một thời điểm.

**Định lý 2.1**: Cho thuộc tính a A và a P. Xấp xỉ dưới của X bằng cách thêm thuộc tính a vào P có thể được cập nhật về X, pX, {a}X và {a}X như sau:

Với Y = {x in pX {a}X ׀ b

**Định lý 2.2**:

Cho thuộc tính a P, xấp xỉ dưới của X bằng cách bớt đi phần tử a của P có thể được cập nhật về X, p – {a}X như sau : X = X - p – {a}X

Với p – {a}X = { x in {b}X I b ⊄ X }

Chú ý: Thuộc tính a là không cần thiết khi p – {a}X(X) =

**Định lý 2.3**: Cho thuộc tính a A và a ∉ P. Xấp xỉ trên của X bằng cách thêm a vào P có thể được cập nhật về pX như sau: X = **x** ( pX **–** Z)

Trong đó Z là tập của các đối tượng bổ sung mà được xác định bằng cách thêm thuộc tính a vào P và nó được định nghĩa là: 



**Định lý 2.4**: Cho thuộc tính a P, Xấp xỉ trên của X bằng cách loại bỏ thuộc tính a từ P có thể được cập nhật về pX như sau: X = X pX Z’

Với Z’ = {x in bX | b  {b}X }

***2.1.2.2 Mô tả thuật toán***

***Từ các khái niệm, tính chất của mục 2.1.2.1, ta có thuật toán như sau***

**Thuật toán 2.2**

**Input:**

1. A decision table with attribute set A and a decision attribute *d in A.*

(Cho hệ thông tin A và d là thuộc tính quyết định trong A)

1. Attribute generalizations provided by users.

(Thuộc tính được thiết lập bởi người dùng)

**Output:**

Certain and possible classification rules for classes in the decision attribute.

(Xác định và quy tắc phân lớp dựa trên thuộc tính quyết định đó)

**Begin**

From the decision table,( Từ hệ quyết định,

for each class Xi in the decision attribute do (Cho mỗi lớp Xi trong hệ quyết định)

**begin**

Compute lower and upper approximations generated by single condition attributes;

(Tính xấp xỉ dưới và xấp xỉ trên tạo ra bởi các thuộc tính quyết định;)

Compute lower and upper boundary sets by single condition attributes;

(Tính toán miền biên dưới và trên bởi các thuộc tính quyết định;)

Find a cover ***R*** for the set of condition attributes;

(Tìm và cô lập thuộc tính a trong R được thiết lập bởi hệ quyết định;)

Compute lower and upper approximations generated by ***R;***

(Tính toán xấp xỉ dưới và trên của thuộc tính được tạo ra bởi R;)

Compute lower and upper boundary sets generated by ***R;***

(Tính toán miền biên dưới và trên được tạo ra bởi R;)

**end;**

**repeat**

Get a generalization gi for attribute a provided by the user;

(Nhận a là thuộc tính khái quát cho gi được cung cấp bởi người dùng;)

***If*** attribute a is in ***R*** (thuộc tính a *R)*

**then**

**begin**

Update approximations and boundary sets by *R - {a};*

(Cập nhật xấp xỉ và giới hạn được cài đặt bởi R - {a};)

Update approximations and boundary sets by *R* {a} based on gi;

(Cập nhật xấp xỉ và giới hạn được cài đặt bởi R ∪ {a} dựa trên gi; )

**end;**

**else**

Update approximations and boundary sets by *R*  {a};

(Cập nhật xấp xỉ và giới hạn được cài đặt bởi R ∪ {a};)

Find a cover for the new set of attributes;( Tìm và bao phủ thuộc tính mới);

**if**  rules desired (Luật quyết định )

**then** generate rules by LERS;( Quy tắc tạo bởi LERS);

**until** terminated by user;( Kết thúc bởi người dùng)

**end.**

**2.2 Kết luận chương 2**

Chương 2 luận văn đã tìm hiểu một số khái niệm về thuộc tính, tập thuộc tính, các loại thuộc tính. Chương này cũng đã khảo sát một số thuật toán rút gọn tập thuộc tính và sinh luật quyết định dựa trên lý thuyết tập thô. Để từ đó xây dựng chương trình rút gọn tập thuộc tính trong hệ thông tin, cũng như việc đưa ra các tập luật quyết định trong hệ thông tin.

