ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC

LÊ VĂN TƯỜNG LÂN

PHÂN LỚP DỮ LIỆU BẰNG CÂY QUYẾT ĐỊNH MỜ DỰA TRÊN ĐẠI SỐ GIA TỬ

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH
MÃ SỐ: 62.48.01.01

LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

Người hướng dẫn khoa học:
1. PGS.TS. Nguyễn Mậu Hân
2. TS. Nguyễn Công Hào

HUẾ - NĂM 2018

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu do tôi thực hiện, dưới sự hướng dẫn khoa học của PGS.TS. Nguyễn Mậu Hân và TS. Nguyễn Công Hào. Các số liệu và kết quả trình bày trong luận án là trung thực, chưa được công bố bởi bất kỳ tác giả nào hay ở bất kỳ công trình nào khác.

LỜI CẢM ƠN

Trong quá trình thực hiện đề tài **"Phân lớp dữ liệu bằng cây quyết định mờ dựa trên đại số gia tử"**, tôi đã nhận được rất nhiều sự giúp đỡ, tạo điều kiện của tập thể Ban giám hiệu, Phòng Đào tạo Sau đại học, Khoa Công nghệ thông tin và các phòng chức năng của Trường Đại học Khoa học, Đại học Huế. Tôi xin bày tỏ lòng cảm ơn chân thành về sự giúp đỡ quý báu đó.

Tôi xin được bày tỏ lòng biết ơn sâu sắc tới PGS.TS. Nguyễn Mậu Hân và TS. Nguyễn Công Hào là những thầy giáo trực tiếp hướng dẫn và chỉ bảo cho tôi hoàn thành luận án.

Tôi xin chân thành cảm ơn gia đình, bạn bè và đồng nghiệp đã động viên, khích lệ, tạo điều kiện và giúp đỡ tôi trong suốt quá trình thực hiện và hoàn thành luận án này.

TÁC GIẢ LUẬN ÁN
Nghiên cứu sinh

Lê Văn Tường Lân

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Viết tắt	Viết đầy đủ
ĐSGT	Đại số gia tử
GĐ1	Giai đoạn 1
GĐ2	Giai đoạn 2
CART	Classification and Regression Trees
Dom	Domain
Gain	Gain Information
GainRatio	Gain Information Ratio
HA	Hedge Algebra
LDT	Linguistic Decision Tree
Sim	Similar
SplitInfo	Split Information

DANH MỤC CÁC KÝ HIỆU

Ký hiệu	Diễn giải ý nghĩa
A_i	Thuộc tính A_i
D	Tập mẫu huấn luyện
D_{A_i}	Tập các giá trị kinh điển của A_i
f	Ánh xạ
$f_h(S)$	Hàm đánh giá tính hiệu quả của cây
$f_n(S)$	Hàm đánh giá tính đơn giản của cây
I_k	Tập tất cả các khoảng mờ mức k của các giá trị ngôn ngữ
LD_{A_i}	Tập các giá trị ngôn ngữ của A_i
$O(log\ n)$	Độ phức tạp *logarit* của thuật toán
$\mu_A(v)$	Hàm định lượng của giá trị ngôn ngữ A (đo độ thuộc của v)
S	Cây quyết định
$sim(x, y)$	Mức độ gần nhau của x và y
v	Giá trị định lượng theo điểm của giá trị ngôn ngữ
$\underline{X}$	Đại số gia tử
Y	Thuộc tính phân lớp

DANH MỤC CÁC BẢNG BIỂU

DANH MỤC CÁC HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

Trong cuộc sống con người, ngôn ngữ được hình thành một cách tự nhiên để đáp ứng nhu cầu trao đổi thông tin của xã hội. Hơn thế, ngôn ngữ là công cụ để con người mô tả các sự vật, hiện tượng trong thế giới thực và dựa trên đó để tư duy, lập luận đưa ra những nhận định, phán quyết nhằm phục vụ cho cuộc sống xã hội của chúng ta. Trong thực tế, các khái niệm mờ luôn tồn tại, ví dụ như *trẻ, rất trẻ, hơi già, quá già*,... nên với việc quan niệm các đối tượng được sử dụng phải luôn rõ ràng ở trong logic cổ điển sẽ không đủ miêu tả các vấn đề của thế giới thực.

