

LỜI CAM ĐOAN

Tôi xin cam đoan những kiến thức, nội dung trình bày trong luận văn là kiến thức do tôi tìm hiểu, nghiên cứu, đọc, dịch tài liệu, tổng hợp và trình bày theo những kiến thức của cá nhân tôi dưới sự hướng dẫn của TS. Trần Mạnh Tuấn.

Các kết quả trong luận văn do tôi nghiên cứu và chưa công bố tại bất kì đâu. Các tài liệu có liên quan được tôi sử dụng trong quá trình làm luận văn đều được ghi rõ nguồn gốc. Tôi xin cam đoan đây luận văn không sao chép của ai mà hoàn toàn do tôi nghiên cứu và thực hiện.

Tôi xin chịu toàn bộ trách nhiệm với những cam đoan của mình.

Hà Nội, ngày 24 tháng 11 năm 2022

Học viên

Trần Ngọc Thái Sơn

LỜI CẢM ƠN

Sau thời gian học tập, nghiên cứu và thực hiện, đến nay tôi đã hoàn thành công trình nghiên cứu của mình. Trong quá trình làm luận văn này, tôi nhận được sự động viên, giúp đỡ của các thầy cô, bạn bè và người thân.

Trước tiên, tôi xin chân thành cảm ơn sâu sắc tới TS. Trần Mạnh Tuấn đã nhiệt tình hướng dẫn, cung cấp tài liệu, tạo điều kiện thuận lợi để tôi được hoàn thành tốt nhất luận văn này.

Tôi xin gửi lời cảm ơn tới các thầy giáo của Viện Công nghệ thông tin. Ban Lãnh đạo, phòng Đào tạo, các phòng chức năng của Học viện Khoa học và Công nghệ đã giảng dạy và tạo ra môi trường học tập, nghiên cứu rất tốt để tôi có thể hoàn thành đề tài của mình.

Sau cùng, tôi xin gửi lời cảm ơn đến gia đình, bạn bè, các bạn cùng lớp cao học ITT20B, là những người luôn đồng hành, giúp đỡ, chia sẻ với tôi trong quá trình thực hiện luận văn này.

Trân trọng!

Học viên

Trần Ngọc Thái Sơn

MỤC LỤC

DANH MỤC BẢNG	5
DANH MỤC HÌNH VẼ, BIỂU ĐỒ	6
MỞ ĐẦU	7
1. Lý do chọn đề tài.....	7
2. Mục đích và nhiệm vụ nghiên cứu.....	7
3. Đối tượng và phạm vi nghiên cứu.....	7
4. Phương pháp nghiên cứu và đóng góp của luận văn	8
5. Cấu trúc của luận văn.....	8
CHƯƠNG 1. CƠ SỞ LÝ THUYẾT.....	9
1.1. Lý thuyết đồ thị.....	9
1.2. Logic mờ	10
1.3. Suy diễn mờ (Fuzzy Inference).....	15
1.4. Đồ thị tri thức mờ.....	18
1.5. Ngôn ngữ MATLAB.....	20
1.6. Tổng kết chương 1	24
CHƯƠNG 2. ĐỒ THỊ TRI THỨC MỜ.....	25
2.1. Mô hình đồ thị tri thức mờ trong phân lớp dữ liệu	25
2.2. Xây dựng luật mờ.....	25
2.3. Biểu diễn đồ thị tri thức mờ từ luật mờ.....	28
2.4. Suy diễn trên đồ thị tri thức mờ	30
2.5. Các độ đo đánh giá quá trình phân lớp dữ liệu	33
2.6. Ví dụ số về việc xây dựng và suy diễn trên đồ thị tri thức mờ	33
2.7. Tổng kết chương 2	43
CHƯƠNG 3. CÀI ĐẶT VÀ XÂY DỰNG ỨNG DỤNG.....	44
3.1. Bài toán ứng dụng	44
3.2. Mô tả dữ liệu	44
3.3. Cài đặt và xây dựng ứng dụng	48
3.4. Kết quả thực nghiệm	50

KẾT LUẬN	55
PHỤ LỤC	56
Danh mục tài liệu tham khảo	68

DANH MỤC BẢNG

Bảng 2. 1: Hệ luật mờ	28
Bảng 2. 2: Tập dữ liệu đầu vào	35
Bảng 2. 3: Bộ dữ liệu đầu vào.....	36
Bảng 2. 4: Bảng ma trận xác định đồ thị từ luật.....	39
Bảng 2. 5: Bảng tính giá trị C với nhãn 1	40
Bảng 2. 6: Bảng tính giá trị C với nhãn 2	40
Bảng 3. 1: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh ung thư Breast.....	45
Bảng 3. 2: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh tiểu đường Diabetes.....	45
Bảng 3. 3: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu đo chất lượng rượu Wine.....	46
Bảng 3. 4: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh gan Liver	46
Bảng 3. 5: Dữ liệu Y học cổ truyền	48
Bảng 3. 6: Kết quả thực nghiệm Accuracy trên bộ dữ liệu UCI.....	51
Bảng 3. 7: Kết quả thực nghiệm thời gian trên bộ dữ liệu UCI.....	51
Bảng 3. 8: Kết quả thực nghiệm Accuracy trên bộ dữ liệu tiền sản giật	53
Bảng 3. 9: Kết quả thực nghiệm thời gian trên bộ dữ liệu tiền sản giật	53

DANH MỤC HÌNH VẼ, BIỂU ĐỒ

Hình 1. 1: Một số dạng hàm thuộc cơ bản	11
Hình 1. 2: Mô hình suy luận mờ với một luật-một tiền đề	16
Hình 1. 3: Mô hình suy luận mờ một luật-nhiều tiền đề.....	17
Hình 1. 4: Mô hình suy luận mờ hai luật hai tiền đề.....	18
Hình 1. 5: Minh họa một đồ thị tri thức mờ.....	19
Hình 2. 1: Mô hình đồ thị tri thức mờ trong phân lớp dữ liệu	25
Hình 2. 2: Sơ đồ tổng quan hệ suy diễn mờ	27
Hình 2. 3: Biểu diễn đồ thị tri thức mờ từ luật mờ theo thuộc tính	29
Hình 2. 4: Biểu diễn đồ thị tri thức mờ	30
Hình 2. 5: Quá trình suy diễn trên đồ thị tri thức mờ	32
Hình 2. 6: Đồ thị FKG cho 6 luật.....	39
Hình 3. 1: Mô hình cho bài toán hỗ trợ chẩn đoán bệnh trong y học cổ truyền .	49
Hình 3. 2: Đồ thị tri thức mờ với bệnh án Y học cổ truyền	52

MỞ ĐẦU

1. Lý do chọn đề tài

Với các nghiên cứu gần đây cho thấy việc sử dụng đồ thị tri thức mờ là một trong những vấn đề mới. Đồ thị tri thức (KG) là một phương pháp mạnh mẽ hỗ trợ xử lý các vấn đề trong khai phá dữ liệu. KG có thể kết hợp với các kỹ thuật khác nhau để giải quyết các bài toán trong học máy. Tuy nhiên, KG gặp khó khăn cho việc xây dựng đồ thị cũng như suy luận gần đúng trên các tập dữ liệu đầu vào có thông tin chưa đầy đủ, chưa chính xác. Khi đó, mô hình đồ thị tri thức mờ (FKG) đã được thiết kế vào năm 2020 để giải quyết các vấn đề của KG kết hợp với logic xây dựng lên đồ thị tri thức mờ. Một biểu đồ được hình thành với các nút được biểu diễn bằng các nhãn ngôn ngữ và các cạnh được xác định bởi sự kết nối giữa các nhãn ngôn ngữ và các nhãn đầu ra. Đồ thị FKG thể hiện lý do theo quy luật tự nhiên trong đó tác động của giá trị của biên ngôn ngữ để đưa ra kết quả đầu ra tương ứng.

Do vậy, việc nghiên cứu về đồ thị tri thức mờ là cần thiết, nó có thể giải quyết hiệu quả được một số bài toán phân lớp dữ liệu.

2. Mục đích và nhiệm vụ nghiên cứu

- Nghiên cứu về lý thuyết mờ, lý thuyết đồ thị, đồ thị tri thức mờ.
- Xây dựng demo và thử nghiệm mô hình với bài toán phân lớp dữ liệu.

3. Đối tượng và phạm vi nghiên cứu

- Logic mờ, đồ thị, đồ thị tri thức mờ (FKG).
- Nghiên cứu về Logic mờ, lý thuyết đồ thị, đồ thị tri thức mờ
- Cài đặt, mô hình dựa trên ngôn ngữ lập trình Matlab
- Xây dựng thực thi demo trên bộ dữ liệu thu thập từ UCI và dữ liệu thu thập thực tế.

4. Phương pháp nghiên cứu và đóng góp của luận văn

Phương pháp nghiên cứu lý luận: Học viên tập trung vào việc đọc hiểu, phân tích bài toán, thu thập dữ liệu cho bài toán thông qua các nguồn tài liệu từ sách, giáo trình, ... liên quan đến kiến thức sử dụng trong luận văn.

Phương pháp nghiên cứu thực tiễn: Tiến hành cài đặt mô hình đồ thị tri thức mờ, xây dựng demo với bộ dữ liệu thu thập. Các bước thực hiện trong quá trình xây dựng mô hình:

- Thu thập dữ liệu.
- Tiền xử lý dữ liệu
- Xây dựng mô hình phân lớp kết quả .
- Đánh giá mô hình.
- Báo cáo luận văn hoàn chỉnh về đồ thị thức mờ và ứng dụng
- Demo mô hình đồ thị tri thức mờ.

5. Cấu trúc của luận văn

Mở đầu: Trình bày tổng quan về đề tài

Chương 1: Cơ sở lý thuyết: trình bày các lý thuyết liên quan sử dụng trong đồ án

Chương 2: Đồ thị tri thức mờ: trình bày về mô hình đồ thị tri thức mờ, cách biểu diễn đồ thị tri thức mờ, suy diễn trên đồ thị tri thức mờ

Chương 3: Cài đặt và ứng dụng: trong chương này em trình bày về cài đặt mô hình tri thức mờ, thực nghiệm trên bộ dữ liệu UCI, đánh giá mô hình đồ thị tri thức mờ.

Kết luận: đánh giá những công việc đã thực hiện được và chưa thực hiện được trong quá trình làm luận văn, đề xuất hướng phát triển trong tương lai.

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT

1.1. Lý thuyết đồ thị

Đồ thị được xây dựng dựa trên các đỉnh (hay nút), các cung (cạnh) là đường nối giữa các đỉnh của đồ thị với nhau. Khi lập trình để biểu diễn trên máy tính người ta có thể sử dụng nhiều cách khác nhau.

Trong các lĩnh vực sử dụng đồ thị nhiều như: Toán học, tin học, ... Khi đó đồ thị là một phương pháp tỏ ra hiệu quả với các bài toán ứng dụng trong thực tế. Đồ thị giải quyết được các bài toán thực tế như: giao thông, du lịch, giáo dục, ... Nó là một công cụ trực quan hóa để biểu diễn, diễn tả các bài toán.

Một đồ thị được xây dựng có cấu trúc rời rạc, nó gồm 2 thành phần: tập các đỉnh và tập các cạnh.

Khi đó đồ thị: $G=(V,E)$

Trong đó đồ thị được gọi là G , tập các đỉnh được gọi là V , tập các cạnh được gọi là E . Mỗi cạnh đồ thị cặp (u,v) với đỉnh u được nối với đỉnh v , hai đỉnh u, v thuộc vào tập V .

Người ta chia đồ thị dựa trên tính chất của các cạnh trong tập E :

- Đồ thị G là đơn đồ thị nếu như giữa hai đỉnh (u,v) của V chỉ có không quá một cạnh trong E để nối từ u tới v .
- Đồ thị G là đồ thị đa cạnh nếu giữa đỉnh u và đỉnh v của V có thể có từ hai cạnh trong E nối từ đỉnh u tới đỉnh v .
- Đồ thị G được gọi là đồ thị vô hướng (undirected graph) khi các cạnh nối giữa đỉnh u và đỉnh v trong đồ thị không định hướng.
- Đồ thị có hướng G (directed graph) là đồ thị các cạnh nối giữa đỉnh v với đỉnh u có định hướng, nó xác định chiều của đường đi của các cạnh trong đồ thị.

Cạnh còn được gọi là cung trên đồ thị có hướng. Nếu cạnh nối đỉnh u với đỉnh v trong đồ thị tương ứng với 2 cung là: $(u \rightarrow v)$, $(v \rightarrow u)$. Khi đó đồ thị vô hướng cũng có thể coi là đồ thị có hướng.

1.2. Logic mờ

Logic mờ xây dựng dựa trên lý thuyết mờ được sử dụng để suy luận, lập luận dựa trên việc xấp xỉ thay vì lập luận chính xác của logic (như lập luận tiến, lập luận lùi).

Logic mờ được sử dụng như là một mặt ứng dụng của lý thuyết mờ để xử lý các giá trị mờ trong thế giới thực trong các bài toán thực tế. [1,2,3,4]

Độ chính xác thường hay nhầm lẫn với xác suất. Nhưng, hai khái niệm này là hoàn toàn khác nhau, độ chính xác trong logic mờ không phải khả năng xảy ra một biến cố hay điều kiện nào đó mà là việc biểu diễn độ liên thuộc với các tập được định nghĩa không rõ ràng.

Logic mờ được xây dựng dựa trên độ thuộc có miền giá trị trong đoạn $[0,1]$, nó biểu diễn các câu không chính xác trong thế giới thực như: “hơi nhanh”, “rất nhanh”, “hơi chậm”, “rất chậm”, “chậm một chút”.... Khi đó, các phép toán trong tập hợp có thể không xác định được đầy đủ các quan hệ giữa các câu này với nhau. Do vậy, sự quan hệ giữa lý thuyết xác suất và logic mờ có liên quan tới nhau.