**CHƯƠNG 3 : PHÂN LỚP DỮ LIỆU DỰA TRÊN TẬP THÔ**

*Trong chương này luận văn sẽ khảo sát ứng dụng tập thô để phân lớp dữ liệu, đưa ra luật quyết định. Từ đó đề xuất mô hình ứng dụng tập thô hỗ trợ hệ thống phát hiện xâm nhập bất thường trong mạng. Các kết quả thử nghiệm cũng được đề cập đến trong chương này của luận văn.*

**3.1 Bộ dữ liệu KDD cup 99**

KDDCUP 99 là tập dữ liệu được sử dụng trong cuộc thi Khám phá kiến thức  
và Khai thác dữ liệu quốc tế lần thứ 3 cùng với Hội nghị Khám phá kiến thức và  
Khai thác dữ liệu KDD-99 lần thứ năm. Nhiệm vụ trong cuộc thi là xây dựng một  
máy phát hiện xâm nhập mạng, một mô hình tiên đoán có khả năng phân biệt giữa  
các kết nối “xấu” hay còn gọi là xâm nhập hoặc tấn công, và các kết nối “tốt” hay  
còn gọi là bình thường. Cơ sở dữ liệu này chứa một tập hợp các dữ liệu được kiểm  
toán, bao gồm một loạt các sự xâm nhập được mô phỏng trong một môi trường  
mạng quân đội.   
Phần mềm phát hiện xâm nhập mạng bảo vệ một mạng lưới máy tính từ người  
sử dụng trái phép và có thể bao gồm cả người trong cuộc. Phát hiện xâm nhập học  
có nhiệm vụ là xây dựng một mô hình tiên đoán (tức là phân loại) có khả năng phân  
biệt giữa kết nối "xấu" được gọi là sự xâm nhập hoặc tấn công, và "tốt" kết nối bình  
thường.  
Năm 1998 Chương trình đánh giá phát hiện xâm nhập đã được DARPA chuẩn  
bị và quản lý bởi MIT Lincoln Labs. Mục tiêu là để khảo sát và đánh giá nghiên cứu  
trong việc phát hiện xâm nhập. Một tập hợp các tiêu chuẩn của dữ liệu được kiểm  
toán, bao gồm một loạt các mô phỏng của sự xâm nhập được cung cấp trong một  
môi trường mạng quân sự. Cuộc thi phát hiện xâm nhập KDD 1999 sử dụng một  
phiên bản của tập dữ liệu này. Lincoln Labs thiết lập một môi trường để có được  
chín tuần dữ liệu thô TCP dump cho một mạng cục bộ (LAN) mô phỏng một mạng  
LAN không quân Mỹ điển hình. Họ hoạt động mạng LAN như thể nó là một môi  
trường Air Force nhưng nó rải rác với nhiều cuộc tấn công.  
Dữ liệu huấn luyện thô là khoảng 4 gigabyte nén dữ liệu nhị phân TCP dump  
lấy từ bảy tuần lưu lượng mạng. Điều này đã được xử lý vào khoảng 5.000.000 bản  
ghi kết nối. Tương tự như vậy, hai tuần của dữ liệu thử nghiệm bắt được khoảng hai  
triệu bản ghi kết nối.

1. DoS - denial of service (Từ chối dịch vụ): Để ngăn cản người dùng hợp pháp truy cập và sử dụng một dịch vụ nào đó, kẻ tấn công sẽ chiếm dụng một lượng lớn tài nguyên mạng như băng thông, bộ nhớ... bằng cách liên tục gửi yêu cầu dịch vụ, do đó làm mất khả năng xử lý các yêu cầu dịch vụ từ người dùng hợp pháp. Có rất nhiều biến thể của tấn công DoS bao gồm TCP - SYN Flood, ICMP/UDP Flood, Smurf, Ping of Death, Teardrop, Mainbomb, Apache2...

2. R2L - Remote to Local: Trong loại tấn công này, tin tặc cố gắng  
đạt được quyền truy cập khu vực vào hệ thống máy tính bằng việc gửi các gói  
tin tới hệ thống thông qua mạng. Một vài cách phổ biến mà loại này thực hiện là đoán mật khẩu thông qua phương pháp từ điển brute-force, FTP Write,…

3. U2R - User to Root: Trong lớp tấn công này, tin tặc với quyền của  
một người dùng bình thường cố gắng để đạt được quyền truy nhập cao nhất  
(đặc quyền của người quản trị hệ thống) vào hệ thống một cách bất hợp  
pháp.Một cách phổ biến của lớp tấn công này là thực hiện bằng phương pháp  
gây tràn bộ đệm.