Năm 1965, L. A. Zadeh đã đề xuất hình thức hóa toán học của khái niệm mờ [79], từ đó lý thuyết tập mờ được hình thành và ngày càng thu hút nhiều nhà nghiên cứu. Bằng các phương pháp tiếp cận khác nhau, nhiều nhà nghiên cứu như Dubois, Prade [21], Mariana [50], Ishibuchi [36], Herrera [8], Yakun Hu [77],… đã đưa ra những kết quả cả về lý thuyết và ứng dụng cho nhiều lĩnh vực như: điều khiển mờ, cơ sở dữ liệu mờ, khai phá dữ liệu mờ. Ý tưởng nổi bật của Zadeh là từ những khái niệm trừu tượng về ngữ nghĩa của thông tin mờ, không chắc chắn như *trẻ-già, nhanh-chậm, cao-thấp*,… và đã tìm ra cách biểu diễn chúng bằng một khái niệm toán học, được gọi là tập mờ.

Tuy nhiên, việc mô hình hóa quá trình tư duy lập luận của con người là một vấn đề khó luôn thách thức các nhà nghiên cứu bởi đặc trưng giàu thông tin của ngôn ngữ và cơ chế suy luận không những dựa trên tri thức mà còn là kinh nghiệm, trực quan cảm nhận theo ngữ cảnh của con người. Cấu trúc thứ tự cảm sinh trên các khái niệm mờ biểu thị bằng các giá trị ngôn ngữ không được thể hiện trên các tập mờ vì hàm thuộc của chúng lại không sánh được với nhau. Hơn thế nữa, việc thiết lập các tập mờ của các giá trị ngôn ngữ một cách cố định dựa theo chủ quan của người thiết lập, trong khi một giá trị ngôn ngữ sẽ mang ngữ nghĩa tương đối khác nhau trong các bài toán khác nhau [2], [7], [8].

Nhằm khắc phục phần nào những nhược điểm trên, năm 1990, N.C. Ho & W. Wechler đã khởi xướng phương pháp tiếp cận đại số đến cấu trúc tự nhiên của miền giá trị của các biến ngôn ngữ [23]-[27]. Theo cách tiếp cận này, mỗi giá trị ngôn ngữ của một biến ngôn ngữ nằm trong một cấu trúc đại số gọi là đại số gia tử (ĐSGT). Dựa trên những tính chất ngữ nghĩa của ngôn ngữ được phát hiện, bằng phương pháp tiên đề hóa nhiều tác giả đã tập trung phát triển lý thuyết ĐSGT với các kết quả như ĐSGT mở rộng, ĐSGT mịn hóa, ĐSGT mở rộng đầy đủ, ĐSGT PN-không thuần nhất. Trên cơ sở đó, đã có nhiều nghiên cứu về lý thuyết cũng như ứng dụng của nhiều tác giả trong các lĩnh vực: điều khiển mờ và lập luận mờ [3], [4], [5], cơ sở dữ liệu mờ [1], [63], phân lớp mờ [28], [31],… và đã cho chúng ta nhiều kết quả rất khả quan, có khả năng ứng dụng tốt. Những kết quả này, dù chưa nhiều, nhưng đã cho thấy ý nghĩa cũng như thế mạnh của ĐSGT trong ứng dụng và đây là một hướng nghiên cứu đang được nhiều nhà khoa học quan tâm.

Thêm vào đó, với sự bùng nổ dữ liệu của thời đại thông tin như hiện nay, lượng dữ liệu được tạo ra hàng ngày là rất lớn. Khối lượng thông tin dữ liệu khổng lồ này vượt khỏi giới hạn khả năng ghi nhớ và xử lý của con người. Nhu cầu cần thiết là nghĩ đến các quá trình tự động tìm kiếm các thông tin hữu ích, các quan hệ ràng buộc dữ liệu trong các kho dữ liệu lớn để phát hiện các tri thức, các quy luật hay khuynh hướng dữ liệu hỗ trợ con người phán đoán, nhận xét, ra quyết định. Nhằm đáp ứng các nhu cầu đó, nhiều nhà khoa học đã đề xuất, nghiên cứu và phát triển các phương pháp mới trong khai phá dữ liệu. Các bài toán được biết đến trong lĩnh vực này như phân lớp và nhận dạng mẫu, hồi quy và dự báo, phân cụm, khai phá luật kết hợp,... với rất nhiều kết quả đã được công bố [6], [10], [11], [32], [36], [38], [49],...