Ý tưởng về Logic mờ được giáo sư Lotfi Zadeh của Đại học California-Berkeley xây dựng từ những năm 1965. Tuy được ứng dụng thành công trong nhiều lĩnh vực, nhưng nó cũng tồn tại nhiều nhược điểm khác nhau trên nền tảng toán học, dẫn đến một số nghiên cứu khoa học về logic mờ không chỉ rõ được luận cứ vững chắc mà chỉ thực hiện thông qua thực nghiệm của thế giới thực. Nó bị phủ nhận bởi một số kỹ sư ngành điều khiển tự động vì khả năng thẩm định và một số lý do khác. Trong toán học người ta luôn mong muốn xây dựng một nền tảng vững chắc trong mọi nền tảng, nhưng thực tế có một số hiện tượng, sự vật chỉ đúng và sai trong từng hoàn cảnh, thời điểm khác nhau khi đó các mô hình toán học biểu diễn không đầy đủ và toàn vẹn được. Ví dụ trong lĩnh vực thống kê, các nhà thống kê luôn khẳng định rằng chỉ có xác suất thể hiện cho các sự không chắc chắn cho sự chặt chẽ của toán học trong một không gian vô hạn, nhưng thực tế thì mọi thứ đều có giới hạn, khi đó xác

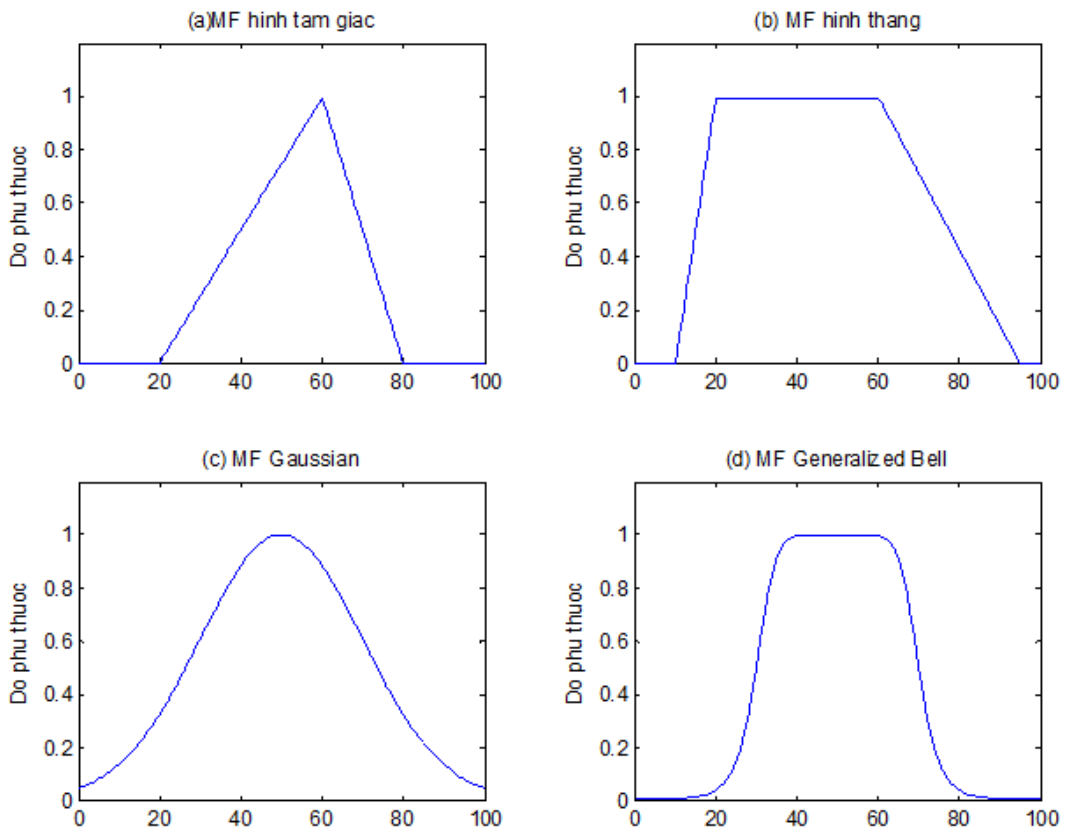
suất thống kê không chỉ ra được sự chặt chẽ đó. Một số nhà khoa học còn phê phán về việc xác định giá trị của hàm thuộc trong logic mờ được xây dựng chưa đầy đủ và thiếu sự chặt chẽ về nền tảng toán học so với các lý thuyết toán học.

Giới thiệu tập mờ (Fuzzy set)

Tập mờ [5,6,7] được xây dựng dựa trên lý thuyết mờ. Khi đó, mỗi tập mờ xây dựng trên một tập nền X, mỗi giá trị x của tập nền X được xác định bởi một giá trị hàm thuộc μ . Do vậy, một tập mờ A được xác định dựa trên cặp (x, μ) như sau:

$$A = \{x, \mu(x) \mid x \in X, \mu \in [0,1]\} \quad (1.1)$$

Trong đó : $\mu(x)$ gọi là hàm thuộc của giá trị x của tập mờ A. Khi đó, được xác định thông qua một hàm để ánh xạ giá trị của tập nền X với các giá trị của hàm thuộc. Ánh xạ được xác định từ 1 phần tử x trong tập nền X tới giá trị hàm thuộc được xác định giá trị trong đoạn [0,1].



Hình 1. 1: Hàm thuộc cơ bản trong logic mờ

Khi đó, một tập mờ được xây dựng dựa trên hai yếu tố cơ bản là: không gian nền và hàm thuộc. Mỗi hàm thuộc được xây dựng dựa trên ý nghĩa của từng bài toán, vấn đề trong thực tế cho phù hợp. Người ta, có thể lựa chọn một số hàm thuộc cơ bản hoặc hàm thuộc được định nghĩa mới. Tuy nhiên, các hàm thuộc phản ánh ý nghĩa thực tiễn của bài toán đề ra.

Hàm thuộc khi xây dựng dựa trên tập nền của bài toán, vấn đề ngữ nghĩa để xác định việc ảnh hưởng đến phân phối giá trị của biến ngữ nghĩa. Một số hàm thuộc cơ bản như sau : hàm phân bố xác suất, hàm lượng giác (hàm sin, hàm cosin,...), hàm phân phối chuẩn, (thể hiện trong hình 1.1).

Để lựa chọn hàm thuộc phù hợp, người ta lựa chọn hàm thuộc phù hợp dựa trên bài toán thực tế. Ngoài các hàm thuộc cơ bản đã có, trong một số ngôn ngữ cũng hỗ trợ một số hàm thuộc có sẵn. Trong Matlab cũng xây dựng một số hàm thuộc trong thư viện Fuzzy. Các yếu tố cơ bản khi lựa chọn hàm thuộc: phân tích thống kê dữ liệu, hiểu ý nghĩa dữ liệu. Tuy nhiên, người dùng có thể xây dựng hàm thuộc phù hợp với bài toán, đảm bảo giá trị của hàm thuộc luôn thuộc trong $[0,1]$. Ví dụ có thể xây dựng hàm thuộc như sau :

$$A = \begin{cases} \sum_{x_i \in X} \mu_A(x_i) & \text{Nếu } X \text{ là tập hợp các đối tượng rời rạc} \\ \int_X \mu_A(x) / x & \text{Nếu } X \text{ là không gian liên tục} \end{cases} \quad (1.2)$$

Các phép toán trên tập mờ

Dựa trên lý thuyết mờ, các phép toán cơ bản của tập mờ được xây dựng dựa trên tập nền và hàm thuộc. Những phép toán cơ bản của tập mờ cũng là các phép toán cơ bản của tập hợp như: phép giao, phép hợp, phép phủ định.

Phép giao: giao giữa 2 tập mờ: A và B trên cùng một tập nền X, được xác định dựa trên một ánh xạ của 2 tập hợp thông qua giá trị của hàm thuộc được xác định bởi công thức (1.3)

$$\mu_{A \cap B}(x) = T[\mu_A(x), \mu_B(x)] \quad (1.3)$$

Khi đó, một tiêu chuẩn T được xác định cho điểm giao nhau của phép toán mờ, tiêu chuẩn T thỏa mãn các yêu cầu về một toán hạng chuẩn.

Toán hạng chuẩn T xác định là ánh xạ $T(\bullet)$ thoả mãn các điều kiện sau:

$$\text{Điểm biên: } T(e, 1) = T(1, e) = e ; T(0, 0) = 0 \quad (1.4)$$

$$\text{Đơn điệu: } T(e, f) \leq T(m, n) \text{ nếu } e \leq m \text{ và } f \leq n \quad (1.5)$$

$$\text{Giao hoán: } T(e, f) = T(f, e) \quad (1.6)$$

$$\text{Kết hợp: } T(e, T(f, m)) = T(T(e, f), m) \quad (1.7)$$

Mỗi điều kiện khác nhau dẫn đến tác động khác nhau của tập mờ. Khi đó ý nghĩa của các điều kiện như sau :

- Điểm biên tác động vùng danh giới và giới hạn của tập mờ.
- Đơn điệu tác động đến tính liên thuộc của các giá trị mờ.
- Giao hoán chỉ ra các giá trị mờ không ảnh hưởng đến thứ tự của chúng khi xuất hiện.
- Kết hợp chỉ ra sự kết hợp giữa các phần tử theo từng cặp.

Dựa trên cơ sở đó, một số phép toán giao thoả mãn chuẩn T-norm thường được sử dụng như sau:

$$\text{Min} \quad T(e,f) = \min(e,f) \quad (1.8)$$

$$\text{Dạng tích: } T(e,f) = e * f \quad (1.9)$$

$$\text{Chuẩn max : } T(e,f) = \max\{e+f-1, 0\} \quad (1.10)$$

$$\text{Chuẩn min} \quad T(e, f) = \begin{cases} \min(e, f) & \text{nếu } e + f > 1 \\ 0 & \text{nếu } e + f \leq 1 \end{cases} \quad (1.11)$$

$$\text{T chuẩn yếu: } Z(x, y) = \begin{cases} \min(x, y) & \text{max}(x, y) = 1 \\ 0 & \text{max}(x, y) < 1 \end{cases} \quad (1.12)$$

Phép hợp : Giống như giao điểm mờ, phép hợp dựa trên sự kết hợp mờ với ánh xạ nhị phân xác định trên S.

$$\mu_{A \cup B}(x) = S[\mu_A(x), \mu_B(x)] \quad (1.13)$$

Những giá trị của phép hợp mờ thường được xác định là những phép toán dựa trên các tiêu chuẩn của phép giao, thoả mãn các tiêu chuẩn cơ bản trong tiêu chuẩn S.

Tiêu chuẩn S là ánh xạ của phép toán 2 ngôi $S(\bullet)$ thoả mãn:

$$\text{Điểm biên: } S(e, 0) = S(0, e) = e; \quad S(1, 1) = 1; \quad (1.14)$$

$$\text{Đơn điệu: } S(e, f) \leq S(m, n) \text{ nếu } e \leq m \text{ và } f \leq n \quad (1.15)$$

$$\text{Giao hoán: } S(e, f) = S(m, n) \quad (1.16)$$

$$\text{Kết hợp: } S(e, S(f, m)) = S(S(e, f), m) \quad (1.17)$$

Từ đó, Zadeh xây dựng các phép toán dựa trên tiêu chuẩn S cho phép toán giao, cụ thể như sau:

$$\text{Phép max} \quad S(e, f) = \max(e, f) \quad (1.18)$$

$$\text{Phép tích:} \quad S(e, f) = e + f - e * f \quad (1.19)$$

$$\text{Chuẩn min} \quad T(e, f) = \min\{e + f, 1\} \quad (1.20)$$

$$\text{Chuẩn max} \quad S(x, y) = \begin{cases} \max(x, y) & x + y < 1 \\ 1 & x + y \geq 1 \end{cases} \quad (1.21)$$

$$\text{S chuẩn yếu:} \quad Z(x, y) = \begin{cases} \max(x, y) & \min(x, y) = 0 \\ 1 & \min(x, y) \neq 0 \end{cases} \quad (1.22)$$

Phép phủ định : là một phép toán cơ bản của logic. Được sử dụng để suy rộng các toán tử. Phép phủ định thoả mãn điều kiện như sau:

Hàm $G: [0,1] \rightarrow [0,1]$ được gọi là hàm phủ định nếu điều kiện sau đây được thoả mãn:

$$\text{Điều kiện biên: } G(0)=1 \text{ và } G(1)=0$$

$$\text{Đơn điệu: } G(A) \geq G(B) \text{ nếu } A \leq B$$

Nếu $G(G(A)) = A$ thì phép phủ định này gọi là phủ định chặt.

Trên cơ sở đó, người ta thường sử dụng một số phép toán phủ định như sau:

$$\text{Zadeh: } G(x) = 1 - x$$

$$\text{Sugeno: } G_p(x) = (1 - x) / (x * p) \quad (1.24)$$

$$\text{Yager: } G_q(x) G_q(x) = (1 - x^q)^{1/q} \quad (1.25)$$

1.3. Suy diễn mờ (Fuzzy Inference)

Suy diễn mờ [8] là một phương pháp diễn giải các giá trị trong vector đầu vào và dựa trên một số bộ quy tắc, gán giá trị cho vector đầu ra. Trong logic mờ, sự thật của bất kỳ tuyên bố nào cũng trở thành vấn đề ở mức độ nào đó.

Suy diễn mờ là quá trình xây dựng ánh xạ từ một đầu vào cho trước đến một đầu ra bằng cách sử dụng logic mờ. Sau đó, việc lập bản đồ cung cấp một cơ sở để từ đó có thể đưa ra các quyết định hoặc các mô hình sáng suốt. Quá trình suy diễn mờ liên quan đến tất cả các phần được mô tả cho đến nay, tức là các hàm liên thuộc, toán tử logic mờ và các quy tắc IF-THEN. Hai kiểu hệ thống suy luận mờ chính có thể được thực hiện: kiểu Mamdani (1977) và kiểu Sugeno (1985).

Một phương pháp suy luận để xây dựng một tập các luật mờ do dạng IF-THEN mờ được coi như chân lý thì được gọi là phương pháp suy diễn mờ (hay được gọi là phương pháp suy diễn xấp xỉ).

Luật được xây dựng trên cơ sở phép kéo theo của logic, khi đó suy luận được xây dựng dựa trên luật suy diễn trong logic. Trong logic mệnh đề mỗi biến mệnh đề biểu diễn cho một sự kiện. Khi đó dựa vào phép suy diễn Modus Ponens của logic. Từ 2 mệnh đề A và B, người ta xây dựng một luật $R: A \rightarrow B$.

Khi đó luật Modus Ponens thể hiện luật mờ như sau:

Sự kiện giả thiết: $x \text{ is } A$

Sự kiện hệ quả là: $y \text{ is } B$

Luật R được xây dựng như sau : IF $x \text{ is } A$ THEN $y \text{ is } B$

Dựa theo cách suy diễn thông thường của con người, luật modus ponens sử dụng như dựa trên cách tính xấp xỉ, được minh họa như sau:

Sự kiện giả thiết: $x \text{ is } A'$

Sự kiện hệ quả là: $y \text{ is } B'$

Luật R' được xây dựng như sau : IF x is A' THEN y is B'

Với A', A được xây dựng trên cùng 1 tập nền X và B, B' xây dựng trên cùng 1 tập nền Y. Do vậy, các giá trị mờ A, A', B, B' được xác định trên các không gian nền tương ứng. Phép toán suy diễn dựa trên giá trị của hàm thuộc suy diễn B' được dựa trên việc lấy max và min của các giá trị hàm thuộc A, A', B như công thức (1.26).

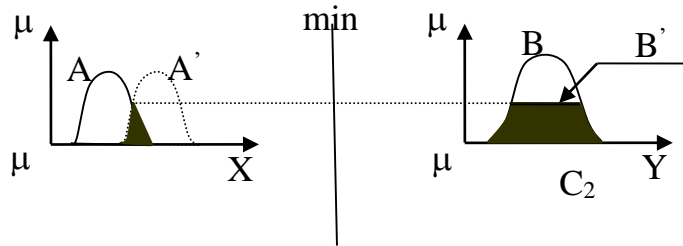
$$\mu_{B'}(y) = \max_x \min[\mu_A(x), \mu_{R'}(x, y)] = \vee_x [\mu_A(x) \wedge \mu_{R'}(x, y)] \quad (1.26)$$

Người ta có thể biểu diễn việc suy diễn dưới dạng (1.27)

$$B' = A' \circ R = A' \circ (A \rightarrow B) \quad (1.27)$$

Dựa vào các luật suy diễn cơ bản của logic người ta xây dựng lên các phép suy diễn với luật mờ:

- Suy diễn từ một tiên đề giả thiết với một luật



Hình 1. 2: Mô hình suy luận mờ với một giả thiết và một luật

Việc thể hiện minh họa trên hình 1.2. Việc xác định giá trị của hàm thuộc được xác định bởi công thức (1.28):

$$\mu_{B'}(y) = [\vee_x (\mu_A(x) \wedge \mu_{A'}(x))] \wedge \mu_B(y) = w \wedge \mu_B(y) \quad (1.28)$$

$$w = \max(\mu_A(x) \wedge \mu_{A'}(x))$$

- Phương pháp suy luận với nhiều giả thiết và một luật

Nếu luật mờ IF-THEN được xác định với hai giả thiết được xác định như sau : “IF x is A and y is B THEN z is C” các thành phần được xác định suy diễn như sau:

Sự kiện giả thiết: x is A' và y is B'

Luật mờ : IF x is A and y is B THEN z is C

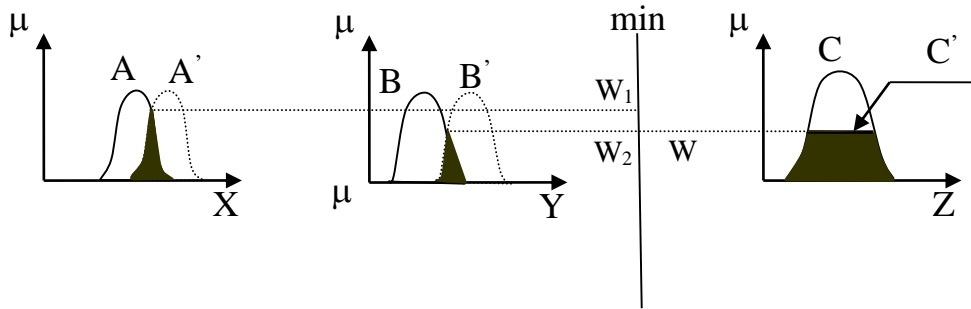
Kết luận suy diễn : z is C'