4. Probe: Trong loại tấn công này, tin tặc quét mạng hoặc  
máy tính để tìm ra điểm yếu dễ tấn công mà thông qua đó tin tặc có thể khai  
thác hệ thống. Điều này có phần giống như theo dõi, giám sát hệ thống. Một  
cách phổ biến của loại tấn công này là thực hiện thông qua việc quét các cổng  
của hệ thống máy tính. Bằng việc này, tin tặc có thể lấy được thông tin về cổng đang mở, dịch vụ đang chạy, và rất nhiều thông tin chi tiết nhạy cảm khác như địa chỉ IP, địa chỉ MAC, các luật tường lửa đang sử dụng, …

**3.2 Mô hình phân lớp dữ liệu dựa trên sinh luật quyết định**

Phân lớp (classification) dữ liệu là một tiến trình xử lý nhằm xếp các mẫu dữ liệu hay các đối tượng vào một trong các lớp đã được định nghĩa trước. Các mẫu dữ liệu hay các đối tượng được xếp về các lớp dựa vào giá trị của các thuộc tính (attributes) cho một mẫu dữ liệu hay đối tượng. Sau khi đã xếp tất cả các đối tượng đã biết trước vào các lớp tương ứng, lúc này mỗi lớp được đặc trưng bởi tập các thuộc tính của các đối tượng chứa trong lớp đó.

Khi nghiên cứu một đối tượng, hiện tượng, chúng ta chỉ có thể dựa vào một số hữu hạn các đặc trưng của chúng. Nói cách khác, ta chỉ xem xét biểu diễn của đối tượng, hiện tượng trong một không gian hữu hạn chiều, mỗi chiểu ứng với một đặc trưng được lựa chọn. Khi đó, phân lớp dữ liệu trở thành phân hoạch tập dữ liệu thành các tập con theo một tiêu chuẩn nhận dạng được. Như vậy, phân lớp là quá trình "nhóm” các đối tượng "giống” nhau vào "một lớp” dựa trên các đặc trưng dữ liệu của chúng.

Quá trình phân lớp dữ liệu dựa trên tập thô thường sử dụng các kỹ thuật sinh luật quyết định như trình bày trong hình 3.3 dưới đây.

Sinh luật quyết định dựa trên tập thô

Dữ liệu dưới dạng bảng quyết định

Lớp 1

Lớp 2

…

Lớpn



Hình 3.3: Mô hình phân lớp dữ liệu dựa trên sinh luật quyết định

**3.3 Cài đặt và thử nghiệm**

**KẾT LUẬN**

***Luận văn đã đạt được một số kết quả như sau:***

Trình bày về các khái niệm liên quan đến hệ thông tin, hệ quyết định, các khái niệm, tính chất về tập thô, các thông tin về xấp xỉ trên, xấp xỉ dưới, cách tìm tập xấp xỉ dưới, xấp xỉ trên của tập thô. Các khái niệm về luật quyết định, ma trận phân biệt được, hàm phân biệt được. Luận văn cũng khảo sát ứng dụng tập thô trong trích chọn đặc trưng và phân lớp dữ liệu.

Tìm hiểu các thuật toán trong việc tìm tập rút gọn của hệ thông tin dựa trên tập thô, các thuật toán sinh luật quyết định trong hệ thông tin dựa trên lý thuyết tập thô.

Xây dựng và cài đặt ứng dụng cho việc rút gọn tập thuộc tính, sinh luật quyết định dựa trên các thuật toán theo bộ dữ liệu.

***Hướng phát triển trong tương lai:***

Tác giả luận văn sẽ tiếp tục nghiên cứu, tìm hiểu các thuật toán, các phương pháp rút gọn tập thuộc tính, các phương pháp sinh tập luật quyết định trong hệ thông tin dựa trên lý thuyết tập thô.

Nâng cao hiệu quả để chương trình chạy nhanh hơn, trích chọn được dữ liệu kể cả dữ liệu đầu vào lớn.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

**Tiếng Anh**