Phân lớp dữ liệu là một quá trình quan trọng của khai phá dữ liệu, đó là quá trình chia các đối tượng dữ liệu thành các lớp dựa trên các đặc trưng của tập dữ liệu. Quá trình phân lớp dữ liệu bao gồm việc xây dựng một mô hình dựa trên việc phân tích các mẫu dữ liệu sẵn có và sử dụng mô hình để phân lớp các dữ liệu chưa biết. Các phương pháp thường được sử dụng trong quá trình học phân lớp như: thống kê, mạng nơron, cây quyết định,… trong đó cây quyết định là một giải pháp hữu hiệu để mô tả quá trình phân lớp dữ liệu. Do cây quyết định

rất hữu dụng nên đã có nhiều nghiên cứu để xây dựng nó mà nổi bật là các thuật toán học quy nạp như ID3, C45 [41], [67],… CART, SLIQ, SPRINT [14], [52], [74],… Fuzzy ID3 [46], [69], [70],… LDT, LID3 [40], [55], [84], [85],...

Trong việc phân lớp dữ liệu bằng cây quyết định, quá trình xây dựng tại mỗi nút của cây, các thuật toán đều tính lượng thông tin và chọn thuộc tính tương ứng có lượng thông tin tối đa làm nút phân tách trên cây. Các thuộc tính này sẽ chia tập mẫu thành các lớp mà mỗi lớp có một phân loại duy nhất hay ít nhất phải có triển vọng đạt được điều này, nhằm để đạt được cây có ít nút nhưng có khả năng dự đoán cao. Tuy vậy, các cách tiếp cận cho việc huấn luyện cây quyết định hiện nay vẫn còn nhiều vấn đề cần giải quyết:

- Breiman L, Friedman J. [14], Guang-Bin Huang, Hongming Zhou [24], Kishor Kumar Reddy [43], Patil N. [54], Quinlan J. R. [60-62], Shou-Hsiung Cheng, Yi Yang và các cộng sự [67], [78] đã dựa vào khái niệm Entropi thông tin để tính lợi ích thông tin và tỷ lệ lợi ích thông tin của các thuộc tính tại thời điểm phân chia các nút. Hướng tiếp cận này cho chúng ta các thuật toán có độ phức tạp thấp nhưng việc phân chia *k-phân* trên các thuộc tính rời rạc làm cho số nút của cây tăng nhanh, làm tăng chiều rộng của cây, dẫn đến tình trạng quá khớp trên cây kết quả nên ảnh hưởng đến khả năng dự đoán.

- Manish Mehta, Jorma Rissanen, Rakesh Agrawal [47], [48], Narasimha Prasad, Mannava Munirathnam Naidu [52], Zhihao Wang, Junfang Wang, Yonghua Huo, Hongze Qiu [87], Haitang Zhang và các cộng sự [32] dựa vào việc tính hệ số *Gini* và tỷ lệ hệ số *Gini* của các thuộc tính để lựa chọn điểm phân chia. Theo hướng tiếp cận này, chúng ta không cần đánh giá mỗi thuộc tính mà chỉ cần tìm điểm chia tách tốt nhất cho mỗi thuộc tính đó. Tuy nhiên, tại mỗi thời điểm chúng ta phải tính một số lượng lớn hệ số *Gini* cho các giá trị rời rạc nên chi phí về độ phức tạp tính toán cao và cây kết quả mất cân xứng vì phát triển nhanh theo chiều sâu, số nút trên cây lớn.