Luật mờ trong giả thiết 2 có thể đưa về dạng: $AxB \rightarrow C$ từ đó ta tính được:

$$C' = (A'xB') \circ R = A' \circ (AxB \rightarrow C) = [A' \circ (A \rightarrow C)] \cap [B' \circ (B \rightarrow C)] \quad (1.29)$$

$$\begin{aligned} \mu_{C'}(z) &= \vee_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \left\{ \vee_x [\mu_{A'}(x) \wedge \mu_A(x)] \right\} \wedge \left\{ \vee_y [\mu_{B'}(y) \wedge \mu_B(y)] \right\} \wedge \mu_C(z) = (w_1 \wedge w_2) \vee \wedge \mu_C(z) \end{aligned} \quad (1.30)$$

Với $w_1 = \vee_x [\mu_{A'}(x) \wedge \mu_A(x)]$ và $w_2 = \vee_y [\mu_{B'}(y) \wedge \mu_B(y)]$



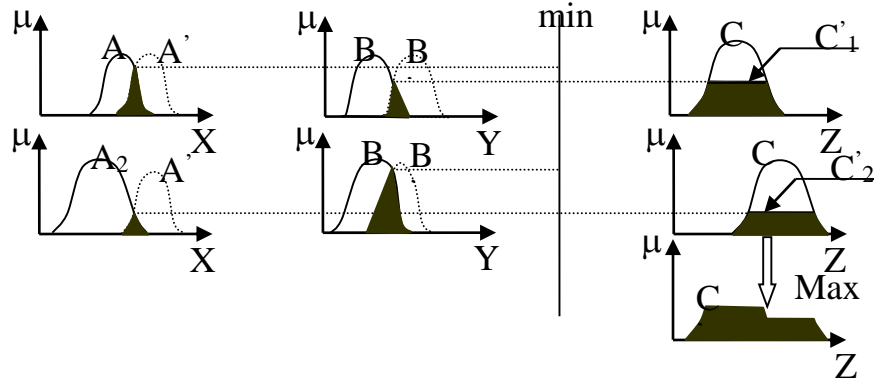
Hình 1. 3: Mô hình suy luận mờ nhiều giả thiết và một luật

Ta lại có:

$$\begin{aligned} \mu_{C'}(z) &= \vee_{x,y} [\mu_{A'}(x) \wedge \mu_{B'}(y)] \wedge [\mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z)] \\ &= \mu_{A' \circ (A \rightarrow C)}(y) \wedge \mu_{B' \circ (B \rightarrow C)}(y) \end{aligned} \quad (1.31)$$

Từ công thức trên ta thấy rằng, kết quả C' có thể được xem như giao của hai biểu thức $C_1' = A' \circ (A \rightarrow C)$ và $C_2' = B' \circ (B \rightarrow C)$. Mỗi một biểu thức liên quan đến phép suy diễn mờ trong trường hợp một luật mờ và một tiền đề đã xét ở trên.

- Suy luận nhiều luật mờ với nhiều tiền đề



Hình 1. 4: Mô hình suy luận mờ có giả thiết với hai luật

Sự diễn dịch của luật mờ phức hợp được tiến hành như hợp của các quan hệ mờ tương ứng với luật mờ. Do vậy ta có:

Sự kiện giả thiết: x is A' và y is B'

Luật mờ 1 : IF x is A_1 and y is B_1 THEN z is C_1

Luật mờ 2 : IF x is A_2 and y is B_2 THEN z is C_2

Kết luận của suy luận: z là C'

Chúng ta có thể sử dụng suy diễn mờ như một phương pháp suy luận để tìm được kết quả của tập mờ đầu ra C'

Để xác định thuật toán suy luận mờ, gọi $R_1 = A_1 \times B_1 \rightarrow C_1$ và $R_2 = A_2 \times B_2 \rightarrow C_2$. Do phép hợp thành max-min có tính chất phân phối với phép \cup nên:

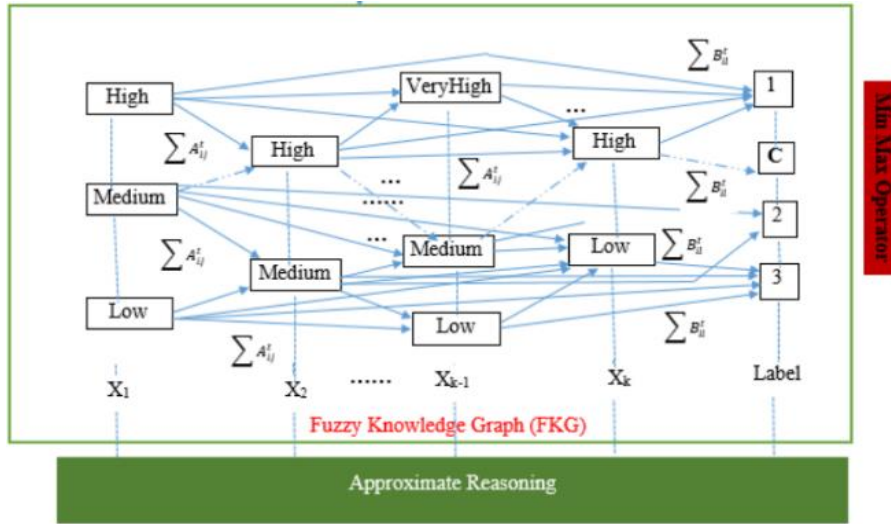
$$C' = (A' \times B') \circ (R_1 \cup R_2) = [A' \times B'] \circ R_1 \cup [A' \times B'] \circ R_2 = C'_1 \cup C'_2 \quad (1.32)$$

Với C'_1 và C'_2 là tập mờ được suy diễn từ luật 1 và 2.

1.4. Đồ thị tri thức mờ

Đồ thị tri thức mờ [9] (FKG) được nhóm nghiên cứu Lê Hoàng Sơn đề xuất năm 2020 và mô hình được xây dựng dựa trên luật mờ, phát triển với mờ phức [10]. Khi đó đồ thị tri thức mờ dựa trên nền tảng của lý thuyết đồ thị.

Các đỉnh tương ứng với các giá trị ngôn ngữ của thuộc tính và giá trị của nhãn đầu ra. Các đỉnh nối với nhau thành các cung tương ứng. Trong hình 1.5 dưới đây [10] minh họa về đồ thị tri thức mờ xây dựng dựa trên một cơ sở luật mờ (*fuzzy rule base*).



Hình 1. 5: Đồ thị tri thức mờ.

Trong [10], đồ thị tri thức mờ sử dụng để suy diễn và dự báo nhãn đầu ra của các mẫu mới đưa vào để kiểm tra. Khi xây dựng đồ thị tri thức mờ ngoài các đỉnh là các giá trị ngôn ngữ của thuộc tính thì các cạnh được xây dựng dựa trên ma trận trọng số các cạnh A, B thông qua các công thức (1.33) và (1.34).

$$A_{ij}^t = \frac{|X_i \rightarrow X_j \text{ trong luật } t|}{|R|} \tag{1.33}$$

trong đó $t = \overline{1, k}, 1 \leq i < j \leq m$, và

$$B_{il}^t = \left(\sum A_{ij}^t \right) * \frac{|X_i \rightarrow \text{lable } l \text{ trong luật } t|}{|R|} \tag{1.34}$$

trong đó $t = \overline{1, k}, 1 \leq i < j \leq m, l = \overline{1, C}$.

Để xác định nhãn của phân lớp dựa trên đồ thị tri thức mờ dựa vào việc tính toán giá trị ngôn ngữ của dữ liệu xác định nhãn bởi công thức (1.35):

$$C_{il} = \sum_t B_{il}^t \quad (1.35)$$

Sau đó sử dụng công thức tính D để xác định khả năng của từng nhãn với mẫu dữ liệu đang cần xác định theo công thức (1.36).

$$D_l = \min_{i=1,k} (C_{il}) + \max_{i=1,k} (C_{il}) \quad (1.36)$$

Cuối cùng nhãn được xác định bởi công thức (1.37) dựa trên việc sử dụng công thức max.

$$Label = p \text{ if } D_p = \max_{i=1,C} (D_l) \quad (1.37)$$

Khi đó, đồ thị tri thức mờ đã tập trung vào việc mô phỏng các luật mờ dựa trên đồ thị tri thức mờ, quá trình suy diễn trên đồ thị tri thức mờ để xác định các nhãn. Do vậy, đồ thị tri thức mờ tập trung vào việc giải các bài toán phân lớp dữ liệu.

1.5. Ngôn ngữ MATLAB

Ngôn ngữ MATLAB được sử dụng nhiều trong các bài toán hỗ trợ tính toán, nó hỗ trợ nhiều trong tính toán và xử lý dữ liệu. Matlab là ngôn ngữ cung cấp các thư viện về: logic mờ, tính toán số liệu trên ma trận, thiết kế giao diện người dùng, vẽ đồ thị,... Khi đó, việc sử dụng ngôn ngữ Matlab giúp cho việc tính toán của bài toán học máy, đặc biệt các bài toán trong logic mờ được dễ dàng hơn. Do vậy, trong luận văn này em lựa chọn Matlab là ngôn ngữ lập trình cho việc mô phỏng demo của chương trình.

MATLAB thường được dùng trong mô phỏng và tính toán của các lĩnh vực: thiết kế điều khiển tự động, truyền thông, xử lý tín hiệu và ảnh, đo lường kiểm tra, tính toán thuật toán, phân tích số liệu. Đặc biệt, trong các nghiên cứu cần hỗ trợ tính toán thì các nhà khoa học cũng hay lựa chọn Matlab để mô phỏng. Với lý thuyết đồ thị thì việc tính toán thường có khối lượng lớn cần được xử lý.

Các kiểu dữ liệu trong Matlab cũng rất phong phú, có hỗ trợ nhiều với các loại dữ liệu số như: số nguyên, số thực, số phức. Đồng thời với các loại

dữ liệu thiếu, không xác định Matlab cũng hỗ trợ để xử lý. Các phép toán cơ bản của Matlab sử dụng để tính toán là: +, -, *, / hoặc \ và ^ (số mũ).

Việc xử lý ma trận matlab hỗ trợ việc khai báo, xử lý và tính toán. Việc xử lý dữ liệu thuộc tính để thể hiện thông qua mảng 2 chiều: mỗi cột là 1 thuộc tính và mỗi dòng là 1 bản ghi. Đồng thời, với đồ thị biểu diễn về dạng ma trận kề, các đỉnh của đồ thị tương ứng là các dòng (cột) của ma trận, các cạnh của đồ thị thể hiện qua trọng số của ma trận.

Một số hàm cơ bản về xử lý ảnh trong MATLAB:

- Đọc và ghi các dữ liệu ảnh:
 - Hàm đọc ảnh `imread()`: đọc tất cả các loại định dạng file ảnh cơ bản, các ảnh được số hóa và biểu diễn dưới dạng ma trận. Mỗi điểm ảnh là 1 vectơ biểu diễn, tùy vào mỗi loại ảnh mà số lượng các thuộc tính của vectơ là bao nhiêu. Trong Matlab, có các hàm hỗ trợ đọc ảnh và chuyển đổi về ma trận, các ma trận được lưu dưới dạng số. Khi đó việc xử lý ảnh dựa trên việc tính toán xử lý trên ma trận biểu diễn của ảnh.
 - Hàm ghi ảnh `imwrite` cho phép chuyển đổi ma trận số sang một định dạng ảnh cơ bản để lưu.
- Các hàm hiển thị ảnh:
 - Hàm `imshow()` và `imagesc()` : Để hiển thị hình ảnh trong Matlab ta dùng 2 lệnh **`imshow()`** và **`imagesc()`**
 - + Hàm `imshow()` dùng để hiển thị ảnh với 8-bit biểu diễn, nó cho phép hiển thị các ảnh số về dạng hình ảnh cơ bản.
 - + Hàm `imagesc()` dùng để hiển thị hình ảnh kèm các trục của đồ thị dựa trên các giá trị màu của hệ màu RGB.
 - Hàm `rgb2gray()` cho phép chuyển đổi ảnh biểu diễn dạng RGB sang ảnh đen-trắng bằng cách chuyển các giá trị màu sắc sang các giá trị đen trắng, nhưng vẫn giữ nguyên độ sáng.
 - Hàm `imadjust()`: sử dụng để điều chỉnh cường độ của ảnh. Hình ảnh đầu ra được làm tăng độ tương phản với câu lệnh này.
- Một số hàm đọc file
 - `importdata ('tên file');`

- dlmread ('tên file') đuôi .txt,.mat;
- xlsread('file name'); đọc file dạng xlsx;
- load 'tên file';
- Một số hàm toán học :
 - sqrt(x) căn bậc hai.
 - log(x) logarit cơ số 10.
 - sign(x) dấu của x.
 - round(x) hàm làm tròn.
 - rand() hàm lấy giá trị ngẫu nhiên.
 - cos(x), sin(x), tan(x).
- Một số hàm cơ bản:
 - Hàm clear dùng để xóa biến này khỏi không gian làm việc, khi đó các giá trị của biến sẽ bị loại bỏ.
 - Hàm Save: Lưu dữ liệu của biến vào file dưới dạng *.mat
 - Hàm load(): dùng để tải dữ liệu từ file với định dạng *.mat lên không gian làm việc.
 - Length(): Chiều dài của vector
 - Size() : Kích thước của ma trận
 - Hàm Rand(): tạo ma trận ngẫu nhiên với các giá trị trong phân bố chuẩn [0,1].
 - Biểu thức rẽ nhánh:
 - + Cấu trúc rẽ 1 nhánh **If...end:**

```

if <biểu thức điều kiện>
    <Câu lệnh1>
    <Câu lệnh 2>
    ...
end

```
 - + Cấu trúc rẽ hai nhánh **If...else...end:**

```

if <biểu thức điều kiện>
    <Các câu lệnh với biểu thức điều kiện đúng>

```

else

<Các câu lệnh với biểu thức điều kiện sai>

end

+ Cấu trúc vòng lặp **For**

for <biến điều khiển> = <giá trị bắt đầu> : <giá trị kết thúc>

<Các câu lệnh trong vòng lặp>

end

- Xây dựng hàm m-file là 1 thủ tục con do người dùng xây dựng. Nó đóng vai trò là 1 hàm, 1 thủ tục mà người dùng định nghĩa lên. Khi sử dụng thì truyền tham số cho chúng sẽ thực thi và trả về giá trị.
- Một hàm sẽ có dạng tổng quát như sau:

function [<các biến đầu ra>] = **function_name**(<các tham số đầu vào>)

Biến đầu ra được đặt trong dấu []:

- Các biến đầu ra được liệt kê theo thứ tự khi nhận giá trị tương ứng và được cách nhau bởi dấu “,”.
- Hàm có thể không có giá trị trả về thì thực thi 1 đoạn câu lệnh
- Nếu có 1 biến đầu ra duy nhất không cần dấu [].

Tham số đầu vào được đặt trong dấu ():

- Tham số đầu vào ngăn cách nhau bởi dấu “,”
- Hàm có thể không có tham số đầu vào.
- Một số hàm khác
 - int2str(i) : chuyển kiểu sang dạng số.
 - num2str(i) : chuyển từ dạng số sang dạng mảng
 - disp(x) : hiển thị giá trị x.