- B. Chandra [11], Chida A. [16], Daveedu Raju Adidela, Jaya Suma. G, Lavanya Devi. G [19], Hesham A. Hefny, Ahmed S. Ghiduk [26], Hou Yuan-long, Chen Ji-lin, Xing Zong-yi [32], Marcos E. Cintra, Maria C. Monard [49], Zeinalkhani M., Eftekhari M. [83] và các cộng sự đã thông qua lý thuyết tập mờ để tính lợi ích thông tin của các thuộc tính mờ cho quá trình phân lớp. Hướng

tiếp cận này đã giải quyết được các giá trị mờ trong tập huấn luyện thông qua việc xác định các hàm thuộc, từ đó các bộ giá trị này có thể tham gia vào quá trình huấn luyện. Cách làm này đã giải quyết được hạn chế là bỏ qua các giá trị dữ liệu mờ của cách tiếp phân lớp rõ. Tuy vậy, hiện vẫn còn gặp phải những hạn chế xuất phát từ bản thân nội tại của lý thuyết tập mờ: hàm thuộc của chúng không so sánh được với nhau, xuất hiện sai số lớn tại quá trình xấp xỉ, phụ thuộc vào sự chủ quan, giá trị ngôn ngữ còn thiếu một cơ sở đại số làm nền tảng.

- Suzan Kantarci-Savas, Efendi Nasibov [69], Zengchang Qin, Jonathan Lawry, Yongchuan Tang [84], [85] và các cộng sự đã xác định các giá trị ngôn ngữ cho tập dữ liệu mờ và xây dựng cây quyết định ngôn ngữ (Linguistic Decision Tree - LDT) bằng phương pháp LID3. Với việc xây dựng các nhãn ngôn ngữ cho các giá trị mờ dựa vào xác suất của các nhãn liên kết trong khi vẫn giữ được các giá trị rõ đã biết, hướng tiếp cận này đã làm giảm sai số đáng kể cho quá trình huấn luyện. Tuy vậy, hướng tiếp cận này làm này sẽ làm phát sinh cây đa phân do có sự phân chia lớn theo chiều ngang tại các nút ngôn ngữ khi tập giá trị ngôn ngữ của thuộc tính mờ lớn.

- N. C. Ho, N. C. Hao, L. A. Phuong, L. X. Viet, L. X. Vinh, N. V. Long, N. V. Lan [1-5], [27], [28], [29], [30], [31] và các cộng sự đã chỉ ra phương pháp định lượng ngữ nghĩa theo điểm dựa trên ĐSGT, nhằm thuần nhất dữ liệu về các giá trị số hay giá trị ngôn ngữ và cách thức truy vấn dữ liệu trên thuộc tính này. Bài toán xây dựng cây quyết định mờ lúc này có thể sử dụng các thuật toán học theo cách tiếp cận cây quyết định rõ trong một ĐSGT đã xây dựng. Tuy vậy, hướng tiếp cận này vẫn còn một số vấn đề như: vẫn xuất hiện sai số lớn khi thuần nhất theo điểm mờ, khó đưa ra dự đoán khi có sự đan xen ở điểm phân chia mờ của cây kết quả, phụ thuộc vào miền trị $[\psi_{min}, \psi_{max}]$ từ miền giá trị rõ của thuộc tính mờ.

Thêm vào đó, tất cả các thuật toán học phân lớp bằng cây quyết định hiện có đều phụ thuộc lớn vào việc chọn tập mẫu của người huấn luyện. Khi chúng ta chọn tập mẫu không đặc trưng thì cây quyết định được sinh ra sẽ không có khả năng dự đoán. Mà trong thế giới thực, việc lưu trữ dữ liệu tại các kho dữ liệu nghiệp vụ nhằm nhiều mục đích khác nhau. Nhiều thông tin phục vụ tốt cho việc dự đoán nhưng nhiều thông tin khác chỉ có ý nghĩa lưu trữ thông thường, phục

vụ cho việc diễn giải thông tin. Các nhóm thuộc tính này làm phức tạp mẫu nên tăng chi phí cho quá trình huấn luyện, quan trọng hơn là chúng gây nhiễu nên cây được xây dựng không có hiệu quả cao. Vì vậy, làm sao để phân lớp dữ liệu bằng cây quyết định đạt hiệu quả là vấn đề mà các nhà khoa học hiện nay vẫn đang quan tâm, nghiên cứu.