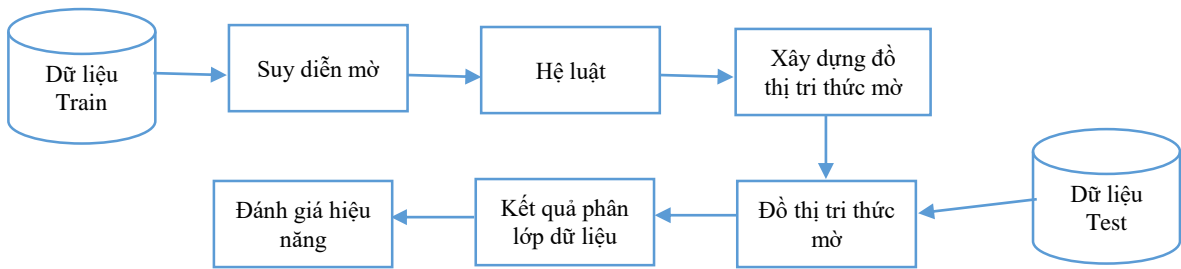
1.6. Tổng kết chương 1

Trong chương 1, em đã trình bày về các kiến thức lý thuyết sử dụng trong quá trình làm luận văn. Các kiến thức lý thuyết cơ sở gồm: lý thuyết mờ, lý thuyết đồ thị, đồ thị tri thức mờ, ngôn ngữ sử dụng để cài đặt thuật toán trong luận văn.

CHƯƠNG 2. ĐỒ THỊ TRI THỨC MỜ

2.1. Mô hình đồ thị tri thức mờ trong phân lớp dữ liệu

Trong chương này, em trình bày mô hình suy diễn mờ trên đồ thị tri thức mờ để thực hiện việc phân lớp dữ liệu (thể hiện Hình 2.1).



Hình 2. 1: Mô hình đồ thị tri thức mờ trong phân lớp dữ liệu

- **Bước 1.** Xây dựng luật mờ từ phương pháp suy diễn mờ (fuzzification), biểu diễn về dạng IF-THEN.
- **Bước 2.** Từ hệ luật xây dựng đồ thị tri thức mờ
- **Bước 3.** Suy diễn trên đồ thị tri thức mờ.
- **Bước 4.** Đánh giá hiệu năng mô hình dựa trên các độ đo phân lớp dữ liệu.

Chi tiết của từng phần sẽ được trình bày trong các mục tiếp theo.

2.2. Xây dựng luật mờ

Dựa trên lý thuyết mờ, xây dựng luật mờ từ dữ liệu [11,12,13,14]. Một thuộc tính được mờ hóa dựa trên một tập nền. Quá trình mờ hóa là ánh xạ từ tập nền X vào $[0,1]$ dựa trên một ánh xạ f . Khi đó một giá trị mờ A được biểu diễn dưới dạng đại số như sau :

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \quad (2.1)$$

Khi đó một giá trị mờ B được suy diễn dựa trên nguyên lý suy rộng thì B là ảnh của A trên ánh xạ f được xác định bởi (2.2).

$$B = f(a) = \frac{\mu_A(x_1)}{y_1} + \frac{\mu_A(x_2)}{y_2} + \dots + \frac{\mu_A(x_n)}{y_n}. \text{ với } y_i=f(x_i), i=1 \dots n \quad (2.2)$$

Khi đó, tập mờ B được xác định dựa trên các giá trị của ánh xạ f với tập nền X. Ứng với mỗi giá trị $x_1, x_2 \in X, x_1 \neq x_2$ mà $f(x_1)=f(x_2)=y^*, y^* \in Y$ thì mức độ phụ thuộc của B là lớn nhất tại $y=y^*$ vào A tại các điểm $x=x_1$ và $x=x_2$, khi đó người ta có thể xác định giá trị hàm thuộc của giá trị mờ B dựa trên công thức (2.3)

$$\mu_B(y) = \max_{x=f^{-1}(y)} (\mu_A(x)) \quad (2.3)$$

Để tổng quát ta xây dựng một hàm f là một ánh xạ từ không gian n chiều $X_1 \times X_2 \times \dots \times X_n$ và A_1, A_2, \dots, A_n là n tập mờ trên tập nền X_1, X_2, \dots, X_n , khi đó nguyên lý suy rộng được xây dựng thông qua ánh xạ f để xác định tập mờ B là:

$$\mu_B(y) = \begin{cases} \max_{(x_1, x_2, \dots, x_n), (x_1, x_2, \dots, x_n) = f^{-1}(y)} [\min_i \mu_{A_i}(x_i)], & \text{if } f^{-1}(y) \neq \emptyset \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases} \quad (2.4)$$

Quan hệ trong tập mờ.

Quan hệ hai ngôi trong tập mờ: một tập mờ được xây dựng dựa trên không gian $X \times Y$ khi ánh xạ mỗi phần tử của không gian này sẽ có một giá trị hàm thuộc trong $[0, 1]$ được xác định.

Khi đó 2 tập X và Y là 2 tập nền của tập mờ thì :

$$R = \{((x, y), \mu_R(x, y)) \mid (x, y) \in X \times Y\} \quad (2.5)$$

Được xác định là quan hệ hai ngôi trong tập mờ trên tập nền $X \times Y$.

Khi đó quan hệ hai ngôi trong tập mờ thường được xây dựng như sau:

Luật mờ : “IF x is A THEN y is B” được sử dụng thường xuyên trong hệ suy diễn mờ. Quan hệ trong tập mờ trên các không gian nền khác nhau được kết hợp thông qua phép hợp thành (khi đó gọi là luật hợp thành). Phép hợp thành có thể dùng nhiều cách khác nhau cho quan hệ trong tập mờ, phương pháp thường dùng được Zadeh xây dựng là luật hợp thành max-min.

Gọi R_1 và R_2 là hai quan hệ trong tập mờ trên tập nền $X \times Y$ và $Y \times Z$, khi đó luật hợp thành Max-min của R_1 và R_2 là một tập mờ xây dựng như sau:

$$R_1 \circ R_2 = \left\{ (x, z), \max_y \min[\mu_{R_1}(x, y), \mu_{R_2}(y, z)] \mid x \in X, y \in Y, z \in Z \right\} \quad (2.6)$$

Hoặc tương đương :

$$\mu_{R_1 \circ R_2}(x, z) = \max_y \min[\mu_{R_1}(x, y), \mu_{R_2}(y, z)] = \bigvee_y [\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)] \quad (2.7)$$

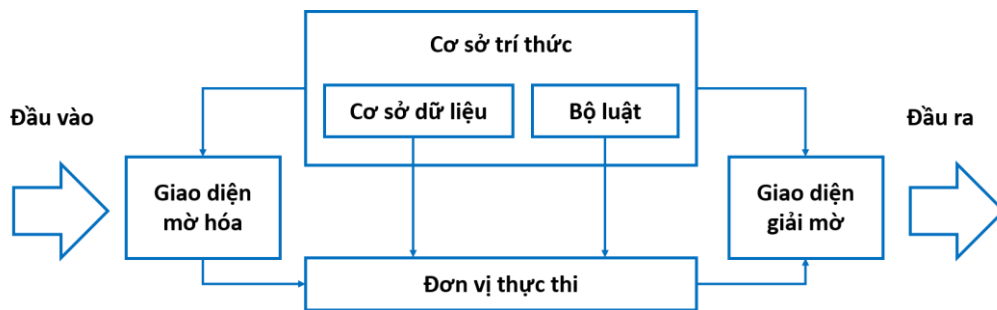
Khi đó \bigvee phép toán lấy max và \wedge phép toán lấy min. Khi R_1 và R_2 được xác định dạng ma trận, phép tính toán $R_1 \circ R_2$ xác định là nhân ma trận cho phép thay thế phép max và min, do vậy người ta gọi đó là luật hợp thành dựa trên các phép toán max-min của luật mờ.

Suy diễn mờ là cách thức liên kết các tri thức trong cơ sở tri thức với các sự kiện trong cơ sở sự kiện để đưa ra các tri thức mới.

Không có một phương pháp suy diễn duy nhất cho mọi loại tri thức, phương pháp suy diễn khi xây dựng phụ thuộc vào cách thức biểu diễn tri thức trong cơ sở tri thức. Khi xây dựng các hệ chuyên gia, hệ suy diễn mờ là một cách hay được sử dụng để suy diễn.

Trong trường hợp thông tin không đầy đủ hoặc không chính xác thì hệ suy diễn mờ tỏ ra hiệu quả.

Hệ suy diễn mờ phần lớn được xây dựng với cấu trúc như hình 2.2



Hình 2. 2: Sơ đồ tổng quan hệ suy diễn mờ

- Phần giao diện mờ hóa thực hiện việc chuyển đổi các giá trị đầu vào thành các giá trị của biến ngôn ngữ.
- Phần 2 là cơ sở tri thức bao gồm:
 - Các tri thức trong cơ sở tri thức được lưu trữ trong cơ sở dữ liệu xác định các giá trị hàm thuộc của giá trị mờ.
 - Hệ luật: là tập hợp các luật mờ có dạng: IF – THEN
- Đơn vị thực thi: có nhiệm vụ thực hiện quá trình suy diễn từ các luật mờ trong hệ luật.

- Giao diện giải mờ: sau khi thu được kết quả suy diễn là các giá trị của biến ngôn ngữ, cần chuyển đổi các giá trị này thành các giá trị rõ làm kết quả đầu ra của hệ suy diễn mờ.

Các phương pháp suy diễn cơ bản đã được trình bày trong chương 1 với các hệ suy diễn cơ bản.

2.3. Biểu diễn đồ thị tri thức mờ từ luật mờ

Trên cơ sở các luật đã được xây dựng trong mục 2.2. Trong mục này, em trình bày cách thức xây dựng đồ thị từ tập luật.

Cho tập luật được xác định bởi mục 2.2 (các luật thể hiện trong bảng 2.1). Xây dựng Đồ thị tri thức mờ (FKG) từ hệ cơ sở luật mờ [15].

Bảng 2. 1: Hệ luật mờ

Rule	X_1	X_2	...	X_{m-i}	X_m	Label
R_1	High	Medium	...	High	High	1
R_2	High	Medium	...	Low	Medium	2
...
R_{k-1}	Low	High	...	Medium	Low	2
R_k	Low	Medium	...	Low	Low	3
Giá trị biến ngôn ngữ	High Medium Low	High Medium Low	...	High Medium Low	High Medium Low	1, 2, 3... C

Một đồ thị vô hướng $G = \langle A, C \rangle$, trong đó $A = \{ a_1, a_2, \dots, a_n \}$ là tập các đỉnh của đồ thị G , $C = \{ c_1, c_2, \dots, c_m \}$ là tập các cạnh của đồ thị G . Khi biểu diễn trên máy tính ta cần xây dựng một ma trận kề của đồ thị G . Gọi ma trận kề là E , khi đó E được xác định là:

$$E = \{ e_{ij} : i, j = \overline{1..n} \}$$

Khi đó, e_{ij} xác định như sau:

$$e_{ij} = \begin{cases} 0 & \text{nếu } (i, j) \text{ không là cạnh thuộc } C \\ 1 & \text{nếu } (i, j) \text{ là cạnh thuộc } C \end{cases}$$

Đồ thị có trọng số được xác định là: $G_1 = \langle A_1, C_1 \rangle$, trong đó $A = \{ a_1, a_2, \dots, a_n \}$ là tập các đỉnh của đồ thị G_1 , $C = \{ c_1, c_2, \dots, c_m \}$ là giá trị trọng số

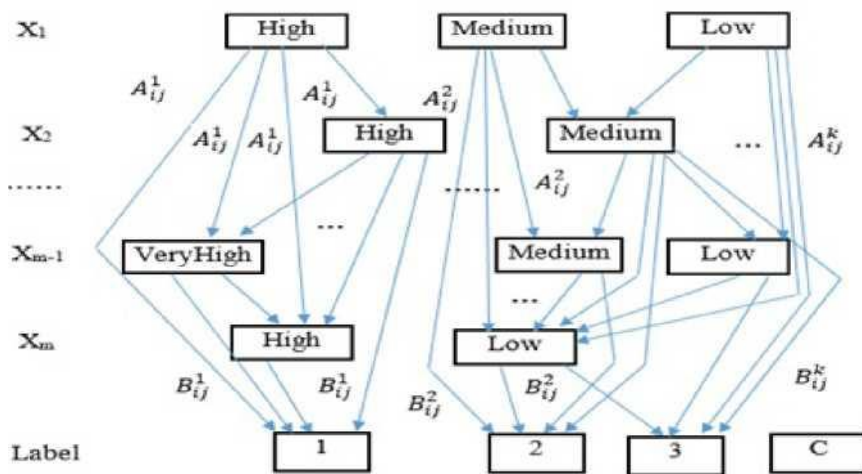
cạnh trên đồ thị G_1 .

Như vậy khi xây dựng FKG với từng luật mờ R_t dựa vào cách biểu diễn của đồ thị. Mỗi giá trị của biến ngôn ngữ của từng thuộc tính và giá trị ngôn ngữ là một đỉnh của đồ thị. Các cạnh của đồ thị được xây dựng dựa trên mối quan hệ giữa các giá trị của biến ngôn ngữ của các thuộc tính, nhân trong từng luật mờ cần biểu diễn. Khi đó một đồ thị FKG được xác định như sau: khi $t = \overline{1, k}$, với quan hệ của thuộc tính (X_i, X_j) , $1 \leq i \leq j \leq m$ trên luật R_t , được xác định như sau:

$$Value(X_i) \rightarrow Value(X_j) \tag{2.8}$$

Khi đó $Value(X_i)$ là giá trị của biến ngôn ngữ tương ứng với thuộc tính X_i . Ứng với mỗi cặp giá trị ngôn ngữ và nhân là $(X_i, Label)$, $1 \leq i \leq m$, mỗi cạnh của đồ thị là quan hệ được xác định: $Value(X_i) \rightarrow Label_t$.

Đối với các bài toán thực tế, số lượng các biến ngôn ngữ tùy thuộc vào các biến đầu vào thực tế và phân bố của chúng. Biểu diễn đồ thị tri thức các luật mờ dựa trên các giá trị ngôn ngữ của thuộc tính được xác định minh họa theo Hình 2.3 như sau:



Hình 2. 3: Biểu diễn đồ thị tri thức mờ từ luật mờ theo thuộc tính

Khi đó A_{ij}^t là trọng số của cạnh được xác định là $Value(X_i) \rightarrow Value(X_j)$ trong luật t với $t = \overline{1, k}$, $1 \leq i \leq j \leq m$, thì A_{ij}^t được tính toán theo công

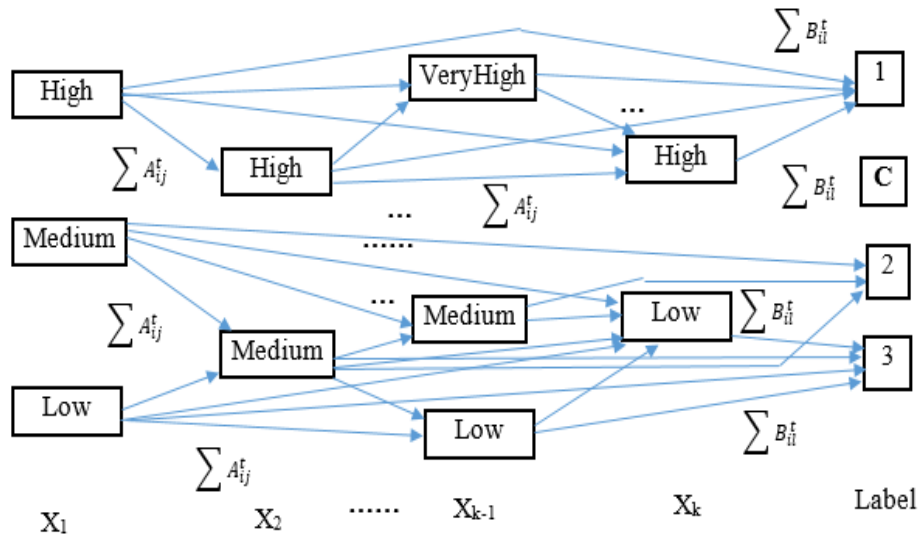
thức (2.9):

$$A_{ij}^t = \frac{|X_i \text{ quan hệ với } X_j \text{ trong luật thứ } t|}{|R|} \quad (2.9)$$

Khi đó, B_{il}^t là trọng số để biểu diễn mối quan hệ của giá trị ngôn ngữ thuộc tính X_i với nhãn l , $t = \overline{1, k}$, $1 \leq i \leq j \leq m$, $l = \overline{1, C}$ thì B_{il}^t được xác định theo công thức (2.10):

$$B_{il}^t = \left(\sum A_{ij}^t \right) * \frac{|X_i \text{ quan hệ với nhãn } l \text{ trong luật thứ } t|}{|R|} \quad (2.10)$$

Biểu diễn đồ thị của các luật mờ theo các thuộc tính được xây dựng trong Hình 2.2, sau đó biến đổi theo các giá trị biến ngôn ngữ của các thuộc tính ra đồ thị FKG như Hình 2.4, khi lưu trữ và xử lý dữ liệu của đồ thị tri thức mờ thể hiện trong một ma trận kê B.