Xuất phát từ việc tìm hiểu, nghiên cứu các đặc điểm và các thách thức về các vấn đề của phân lớp dữ liệu bằng cây quyết định, luận án đã chọn đề tài là: "*Phân lớp dữ liệu bằng cây quyết định mờ dựa trên đại số gia tử*".

2. Đối tượng và phạm vi nghiên cứu

Phân lớp dữ liệu là vấn đề lớn và quan trọng của khai phá dữ liệu. Cây quyết định là giải pháp hữu hiệu của bài toán phân lớp, nó bao gồm từ mô hình cho quá trình học đến các thuật toán huấn luyện cụ thể để xây dựng cây. Luận án tập trung nghiên cứu mô hình linh hoạt cho quá trình huấn luyện cây từ tập mẫu huấn luyện, nghiên cứu phương pháp xử lý giá trị ngôn ngữ và xây dựng các thuật toán học phân lớp dữ liệu bằng cây quyết định mờ đạt nhằm đạt hiệu quả trong dự đoán và đơn giản đối với người dùng.

3. Phương pháp nghiên cứu

Luận án tập trung vào các phương pháp chính:

- *Phương pháp nghiên cứu tài liệu, tổng hợp và hệ thống hóa:* tìm kiếm, thu thập tài liệu về các công trình nghiên cứu đã được công bố ở các bài báo đăng ở các hội thảo và tạp chí lớn; nghiên cứu các phương pháp xây dựng cây quyết định đã có, nhằm phân tích những thuận lợi và khó khăn trong quá trình học phân lớp dữ liệu bằng cây quyết định. Đề xuất các thuật toán học phân lớp bằng cây quyết định mờ theo hướng tăng độ chính xác cho quá trình sử dụng cây kết quả để dự đoán nhằm thỏa mãn mục tiêu cụ thể của người dùng.

- *Phương pháp thực nghiệm khoa học:* sử dụng các bộ dữ liệu chuẩn không chứa giá trị mờ Northwind và các bộ dữ liệu có chứa giá trị mờ Mushroom và Adult cho quá trình thử nghiệm, đánh giá. Thực hiện việc thử nghiệm, đánh giá các thuật toán đã đề xuất trong các công trình trước đây với các thuật toán được đề xuất trong luận án nhằm minh chứng cho tính hiệu quả về độ chính xác trong quá trình dự đoán.

4. Mục tiêu và nội dung của luận án

Sau khi nghiên cứu và phân tích các vấn đề về phân lớp dữ liệu bằng cây quyết định của các nghiên cứu trong và ngoài nước, luận án đưa ra mục tiêu nghiên cứu chính như sau:

- Xây dựng mô hình học phân lớp dữ liệu bằng cây quyết định mờ và phương pháp trích chọn đặc trưng để chọn tập mẫu huấn luyện cho quá trình học phân lớp. Đề xuất phương pháp xử lý giá trị ngôn ngữ của các thuộc tính chưa thuần nhất dựa vào ĐSGT.

- Đề xuất các thuật toán học bằng cây quyết định mờ cho bài toán phân lớp nhằm đạt hiệu quả trong dự đoán và đơn giản đối với người dùng.

Để đáp ứng cho các mục tiêu nghiên cứu trên, luận án tập trung nghiên cứu các nội dung chính sau:

- Nghiên cứu các thuật toán học cây truyền thống CART, ID3, C45, C50, SLIQ, SPRINT trên mỗi tập mẫu huấn luyện để tìm phương pháp học đạt hiệu quả dự đoán cao.

- Nghiên cứu xây dựng phương pháp trích chọn đặc trưng để chọn tập mẫu huấn luyện cho việc học cây quyết định từ các kho dữ liệu nghiệp vụ.

- Nghiên cứu xây dựng một mô hình học phân lớp dữ liệu bằng cây quyết định linh hoạt từ tập mẫu huấn luyện.

- Nghiên cứu để đề xuất phương pháp xử lý giá trị ngôn ngữ của các thuộc tính chưa thuần nhất trên tập mẫu huấn luyện dựa vào bản chất của ĐSGT.