Hình 2. 4: Biểu diễn đồ thị tri thức mờ

2.4. Suy diễn trên đồ thị tri thức mờ

Suy diễn trên đồ thị tri thức mờ [16,17,18] được thể hiện ở trong quá trình Testing. Bằng việc sử dụng Đồ thị tri thức mờ để biểu diễn luật, quá trình Testing gán các nhãn đối với từng bộ dữ liệu đầu vào Testing (Hình 2.5). Đối với mỗi bộ dữ liệu Testing đầu vào, sẽ được thực hiện mờ hóa để chuyển các biến đầu vào giá trị rõ thành các biến đầu vào giá trị mờ thông qua việc sử dụng hàm thuộc mờ. Sau đó các giá trị ngôn ngữ sẽ được xác định dựa trên hàm thuộc đầu vào tương ứng đối với mỗi thuộc tính trong bộ dữ liệu đầu

vào. Cuối cùng, thuật toán suy diễn nhanh FISA (Fast Inference Search Algorithm) được xây dựng để thực hiện suy diễn trên đồ thị tri thức mờ (FKG) để tìm nhãn của các mẫu dữ liệu cần kiểm tra.

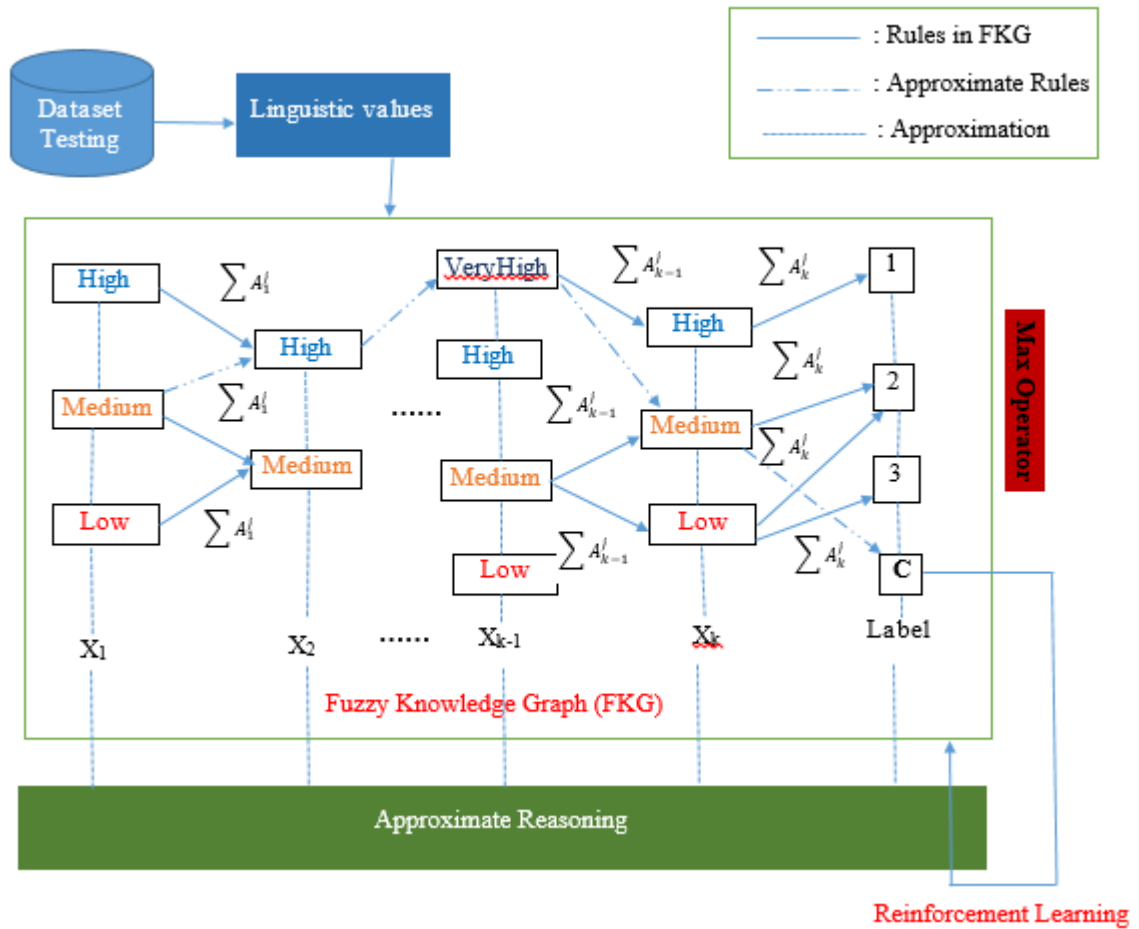
Bước 1 các giá trị ngôn ngữ của thuộc tính tương ứng với mỗi nhãn trong từng luật mờ trên đồ thị tri thức mờ FKG được tính toán theo công thức sau:

$$C_{il} = \sum_t B_{il}^t \quad (2.11)$$

Với giá trị B_{il}^t biểu diễn mối quan hệ của giá trị ngôn ngữ thuộc tính X_i với nhãn l . Ý tưởng của gán nhãn (công thức (2.12)) dựa trên suy luận xấp xỉ, điều đó có nghĩa là các giá trị ngôn ngữ của từng bản ghi mới trong bộ dữ liệu Testing được tính toán suy luận xấp xỉ theo từng thuộc tính tương ứng trong đồ thị FKG, với mục đích tìm ra nhãn tương ứng phù hợp nhất.

Với mỗi bản ghi mới thì nhãn tương ứng với từng luật được tính bởi toán tử MIN-MAX như sau:

$$D_l = \min_{i=\overline{1,k}} (C_{il}) + \max_{i=\overline{1,k}} (C_{il}) \quad (2.12)$$



Hình 2. 5: Quá trình suy diễn trên đồ thị tri thức mờ

Nhãn của mẫu mới được xác định dựa trên giá trị max của trọng số D tương ứng với từng mẫu được xác định theo công thức (2.13).

$$Label = p \text{ nếu } D_p = \max_{l=1, \dots, C} (D_l) \quad (2.13)$$

Thuật toán FISA sẽ mô tả cách thức gán nhãn đầu ra cho từng bản ghi trong bộ dữ liệu Testing.

Thuật toán FISA:

Input Tập dữ liệu kiểm tra, đồ thị tri thức mờ

Output nhãn của tập dữ liệu kiểm tra cần xác định

Begin

Bước 1: Từ dữ liệu kiểm tra thực hiện mờ hóa xác định giá trị biến ngôn

ngữ cho các thuộc tính.

Bước 2: Lập với từng mẫu kiểm tra trong bộ dữ liệu kiểm tra

Bước 3: Tính ma trận C: $C_{il} = \sum_t B_{il}^t$

Bước 4: Tính ma trận D cho từng nhãn: $D_l = \min_{i=1,\overline{k}} (C_{il}) + \max_{i=1,\overline{k}} (C_{il})$

Bước 5: Xác định nhãn cho từng mẫu kiểm tra: $Label = p$ nếu $D_p = \max_{l=1,\overline{c}} (D_l)$

End

Thuật toán FISA giúp gán nhãn cho các quy tắc mờ đầu vào sau khi đầu vào mờ. Độ phức tạp về thời gian, trong đó số mẫu trong tập dữ liệu, là số thuộc tính của mỗi mẫu; là số nhãn của mỗi thuộc tính và là số mẫu tối đa trong mỗi nhãn của mỗi thuộc tính. Mô hình đề xuất được phân tích và thấy rằng sẽ được thực thi trong một khoảng thời gian ngắn.

2.5. Các độ đo đánh giá quá trình phân lớp dữ liệu

Giá trị Accuracy [5] được tính một cách đơn giản, với việc tính tỉ lệ dự đoán đúng so với tổng số lần dự đoán. Công thức tính Accuracy như sau.

$$Accuracy = \frac{a + d}{a + b + c + d} \quad (2.14)$$

Trong đó:

- a là kết quả dự đoán true positive,
- b là kết quả dự đoán false positive,
- c là kết quả dự đoán false negative,
- d là kết quả dự đoán true negative,

Giá trị Accuracy càng cao thì kết quả dự đoán càng chính xác.

2.6. Ví dụ số về việc xây dựng và suy diễn trên đồ thị tri thức mờ

Trong mục này, em xin trình bày ví dụ để minh họa rõ cho việc xây

dụng luật và suy diễn trên đồ thị tri thức mờ.

Ví dụ 1 với bộ dữ liệu đầu vào có 10 bản ghi và 2 nhãn đầu ra như Bảng 2.2 sau:

Bảng 2. 2. Tập dữ liệu đầu vào

STT	Nhãn	TT1	TT2	TT3	TT4	TT5	TT6	TT7	TT8	TT9
1	2	5	2	2	2	2	2	3	1	1
2	2	3	1	1	1	2	2	3	1	1
3	1	8	10	10	8	7	10	9	7	1
4	2	2	1	1	1	2	1	1	1	5
5	2	4	2	1	1	2	1	2	1	1
6	2	2	1	1	1	2	1	2	1	1
7	1	5	3	3	3	2	3	4	4	1
8	1	8	7	5	10	7	9	5	5	4
9	1	7	4	6	4	6	1	4	3	1
10	1	10	7	7	6	4	10	4	1	2

Bước 1: Sinh luật mờ phức.

Để sinh luật mờ phức từ bộ dữ liệu đầu vào, ta thực hiện tuần tự:

Thực hiện sinh phần thực và phần ảo cho dữ liệu.

Đối với từng bản ghi, với từng thuộc tính, việc sinh phần thực và ảo được thực hiện theo suy tắc như sau: Phần thực được lấy chính giá trị gốc đầu vào của thuộc tính; phần ảo được tính bằng cách tính tổng phương sai theo cột và theo hàng ($\text{Var.R}(\text{bản ghi}) + \text{var.A}(\text{thuộc tính})$)

- Sử dụng thuật toán FCM để thực hiện phân cụm dữ liệu theo từng thuộc tính và từ đó có thể sinh nhãn ngôn ngữ.
- Xây dựng bộ luật ban đầu và thực hiện việc giảm luật trùng, luật yếu để được bộ luật cơ sở

Kết quả của quá trình trên, ta thu được bộ luật cơ sở như sau:

Bảng 2. 3: Bộ dữ liệu đầu vào

STT	Nhãn	TT1	TT2	TT3	TT4	TT5	TT6	TT7	TT8	TT9
1	2	M	VL	VL	VL	VL	VL	L	VL	VL
2	1	H	VH	VH	H	H	VH	VH	H	VL
3	2	VL	VL	VL	VL	VL	VL	VL	VL	M
4	2	L	VL	VL	VL	VL	VL	VL	VL	VL
5	2	VL	VL	VL	VL	VL	VL	VL	VL	VL
6	1	M	L	L	L	VL	L	M	L	VL
7	1	VH	H	H	M	L	VH	M	VL	L

Bước 2. Mờ hóa dữ liệu.

Giả sử có bộ dữ liệu test sau: $X1 = (5,1,1,1, 2,1,3,1,1)$

Ta thực hiện mờ hóa dữ liệu bằng cách sử dụng hàm mờ Gausse như sau:

$$\mu A_{p,q}(x_{p,q}) = e^{-\frac{(x_{p,q} - c_q)^2}{2\delta^2}} + i \left| \frac{(x_{p,q} - c_q)}{\delta^2} e^{-\frac{(x_{p,q} - c_q)^2}{2\delta^2}} \right|$$

Kết quả thu được giá trị mờ hóa tương ứng với mỗi thuộc tính của bộ $X1$

như sau: (1.000,0.1279) ; (1.000,0.0776); (1.000,0.0517);(1.000,0.0016);
(0.9998,0.1571); (1.000, 0.1448) ; (0.9995, 0.0075); (1.000, 0.0768); (1.000,
0.1452)

Bước 3. Tính toán trọng số của luật

Ta thực hiện tính lần lượt trọng số đối với từng luật trong bộ luật cơ sở:

Ví dụ với luật 1, ta có:

$$\begin{aligned} Re(R1) &= \text{Min}(1.0000, 1.0000, 1.0000, 1.0000, 0.9998, 1.0000, \\ &0.9995, 1.0000, 1.0000) \\ &= 0.9995 \end{aligned}$$

$$\begin{aligned} Im(R1) &= \text{Min}(0.1279, 0.0776, 0.0517, 0.0016, 0.1571, 0.1448, 0.0075, \\ &0.0768, 0.1452) \\ &= 0.0016 \end{aligned}$$

Vậy, đối với luật 1 ta tính được trọng số của bản ghi test với luật 1 như sau: $\omega I = 0.9995e^{0.0016}$

Tương tự, ta tính được trọng số của bản ghi với 6 luật còn lại.

Bước 4: Tổng hợp kết quả đầu ra ở đây ta chọn công thức tính trung bình trọng số để tổng hợp kết quả đầu ra:

$$Y_{op} = \frac{\sum_{i=1}^7 \omega_i Z_i}{\sum_{i=1}^7 \omega_i}$$

Kết quả: $y_{op} = 2.0796$ và nhãn đầu ra được gán cho bản ghi mới là 2

- **Ví dụ 2** Cho bộ luật gồm có 6 luật mờ với 2 nhãn $k=1$ và $k=2$ như sau:

R_1 : If x_1 is Medium1 and x_2 is Medium2 and x_3 is High3 then k is 1

R_2 : If x_1 is Low1 and x_2 is High2 and x_3 is Medium3 then k is 1

R_3 : If x_1 is Low1 and x_2 is Medium2 and x_3 is High3 then k is 1

R_4 : If x_1 is Medium1 and x_2 is Low2 and x_3 is Low3 then k is 2

R_5 : If x_1 is High1 and x_2 is Low2 and x_3 is Medium3 then k is

2

R_6 : If x_1 is High1 and x_2 is Low2 and x_3 is Low3 then k is 2

Bước 1. Xây dựng Đồ thị tri thức mờ

Xem xét với luật R_1 ta có thể tính được giá trị A_{ij}^1 bằng công thức (2.9) được các giá trị như sau:

$$A_{12}^1 = \frac{|\text{Medium1} \rightarrow \text{High2}|}{|R|} = 1/6$$

$$A_{13}^1 = \frac{|\text{Medium1} \rightarrow \text{High3}|}{|R|} = 1/6$$

$$A_{23}^1 = \frac{|\text{High2} \rightarrow \text{High3}|}{|R|} = 1/6$$

Sau đó, tính toán được giá trị đối với nhân đầu ra theo công thức (2.10) như

$$\text{sau:} = (A_{12}^1 + A_{13}^1 + A_{23}^1) * \frac{|\text{Medium1} \rightarrow 1|}{|R|} = \left(\frac{1}{6} + \frac{1}{6} + \frac{1}{6}\right) * \frac{1}{6} = \frac{1}{12}$$

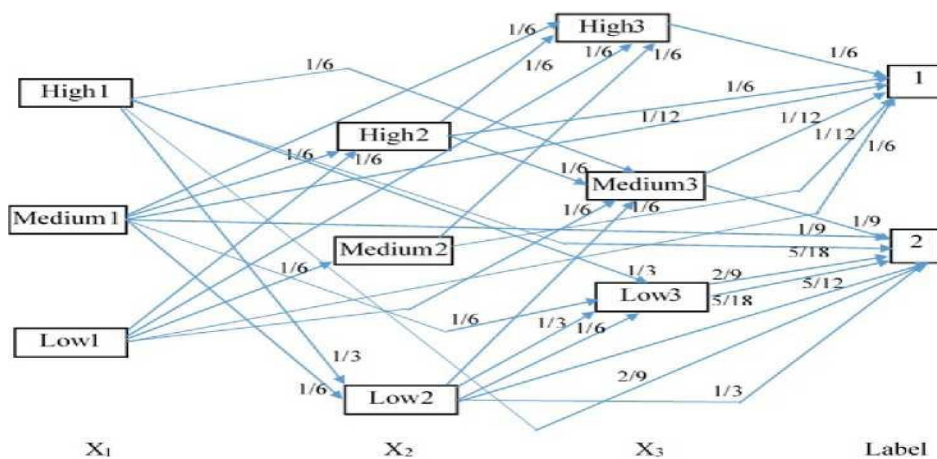
Tương tự ta tính được giá trị $B_{21}^1 = \frac{1}{6}$ và $B_{31}^1 = \frac{1}{6}$

Kết quả thu được ma trận kề chứa các giá trị tính toán A_{ij}^1, B_{ij}^1 từ 6 luật như bảng 2.4

Bảng 2. 4: Bảng ma trận xây dựng đồ thị từ luật

Rule	A_{12}^t	A_{12}^t	A_{12}^t	B_{11}^t	B_{21}^t	B_{31}^t
1	1/6	1/6	1/6	1/12	1/6	1/6
2	1/3	1/6	1/3	5/18	5/12	5/18
3	1/6	1/6	1/6	1/6	1/12	1/6
4	1/6	1/6	1/6	1/6	1/6	1/12
5	1/3	1/6	1/6	2/9	1/3	1/9
6	1/6	1/6	1/3	1/9	1/3	2/9

Đồ thị FKG cho 6 luật được mô tả trong Hình 2.6 sau đây:



Hình 2. 6: Đồ thị FKG cho 6 luật

Bước 2. Áp dụng thuật toán FISA đối với quá trình Testing

Trong quá trình Testing, mục đích cần phải gán được nhãn cho mỗi bản ghi bằng sử dụng thuật toán FISA trên đồ thị FKG đã thu được. ở ví dụ này minh họa việc gán nhãn đối với bộ luật mờ sinh ra sau khi mờ hoá bộ dữ liệu Testing đầu vào.

1) Trường hợp 1: Giả sử gán nhãn cho 6 luật đầu vào mà có điều kiện tương tự như với 6 luật mờ ban đầu. Điều đó có nghĩa các luật mới đều phù hợp với bộ luật cơ sở mờ ban đầu. Bộ 6 luật đầu vào được đặt tên tương ứng là Rule1, ... , Rule 6.

Dựa theo công thức (2.12) ta thu được C_{il} đối với từng nhãn đầu ra l (1 or 2).

Xem xét luật R_1 , với nhãn đầu ra là $l = 1$ ta thu được:

$$C_{11} = \sum_t B_{11}^t = B_{11}^t = 1/12$$

$$C_{21} = \sum_t B_{21}^t = B_{21}(\text{rule R1}) + B_{21}(\text{rule R4}) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$$

$$C_{31} = \sum_t B_{31}^t = B_{31}(\text{rule R1}) + B_{31}(\text{rule R3}) = \frac{1}{6} + \frac{1}{6} = \frac{1}{3}$$

Với nhãn đầu ra $l = 2$, tính toán theo cách tương tự ta thu được $C_{12} = 1/9$,

$C_{22} = 0$, $C_{32} = 0$. Cuối cùng ta thu được kết quả các giá trị của C_{il} tương ứng với mỗi nhãn đầu ra như bảng 2.5, 2.6

Với nhãn 1:

Bảng 2. 5: Bảng tính giá trị C với nhãn 1

Rule	C₁₁	C₂₁	C₃₁
1	1/12	1/3	1/3
2	0	0	0
3	1/3	1/12	1/3
4	1/3	1/3	1/12
5	0	0	1/12
6	1/12	0	0

Bảng 2. 6: Bảng tính giá trị C với nhãn 2

Rule	C₁₂	C₂₂	C₃₂
1	1/9	0	0
2	1/2	13/12	1/2

3	0	0	0
4	0	0	1/9
5	1/2	13/12	1/9
6	1/9	13/12	1/2

Áp dụng toán tử trong công thức 2.13 để tính toán giá trị D_l đối với từng nhãn đầu ra ($l = 1, l = 2$), cụ thể như sau:

Với luật 1:

$$D_l = \min(C_{11}, C_{21}, C_{31}) + \max(C_{11}, C_{21}, C_{31}) = 1/2 + 1/3 = 5/6$$

$$D_l = \min(C_{12}, C_{22}, C_{32}) + \max(C_{12}, C_{22}, C_{32}) = 1/9$$

Do $D_1 > D_2$ nên theo (4.5), nhãn đầu ra của luật 1 là 1.

Tương tự, ta thu được nhãn đầu ra của các luật 2, luật 3, luật 4, luật 5, luật 6 tương ứng là 2, 1, 1, 2, 2. Theo như kết quả cho thấy thì nhãn đầu ra của

6 luật đầu vào khi được thực hiện thông qua thuật toán FISA có kết quả tương tự như với 6 luật cơ sở ban đầu.

2) Trường hợp 2: Trong trường hợp này, thuật toán FISA được sử dụng để

gán các nhãn đầu ra đối với từng mẫu bản ghi đầu vào với các bản ghi có thể không có sẵn trong bộ 6 luật cơ sở ban đầu. Cụ thể như sau:

i) Giả sử có luật 1 như sau:

If x_1 is Medium1 and x_2 is High2 and x_3 is Medium3 then Label =?

Dựa vào đồ thị tri thức của các luật chỉ rõ trong hình 2.3 có thể chỉ ra rằng

không có cạnh nào nối từ nhãn ngôn ngữ Medium1 tới High2 và nối từ nhãn High2 tới Medium3. Tuy nhiên từng cạnh riêng tới nhãn vẫn có tồn tại trong đồ thị. Bằng việc sử dụng mô hình suy diễn trong Suy luận xấp xỉ trong thuật toán FISA thì ta có thể gán nhãn cho từng luật theo cách sau.

Với nhãn 1: ta có $C_{Medium1,1} = 1/12$, $C_{High2,1} = 1/3 \cdot C_{Medium3,1} = 1/12$

$$\begin{aligned} D_1 &= \min(C_{Medium1,1}, C_{High2,1}, C_{Medium3,1}) + \max(C_{Medium1,1}, C_{High2,1}, C_{Medium3,1}) \\ &= 1/12 + 1/3 = 5/12 \end{aligned}$$

Với nhãn 2 ta có: $C_{Medium1,2} = 1/9$, $C_{High2,2} = 0$, $C_{Medium3,2} = 1/9$

$$\begin{aligned} D_2 &= \min(C_{Medium1,2}, C_{High2,2}, C_{Medium3,2}) + \max(C_{Medium1,2}, C_{High2,2}, C_{Medium3,2}) \\ &= 0 + 1/9 = 1/9 \end{aligned}$$

Do $D_1 > D_2$ nên suy ra nhãn của luật là 1.

ii) Xét luật 2 như sau:

If x_1 is High1 and x_2 is Medium2 and x_3 is Low3 then Label =?

Với luật đang xét và trên đồ thị tri thức FKG diễn tả các luật cơ sở thì cũng không có cạnh từ nhãn Medium2 tới Low3. Do đó, ta cần sử dụng thuật toán suy luận xấp xỉ trong FISA để gán nhãn cho luật mới này. Tính toán tương tự như luật trên ta thu được $D_1 = 1/12$, $D_2 = 19/18$. Vì vậy mà nhãn đầu ra của luật mới này được gán bằng 2.

iii) Xét trường hợp luật cuối cùng với trường hợp giả sử không có cạnh nào trong đồ thị tri thức biểu diễn luật đó:

If x_1 is Medium1 and x_2 is Medium2 and x_3 is Medium3 then Label =?

Trên đồ thị FKG, không có cạnh nào đi từ nhãn Medium1 tới Medium2 và cũng không có cạnh nào nối từ nhãn Medium2 tới Medium3. Do đó phương

pháp suy luận xấp xỉ trong thuật toán FISA được dùng để gán nhãn đầu ra đối với trường hợp này. Tính toán tương tự như trên thì thu được $D_1 = 1/6$, $D_2 = 1/4$. Do đó, nhãn đầu ra được gán là 2.

Với ví dụ về số các trường hợp có thể xảy ra đối với dữ liệu đầu vào đã minh họa rõ về khả năng suy diễn đầu ra dựa trên đồ thị tri thức FKG và mô hình suy luận xấp xỉ trong thuật toán FISA. Nó giúp cho chúng ta có thể suy luận một cách linh động, mềm dẻo hơn đối với những trường hợp thiếu tri thức

hay những hệ thống mà có quá ít luật biểu diễn.

2.7. Tổng kết chương 2

Trong chương này, em đã trình bày mô hình suy diễn mờ dựa trên đồ thị tri thức mờ. Từ dữ liệu huấn luyện sử dụng suy diễn mờ xây dựng hệ luật mờ, sau đó xây dựng đồ thị tri thức mờ từ hệ luật mờ. Việc suy diễn để phân lớp dữ liệu kiểm thử dựa trên đồ thị tri thức mờ xây dựng. Cuối cùng đánh giá quá trình thực nghiệm của mô hình xây dựng dựa trên các độ đo đánh giá của phân lớp dữ liệu.

CHƯƠNG 3. CÀI ĐẶT VÀ XÂY DỰNG ỨNG DỤNG

3.1. Bài toán ứng dụng

Trong phần cài đặt thực nghiệm của luận văn gồm 2 phần: phần thử nghiệm trên bộ dữ liệu UCI, phần thử nghiệm trên bộ dữ liệu Y học cổ truyền.

Với bộ dữ liệu trên UCI, từ tập dữ liệu em chia thành 2 phần: 1 phần dữ liệu để huấn luyện xây dựng đồ thị tri thức mờ, 1 phần dữ liệu để kiểm tra suy diễn trên đồ thị tri thức mờ xây dựng. Trong luận văn này em sử dụng 4 bộ dữ liệu trên UCI để thực hiện việc cài đặt và thực nghiệm: Breast, Liver, Diabetes, Iris.

Với bộ dữ liệu Y học cổ truyền: Từ tập dữ liệu về các chỉ số xét nghiệm của bệnh nhân trong quá trình thăm khám tại cơ sở Y tế, các bác sỹ tiến hành kiểm tra và đưa ra các kết luận về tình trạng bệnh dựa trên các triệu chứng. Đây là các dữ liệu để xây dựng đồ thị tri thức mờ. Dựa trên đồ thị tri thức mờ, khi 1 bệnh nhân đến khám với các chỉ số xét nghiệm, hệ thống trợ giúp bác sỹ chẩn đoán xem bệnh nhân bị bệnh gì?

3.2. Mô tả dữ liệu

Dữ liệu thực nghiệm được lấy từ kho dữ liệu UCI và bộ dữ liệu thực, bao gồm các bộ dữ liệu sau:

- Bộ dữ liệu ung thư vú Breast Wisconsin Dataset (WBCD): Bộ dữ liệu ung thư vú được lấy từ Bệnh viện đại học Wisconsin, Madison, Mỹ. Bộ dữ liệu bao gồm kết quả của 699 bệnh nhân (trong đó 458 bệnh nhân được chẩn đoán là u lành tính và 241 bệnh nhân được chẩn đoán là ung thư).

Bảng 3. 1: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh ung thư Breast

Số thứ tự	Thuộc tính	Mô tả
1	Tuổi (năm)	Tuổi tính đến ngày xét nghiệm
2	BMI (kg / m ²)	Chỉ số BMI
3	Glucose (mg / dL)	Chỉ số Glucose
4	Insulin (μ U / mL)	Chỉ số Insulin
5	HOMA	Chỉ số HOMA
6	Leptin (ng / mL)	Chỉ số Leptin
7	Adiponectin (μ g / mL)	Chỉ số Adiponectin
8	(ng / mL)	Chỉ số Resistin
9	(pg / dL)	Chỉ số MCP-1

- Bệnh tiểu đường Diabetes: Dữ liệu bệnh tiểu đường được lấy từ kho dữ liệu bệnh của Đại học Y Virginia. Dữ liệu được thu thập từ 391 bệnh nhân người Mỹ gốc Phi ở miền trung Virginia để tìm hiểu dự báo về bệnh tiểu đường.

Bảng 3. 2: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh tiểu đường Diabetes

Số thứ tự	Thuộc tính	Mô tả
1	Ngày	MM-DD-YYYY
2	Giờ	XX:YY
3	Mã	Các thông tin được mã hóa
4	Giá trị	Giá trị tương ứng

- Bộ dữ liệu cây Diên Vĩ Iris: Dữ liệu được lấy từ bài báo kinh điển của RA.Fisher. Đây có lẽ là cơ sở dữ liệu nổi tiếng nhất được tìm thấy trong tài liệu nhận dạng mẫu. Tập dữ liệu chứa 3 lớp, mỗi lớp 50 trường hợp, trong đó mỗi lớp đề cập đến một loại cây diên vĩ. Một lớp có thể phân tách tuyến tính với 2 lớp còn lại; cái sau không thể tách rời tuyến tính với nhau.

Bảng 3. 3: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu cây Diên Vĩ Iris

Số thứ tự	Thuộc tính	Mô tả
1	Chiều dài đài hoa	Tính bằng cm
2	Chiều rộng đài hoa	Tính bằng cm
3	Chiều dài cánh hoa	Tính bằng cm
4	Chiều rộng cánh hoa	Tính bằng cm

- Bộ dữ liệu bệnh gan Liver: Đây là những kết quả xét nghiệm (sinh hóa máu và công thức máu) và chẩn đoán bệnh được trích từ hồ sơ bệnh án của các bệnh nhân tại Bệnh viện Gang Thép và Bệnh viện Đa khoa Thái Nguyên. Tập dữ liệu Liver gồm 4156 bệnh nhân đến khám và điều trị bệnh do rối loạn men gan. Trong số 4156 hồ sơ bệnh nhân, có 1202 hồ sơ bệnh nhân được chẩn đoán là mắc bệnh viêm gan. Cụ thể như trong bảng sau:

Bảng 3. 4: Các thuộc tính dữ liệu đầu vào trong tập dữ liệu bệnh gan Liver

Số thứ tự	Thuộc tính	Mô tả
1	Tuổi (Age)	Tuổi tính đến ngày xét nghiệm
2	Giới tính (Gender)	0: nam; 1: nữ
3	3 AST	Chỉ số men AST
4	ALT	Chỉ số men ALT
5	GGT	Chỉ số GGT
6	Albumin	Chỉ số Albumin
7	TB	Chỉ số Total Bilirubin
8	DB	Chỉ số Direct Bilirubin
9	DB/TB	Tỷ số DB/TB

Trong thực nghiệm, mỗi bộ dữ liệu sẽ được chia làm 4 nhóm bao gồm bộ dữ liệu training, bộ dữ liệu Testing, bộ dữ liệu Validation và bộ dữ liệu New data. Bộ dữ liệu New data là bộ dữ liệu bao gồm các bản ghi mới được sử dụng với mục đích kiểm tra mô hình đề xuất trong trường hợp không có bản ghi

nào
xuất hiện trong phần Training (dùng để đo độ tương thích và khả năng suy
luận
của mô hình đề xuất).

Bộ dữ liệu Y học cổ truyền tại Bệnh viện Đa khoa Hưng Yên: dữ liệu thu thập dựa trên các bệnh nhân mang thai với bệnh tiền sản giật. Trong quá trình thu thập được 199 mẫu trong thời kỳ thai nhi với 19 thuộc tính có ảnh hưởng đến bệnh tiền sản giật. Trong đó có 114 bệnh nhân có thai phụ bình thường, 57 bệnh nhân có dấu hiệu tiền sản giật và 28 bệnh nhân có dấu hiệu tiền sản giật nặng. Các thông tin cụ thể của dữ liệu được minh họa ở bảng 3.5.

Bảng 3. 5: Dữ liệu Y học cổ truyền

STT	TUỔI	NGHỀ NGHIỆP	TUỔI THAI (tuần)	SỐ LẦN CÓ THAI	PARA	CHIỀU CAO	CÂN NẶNG	HA TRÊN (90-129)	HA DƯỚI (60-84)	HGB (120-160)	PLT (150-450)	URE (2.5-6.7)	CREATININ (50.4-98.1)	ACID URIC (150-350)	ALT (<31/37 oC)	AST (<31/37 oC)	PROTEIN TOÁN PHẢN (64-83)	ALBUMIN (35-52)	LDH < 247	Protein niệu (g/l) 0.1-0.25	Kết luận
1	37	Công nhân	24	3	2002	157	60	140	90	116	113	4.97	61.45	367	160	44.8	55.6	27.4	274.2	3	TSG NẶNG
2	21	Nông dân	34.4	1	0000	152	46	140	100	122	65	6.4	67.71	458	65.09	84.9	57.75	26.41	577.6	9.9	TSG
3	32	Tự do	31	2	1011	160	62	210	120	129	141	4.39	67.36	449	11.03	44.09	55.84	29.24	288.42	6.4	TSG NẶNG
4	30	Dược sĩ	32	1	0000	155	55	160	100	144	229	5.54	72.96	463	44.2	42.8	73.3	38.5	255.9	0.25	TSG
5	33	Tự do	33	1	0000	153	63	160	100	126	196	5.65	66.62	447	29.6	30.8	59.7	30.1	277	9.4	TSG NẶNG
6	26	Nhân viên	33	1	0010	168	65	140	90	131	190	6.67	64.46	551	29.65	28.07	59.11	31.79	155.2	1.2	TSG
7	26	Nhân viên	34	1	0000	158	61	160	100	121	190	3.14	50.6	300	11.7	15.5	65.51	35.63	160.6	0.5	TSG
8	23	Tự do	30	2	1001	160	60	130	90	109	178	4.09	57.31	355	10.7	20	63.9	27.4	171.1	0.9	TSG
9	38	Nông dân	32	2	1040	160	63	140	100	120	203	5.26	70.5	398	21.6	29.9	61.7	28.1	201	3.3	TSG
10	29	Nông dân	33	2	1001	153	57	150	100	140	42	5.97	68.57	455	18.62	38.31	61.42	26.88	317.3	4.8	TSG nặng
11	29	Công nhân	32	2	10001	153	52	150	110	79	247	9.37	127.51	477	13.87	55.73	64.74	30.36	339.51	0.97	TSG
12	33	Giáo viên	21	2	1001	156	59	150	100	103	234	11.26	132.0	529	14.4	18.6	54.46	25.26	276.3	0.3	TSG
13	34	Công nhân	33	3	2002	160	65	210	140	136	118	3.62	65.00	399	58.91	52.71	63.73	27.94	220.85	0.15	TSG

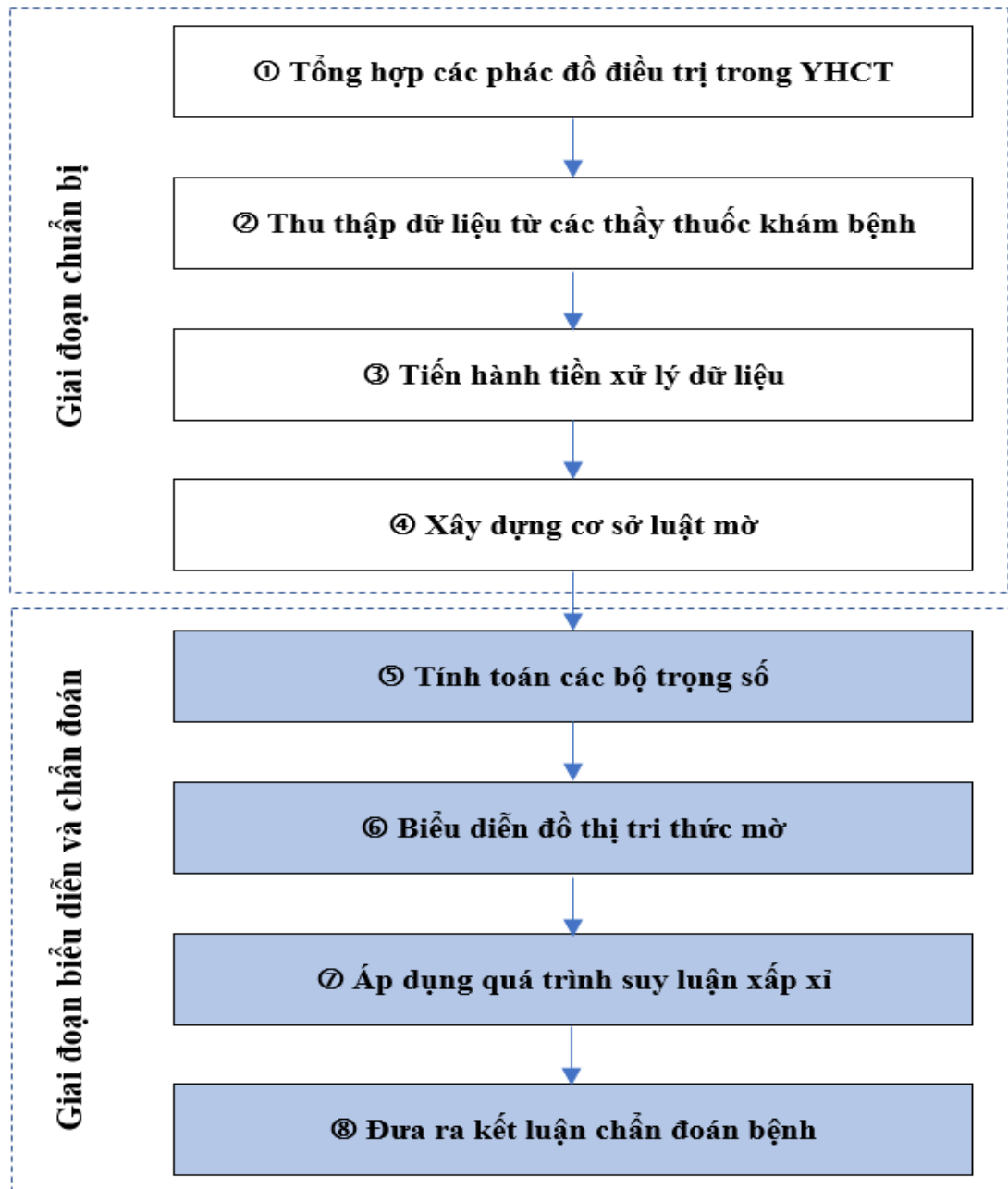
3.3. Cài đặt và xây dựng ứng dụng

Quá trình thực nghiệm sử dụng ngôn ngữ MATLAB 2014, cài đặt trên máy tính có thông tin cấu hình: HP Intel(R) Core (TM) i5-7200U, CPU @ 2.50GHz 2.70 GHz.

Mô hình cho bài toán ứng dụng: Bài toán hỗ trợ chẩn đoán bệnh nhân đang mang thai trong y học cổ truyền với bệnh tiền sản giật gồm các bước được chia thành hai giai đoạn (hình 3.1). Ở mô hình này được chia làm 2 giai đoạn: Giai đoạn chuẩn bị dữ liệu và xây dựng hệ luật mờ; Giai đoạn biểu diễn và hỗ trợ chẩn đoán.

Bài toán hỗ trợ chẩn đoán bệnh trong y học cổ truyền thực hiện như sau:

Dựa trên các phác đồ điều trị của Y học cổ truyền thực hiện quá trình tiền xử lý dữ liệu và dựa trên tập dữ liệu thu thập tại bệnh viện từ các bác sĩ. Xây dựng hệ luật mờ, từ hệ luật mờ xây dựng đồ thị tri thức mờ. Khi một bệnh nhân đến khám dựa trên 19 thông tin về các thuộc tính, dựa trên đồ thị tri thức mờ hệ thống tự động hỗ trợ các bác sĩ trong quá trình chẩn đoán bệnh.



Hình 3. 1: Mô hình cho bài toán hỗ trợ chẩn đoán bệnh trong y học cổ truyền

Giai đoạn chuẩn bị dữ liệu gồm 4 bước:

- Bước 1: Tổng hợp từ phác đồ điều trị trong Y học cổ truyền để xây dựng các thông tin thu thập dữ liệu.

- Bước 2: Thu thập dữ liệu từ các cơ sở y tế: cư trú vào các bệnh nhân đang mang thai từ 1-5 tuần.

- Bước 3: Tiền xử lý dữ liệu: dữ liệu thu thập loại bỏ những thuộc tính thông tin của bệnh nhân và giữ lại những thuộc tính xét nghiệm liên quan đến bệnh tiền sản giật.

- Bước 4: Xây dựng cơ sở luật mờ: sử dụng suy diễn mờ xây dựng luật mờ từ dữ liệu sau khi tiền xử lý.

Giai đoạn biểu diễn và chẩn đoán gồm 4 bước:

- Bước 5: Tính toán các trọng số: thực hiện tính toán các trọng số xây dựng đồ thị.

- Bước 6: Biểu diễn đồ thị tri thức mờ: sử dụng ma trận B để biểu diễn đồ thị tri thức mờ được trình bày ở chương 2.

- Bước 7: Áp dụng quá trình suy luận xấp xỉ: sử dụng suy diễn trên đồ thị tri thức mờ được trình bày ở chương 2.

- Bước 8: Đưa ra kết quả hỗ trợ chẩn đoán: đánh giá kết quả thu được để đưa ra kết quả hỗ trợ các bác sĩ khi chẩn đoán bệnh Tiền sản giật.

3.4. Kết quả thực nghiệm

3.4.1. Kết quả thực nghiệm trên bộ dữ liệu lấy từ UCI

Các kết quả thực nghiệm được đánh giá trên độ chính xác accuracy và thời gian thực thi mô hình. Mô hình đồ thị tri thức mờ (FKG) so sánh với mô hình suy diễn mờ (FIS).

Quá trình thực nghiệm chia ngẫu nhiên dữ liệu 70% dùng để training và 30% dùng để testing. Kết quả lấy là trung bình cộng của 20 lần thực nghiệm.

Kết quả về độ chính xác (accuracy) trên các bộ dữ liệu thì mô hình FKG cho kết quả tốt hơn so với mô hình FIS.

Bảng 3. 6: Kết quả thực nghiệm Accuracy trên bộ dữ liệu UCI

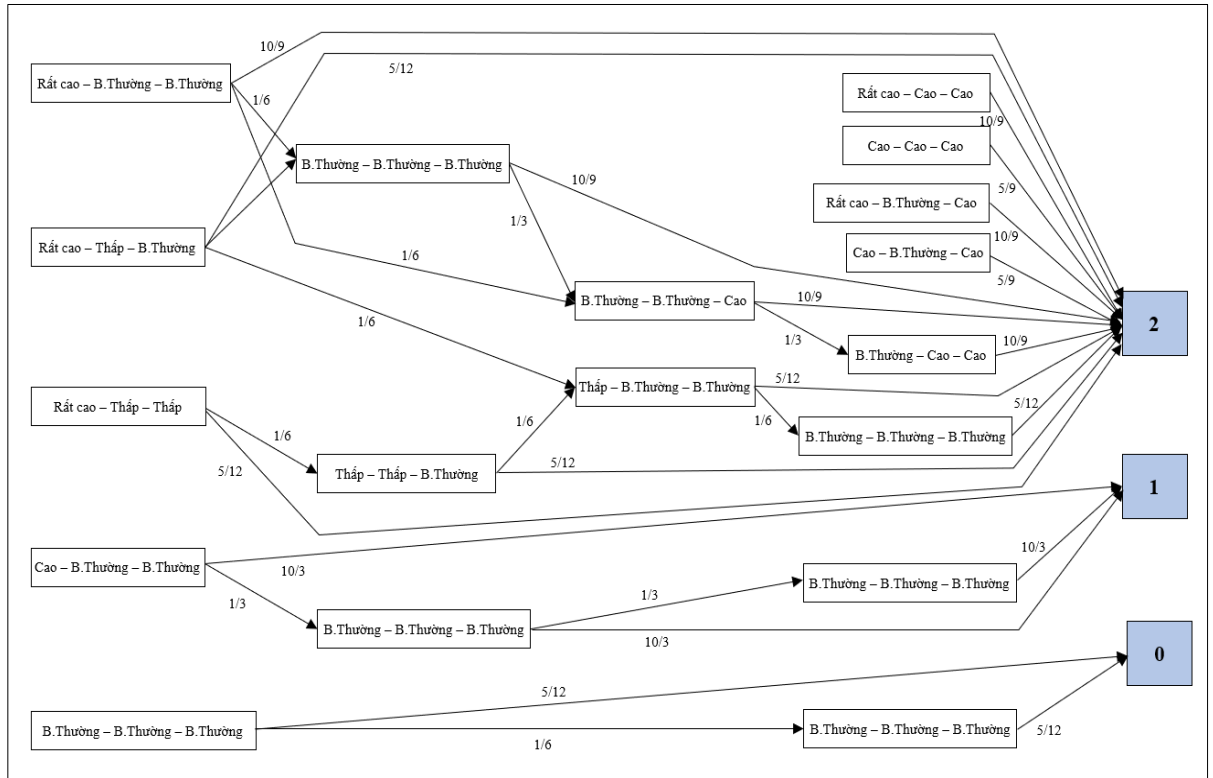
Data	FIS (%)	FKG (%)
Breast	74.63	85.78
Liver	80.73	85.73
Diabetes	78.78	85.62
Iris	82.94	89.78

Kết quả thực nghiệm về thời gian được xác định trong bảng 3.7. Mô hình đồ thị tri thức cho thời gian thực thi của mô hình FKG giảm đi rất nhiều so với mô hình FIS.

Bảng 3. 7: Kết quả thực nghiệm thời gian trên bộ dữ liệu UCI

Data	FIS (giây)	FKG (giây)
Breast	1.92	0.78
Liver	5.02	2.37
Diabetes	2.36	1.09
Iris	1.87	0.67

3.4.2. Kết quả thực nghiệm trên bộ dữ liệu Y học cổ truyền



Hình 3. 2: Đồ thị tri thức mờ với bệnh án Y học cổ truyền

Từ các bệnh án thu thập của 546 bệnh nhân thu được đồ thị tri thức mờ (biểu diễn minh họa hình 3.2).

Độ đo về độ chính xác (*accuracy*) và thời gian tính toán (*computational time*) là các tiêu chí được dùng để đánh giá so sánh hiệu năng của các phương pháp FKG với mô hình suy diễn mờ (FIS) với các cách chia dữ liệu khác nhau. Cụ thể như sau:

Case 1: 70% dữ liệu train và 30% dữ liệu test

Case 2: 60% dữ liệu train và 40% dữ liệu test

Case 3: 50% dữ liệu train và 50% dữ liệu test

Kết quả thực nghiệm về *accuracy* được xác định trong bảng 3.8. Mô hình đồ thị tri thức cho kết quả tốt hơn mô hình suy diễn mờ trong cả 3 trường hợp phân chia dữ liệu. Tuy nhiên khi dữ liệu thay đổi kết quả của FKG ổn định không bị giảm nhiều so với mô hình FIS.

Bảng 3. 8: *Kết quả thực nghiệm Accuracy trên bộ dữ liệu tiền sản giật*

Case	FIS (%)	FKG (%)
Case 1	79.53	84.72
Case 2	81.73	83.26
Case 3	74.37	83.08

Kết quả thực nghiệm về thời gian được xác định trong bảng 3.9. Mô hình đồ thị tri thức cho thời gian thực thi của mô hình FKG giảm đi rất nhiều so với mô hình FIS.

Bảng 3. 9: *Kết quả thực nghiệm thời gian trên bộ dữ liệu tiền sản giật*

Case	FIS (giây)	FKG (giây)
Case 1	4.763	1.273
Case 2	3.653	1.637
Case 3	4.273	1.928

Với kết quả thực nghiệm dựa trên 2 thang đo là độ chính xác và thời gian mô hình đồ thị tri thức mờ (FKG) cho kết quả tốt hơn so với mô hình suy diễn mờ (FIS) trên cơ sở bộ dữ liệu của UCI và bộ dữ liệu về Y học cổ truyền với bệnh tiền sản giật thu thập tại bệnh viện đa khoa Hưng Yên.

3.5. Tổng kết chương 3.

Trong chương này, em đã trình bày về bài toán em sử dụng trong luận văn. Cài đặt và thực nghiệm mô hình. Trong quá trình thực nghiệm em sử

dụng 2 loại dữ liệu: 1 loại dữ liệu lấy từ UCI và 1 loại dữ liệu thu từ cơ sở Y tế. Với dữ liệu lấy từ UCI, sử dụng trên 4 bộ dữ liệu: Breast, Liver, Diabetes, Iris.

KẾT LUẬN

Luận văn nghiên cứu về đồ thị tri thức mờ và áp dụng vào các bài toán phân lớp dữ liệu. Luận văn đã nêu ra và giải quyết được một số vấn đề như sau:

- Tìm hiểu được những kiến thức tổng quan về Logic mờ, lý thuyết đồ thị, các hệ suy diễn mờ... làm nền tảng để phát triển cơ sở lý thuyết cho nghiên cứu.

- Nghiên cứu cách thức xây dựng luật mờ từ dữ liệu.

- Nghiên cứu cách xây dựng và suy diễn trên đồ thị tri thức mờ.

- Cài đặt và thực thi đồ thị tri thức mờ với 2 loại dữ liệu là: dữ liệu chuẩn và dữ liệu thu thập.

Tuy nhiên trong quá trình nghiên cứu luận văn, em thấy còn có một số vấn đề em chưa thực hiện được:

- Dữ liệu tiền sản giật thu thập chưa thực sự nhiều.

- Chưa phát triển ứng dụng để đưa vào sử dụng.

- Cần cải tiến hơn nữa để nâng cao hiệu suất.

Đồ thị tri thức mờ trong quá trình em nghiên cứu trong luận văn, em thấy đây là một vấn đề hay và có một số hướng phát triển tiếp theo như sau:

- Biểu diễn đồ thị tri thức mờ với nhiều thuộc tính kết hợp.

- Phát triển đồ thị tri thức mờ dựa trên các tập mờ nâng cao.

- Xây dựng và thử nghiệm với các ứng dụng dữ liệu ít và được bổ sung tăng cường khi triển khai.

PHỤ LỤC

Code suy diễn mờ:

```

function [centers centers_var t t_var] = MakeRules(train_data)
data_num attribute_num] = size(train_data);
cluster = zeros(1, attribute_num);
cluster(1) = 2;
for i=2:attribute_num
    temp = length(unique(train_data(:,i)));
    if temp == 2
        cluster(i) = 2;
    elseif temp == 3
        cluster(i) = 3;
    else
        cluster(i) = 5;
    end
end

m = 2;
esp = 0.01;
maxTest = 200;
center_vector = cell (1, attribute_num);
centers = cell(attribute_num-1, 1);
for feature_index=1:attribute_num
    feature_data = train_data(:,feature_index);
    V = 0;

```



```

V_var = 0;
min_value = min(train_data(:,feature_index));
max_value = max(train_data(:,feature_index));
delta = max_value - min_value;
if cluster(feature_index) == 2
    V(1,1) = min_value - 0.5;
    V(2,1) = max_value;
elseif cluster(feature_index) == 3
    V(1,1) = min_value;
    V(2,1) = min_value + delta/2;
    V(3,1) = max_value;
else
    V(1,1) = min_value;
    V(2,1) = min_value + delta/4;
    V(3,1) = min_value + 2*delta/4;
    V(4,1) = min_value + 3*delta/4;
    V(5,1) = max_value;
end
[center,U] =
FCM_Function(feature_data,cluster(feature_index),V,m,esp,maxTest);
U = U';
center_vector{feature_index}(:,1) = center(:,1);
for i=1:data_num
    maximum = max(U(i,:));
    for j=1:cluster(feature_index)

```

```

        if (maximum == U(i,j))
            rules(i,feature_index) = j;
        end
    end
end

if feature_index ~= 1
    center = center';
    centers{feature_index-1} = center(1,:);
end

end

[t,sigma_M] = RuleWeight(rules, train_data,cluster,center_vector);
sigma_M(1,:) = [];
for i=1:(attribute_num-1)
    sigma_M(i,2:5) = sigma_M(i,1);
end

sigma_M_var(1,:) = [];
beta = zeros(data_num,attribute_num);
for i=1:data_num
    beta(i,:) = [[1 train_data(i,2:attribute_num)]\train_data(i,1)'];
end

label = train_data(:,1);
for i=1:data_num
    rules(i,(attribute_num+1)) = min(t(i,2:attribute_num));
end

```

```

for i=1:data_num-1
    for j=i+1:data_num
        if(rules(i,2:attribute_num) == rules(j,2:attribute_num))
            if(rules(i,(attribute_num+1)) > rules(j,(attribute_num+1)))
                rules(j,:)=0;
            else
                rules(i,:)=0;
            end
        end
    end
end
end

```

end

Rules with weight < 0.5 will be removed

```

for i=1:data_num
    if(rules(i,(attribute_num+1)) < 0.5)
        rules(i,:) = 0;
    end
end

```

end

```
rules(:,1) = [];
```

Filter rules

```
RuleCheck = zeros(1,attribute_num);
```

```
j = 1;
```

```

for i=1:data_num
    if (rules(i,:) ~= RuleCheck(1,:))
        FilteredRules(j,:) = [rules(i,1:(attribute_num-1)) label(i)];
        FilterT(j,:)=t(i,:);
    end
end

```

```

    j = j + 1;
end
end
W_rule=min(FilterT');
ruleList = FilteredRules;
Code xây dựng đồ thị tri thức:
for i =1:r1
    j=1;
    for l=1:r2-2
        for h=l+1:r2-1
            dem=0;
            for t=1:r1
                if(ruleList1(t,l)==ruleList1(i,l))
                    &(ruleList1(t,h)==ruleList1(i,h))
                        dem=dem+1;
                    end
                end
            end
            A(i,j)=dem/r1;
            j=j+1;
        end
    end
end
[k1 k2]= size(A);
for i=1:2
    for j=1:k2

```

```

tong=0;
for l=1:k1
    if ruleList1(l,r2)==i-1
        tong=tong+A(l,j);
    end
end
T(i,j)=tong;
end
end
for i=1: r1
    dem=1;
    for j=1:r2-2
        for k=j+1:r2-1
            dem1=0;
            dem2=0;
            for l=1:r1
                if (ruleList1(l,j)==ruleList1(i,j))&(ruleList1(l,r2)==ruleList1(i,r2))
                    dem1=dem1+1;
                end
                if (ruleList1(l,k)==ruleList1(i,k))&(ruleList1(l,r2)==ruleList1(i,r2))
                    dem2=dem2+1;
                end
            end
        end
    end
    M1=dem1/r1;

```

```

M2=dem2/r1;
for l1=1:2
    if ruleList1(i,r2)==l1
        B(i,dem)=T(l1,dem)*min(M1,M2);
    end
end
dem=dem+1;
end
end
end

```

Code suy diễn trên đồ thị tri thức mờ:

```

for i=1: a1
    dem=1;
    for j=1:r2-2
        for k=j+1:r2-1
            dem1=0;
            dem2=0;
            for l=1:r1
                if (te(i,j)==ruleList1(l,j))&(te(i,k)==ruleList1(l,k))
                    if (ruleList1(k,r2)==0)
                        dem1=dem1+B(l,dem);
                    else
                        dem2=dem2+B(l,dem);
                    end
                end
            end
        end
    end
end

```

```

    end
    C1(i,dem)=dem1;
    C2(i,dem)=dem2;
    dem=dem+1;
end
end
end
minC1=min(C1');
maxC1=max(C1');
D1=minC1+maxC1;
minC2=min(C2');
maxC2=max(C2');
D2=minC2+maxC2;
for k=1:a1
    if D1(k)>D2(k)
        nhan(k)=0;
    else
        nhan(k)=1;
    end
end
end

```

Độ đo về độ chính xác (*accuracy*) và thời gian tính toán (*computational time*)

Code mô hình FIS để so sánh

```

function [Time, Accuracy, Recall, Precision] = Run_Train_FIS(temp_data,
centers, centers_var,ruleList, ruleList_var)
eta=0.000001;

```

```

term_num=5;
FIS_para_filename = strrep('Train','.txt','.mat');
FIS_para_filename = strrep(FIS_para_filename, 'Database', 'FIS');
addpath('..\output');
load(['..\output\' FIS_para_filename]);
train_output = temp_data(:,1);
temp_data(:,1) = [];
train_input = temp_data;
input_num=size(train_input,1);
attri_num=size(train_input,2);
rule_num=size(ruleList,1);
rule_length=size(ruleList,2);
re_degree_M=ones(input_num,attri_num,term_num);
wtsum = zeros(input_num,1);
M_DataPerRule = zeros(input_num,rule_num);
start_time_train = cputime;
for i=1:input_num
    for j=1:attri_num
        for k=1:length(centers{j})
            degree_M(i,j,k) = gaussmf(train_input(i,j),[sigma_M(j,k)
centers{j}(k)]);
        end
    end
end
min_M=1;

```



```

for i=1:input_num
    for j=1:rule_num
        min_M=1;

        for k=1:attri_num
            degree_M(i,k,ruleList(j,k));
            if min_M>degree_M(i,k,ruleList(j,k));
                min_M=degree_M(i,k,ruleList(j,k));
            end
        end
        M_DataPerRule(i,j)=re_min_M;
    end
end

result=zeros(input_num,1);
for i=1:input_num
    result(i)=0; wtsum(i)=0;
    for j=1:rule_num
        wtsum(i)=wtsum(i)+M_DataPerRule(i,j)*cos(M_DataPerRule(i,j));
        result(i)=result(i)+M_DataPerRule(i,j)*cos(M_DataPerRule(i,j))*ruleList(j,rule_length);
    end
    result(i)=sqrt(result(i)^2 /wtsum(i)^2);
end
temp=result;

```

```
data_num=size(result);
for i=1:data_num
    if temp(i)>=0.41*(max(train_output)+min(train_output))
        temp(i)=max(train_output);
    else
        temp(i)=min(train_output);
    end
end
end_time_train = cputime;
Time = end_time_train - start_time_train;
TN=0;
TP=0;
FN=0;
FP=0;
for i=1:size(temp)
    if (temp(i)==train_output(i))& (temp(i)==0)
        TP=TP+1;
    end
    if (temp(i)==train_output(i))& (temp(i)==1)
        TN=TN+1;
    end
    if (train_output(i)==0)& (temp(i)==1)
        FN=FN+1;
    end
    if (train_output(i)==1)& (temp(i)==0)
```

FP=FP+1;

end

end

Accuracy=(TN+TP)/(TN+TP+FN+FP);

Danh mục tài liệu tham khảo

- [1]. Wenqiang Liu, Jun Liu, Mengmeng Wu, Samar Abbas, Qinghua Zheng, Representation learning over multiple knowledge graphs for knowledge graphs alignment. *Neurocomputing*, Volume 320, 3 December 2018, Pages 12-24
- [2]. Yantao Jia, Yuanzhuo Wang, Xiaolong Jin, Xueqi Cheng, Path-specific knowledge graph embedding, *Knowledge-Based Systems*, Volume 151, 1 July 2018, Pages 37-44
- [3]. Thao, N. X., Cuong, B. C., Ali, M., & Lan, L. H. (2018). Fuzzy equivalence on standard and rough neutrosophic sets and applications to clustering analysis. In *Information Systems Design and Intelligent Applications* (pp. 834-842). Springer, Singapore.
- [4]. Giang, N. L., Son, L., H., Ngan, T. T., Tuan, T. M., Phuong, H. T., Abdel-Basset, M., de Macêdo, A. R. L., & Albuquerque, V. (2019). Novel Incremental Algorithms for Attribute Reduction from Dynamic Decision Tables using Hybrid Filter–Wrapper with Fuzzy Partition Distance. *IEEE Transactions on Fuzzy Systems*, 28(5), pp.858-873.
- [5]. Luong Thi Hong Lan et al., “A New Complex Fuzzy Inference System with Fuzzy Knowledge Graph and Extensions in Decision Making” *IEEE Access*. Vol. 8, 2020 (164899-164921). DOI: 10.1109/ACCESS.2020.3021097.
- [6]. Lien Faelens, Kristof Hoorelbeke, Eiko Fried, Rudi De Raedt, Ernst H. W. Koster, Negative influences of Facebook use through the lens of network analysis, *Computers in Human Behavior*, Volume 96, July 2019, Pages 13-22
- [7]. Wenqiang Liu, Jun Liu, Mengmeng Wu, Samar Abbas, Qinghua Zheng, Representation learning over multiple knowledge graphs for knowledge graphs alignment. *Neurocomputing*, Volume 320, 3 December 2018, Pages 12-24

- [8]. De Matteis, A. D., Marcelloni, F., & Segatori, A. (2015, August). A new approach to fuzzy random forest generation. In *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on* (pp. 1-8). IEEE.
- [9]. James, S. C., Zhang, Y., & O'Donncha, F. (2018). A machine learning framework to forecast wave conditions. *Coastal Engineering, 137*, 1-10.
- [10]. Luong Thi Hong Lan et al., “A New Complex Fuzzy Inference System with Fuzzy Knowledge Graph and Extensions in Decision Making” IEEE Access. Vol. 8, 2020 (164899-164921). DOI: 10.1109/ACCESS.2020.3021097.
- [11]. N. K. Geetha, P. Sekar, An unprecedented multi attribute decision making using graph theory matrix approach, Engineering Science and Technology, an International Journal, Volume 21, Issue 1, February 2018, Pages 7-16.
- [12]. Phuong, N. H., & Kreinovich, V. (2001). Fuzzy logic and its applications in medicine. *International journal of medical informatics, 62(2)*, 165-173.
- [13]. Khang, T. D., Phong, P. A., Dong, D. K., & Trang, C. M. (2010, July). Hedge Algebraic Type-2 Fuzzy Sets. In *International Conference on Fuzzy Systems* (pp. 1-8). IEEE.
- [14]. Son, L.,H., Tuan, T. M. (2016). A cooperative semi-supervised fuzzy clustering framework for dental X-ray image segmentation. *Expert Systems with Applications, 46*, 380-393.
- [15]. Cù, K. L., Trần, M. T., Lê, H. S., Lương, T. H. L., Phạm, M. C., Nguyễn, T. T., & Phạm, V. H. (2021). Chẩn đoán bệnh trong y học cổ truyền: Hướng tiếp cận dựa trên đồ thị tri thức mờ dạng cặp. *Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông*, 59-68.
- [16]. Üncü, Ü. (2010). Evaluation of pulmonary function tests by using fuzzy logic theory. *Journal of medical systems, 34(3)*, 241-250.

[17]. Zaidi, N. L. B., Santen, S. A., Purkiss, J. A., Teener, C. A., & Gay, S. E. (2016). A Hybrid Interview Model for Medical School Interviews: Combining Traditional and Multisampling Formats. *Academic Medicine*, 91(11), 1526-1529.

[18]. Boriss Siliverstovs, Daniel S. Wochner, Google Trends and reality: Do the proportions match?: Appraising the informational value of online search behavior: Evidence from Swiss tourism regions, *Journal of Economic Behavior & Organization*, Volume 145, January 2018, Pages 1-23.