- Nghiên cứu để đề xuất các thuật toán học phân lớp bằng cây quyết định mờ nhằm đạt hiệu quả trong dự đoán và đơn giản đối với người dùng. Phân tích và đánh giá kết quả của các thuật toán học đã đề xuất với các thuật toán khác trên các bộ mẫu chuẩn không chứa giá trị mờ Northwind và các bộ dữ liệu có chứa giá trị mờ Mushroom, Adult để đối sánh.

5. Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học

Những đóng góp chính của luận án về khoa học:

- Xây dựng mô hình học phân lớp dữ liệu bằng cây quyết định mờ từ tập

mẫu huấn luyện. Đề xuất phương pháp trích chọn đặc trưng để chọn tập mẫu huấn luyện cho việc học phân lớp bằng cây quyết định từ các kho dữ liệu nghiệp vụ, nhằm hạn chế sự phụ thuộc ý kiến của chuyên gia trong quá trình chọn tập mẫu huấn luyện.

- Đề xuất phương pháp xử lý giá trị ngôn ngữ của các thuộc tính chưa thuần nhất trên tập mẫu huấn luyện dựa vào bản chất của ĐSGT.

- Luận án đã xây dựng các hàm mục tiêu của bài toán phân lớp bằng cây quyết định, sử dụng tính có thứ tự của các giá trị ngôn ngữ trong ĐSGT. Đưa ra các khái niệm đối sánh khoảng mờ, khoảng mờ lớn nhất để từ đó đề xuất các thuật toán học cây quyết định mờ MixC4.5, FMixC4.5, HAC4.5 và HAC4.5* cho bài toán phân lớp, nhằm góp phần cải thiện, nâng cao độ chính xác trong quá trình học phân lớp dữ liệu bằng cây quyết định cho bài toán phân lớp dữ liệu .

Ý nghĩa thực tiễn

- Góp phần chứng tỏ khả năng ứng dụng phong phú của ĐSGT trong biểu diễn và xử lý thông tin mờ, không chắc chắn.

- Luận án đã góp phần vào việc giải quyết vấn đề định lượng cho các giá trị ngôn ngữ mà không phụ thuộc cố định vào miền trị *Min-Max* của các giá trị kinh điển của thuộc tính mờ trong tập mẫu.

- Dựa trên các khái niệm về khoảng mờ và khoảng mờ lớn nhất, luận án đã đề xuất các thuật toán cho quá trình học cây, nhằm tăng khả năng dự đoán cho bài toán phân lớp dữ liệu bằng cây quyết định. Làm phong phú thêm các phương pháp học cho bài toán phân lớp nói chung và phân lớp bằng cây quyết định nói riêng.

- Luận án có thể được sử dụng làm tài liệu tham khảo cho các sinh viên đại học, học viên cao học ngành Công nghệ thông tin nghiên cứu về học phân lớp bằng cây quyết định.

6. Bố cục của luận án

Ngoài phần mở đầu, kết luận và tài liệu tham khảo, luận án được chia làm 3 chương nội dung:

Chương 1: cơ sở lý thuyết về đại số gia tử và tổng quan phân lớp dữ liệu bằng cây quyết định. Chương này tập trung nghiên cứu, phân tích và đánh giá

các vấn đề liên quan mật thiết đến luận án như: khái niệm mờ, tập mờ và khái niệm biến ngôn ngữ, phương pháp lập luận xấp xỉ trực tiếp trên ngôn ngữ, khái niệm và tính chất về ĐSGT. Luận án cũng trình bày các vấn đề cơ bản của bài toán phân lớp dữ liệu bằng cây quyết định, các hạn chế trên cây quyết định truyền thống và sự cần thiết của bài toán phân lớp bằng cây quyết định mờ. Ở đây, luận án đã phát biểu hình thức bài toán phân lớp dữ liệu bằng cây quyết định và cũng tập trung nghiên cứu, phân tích và đánh giá các công trình nghiên cứu đã công bố gần đây, chỉ ra các vấn đề còn tồn tại để xác định mục tiêu và nội dung cần giải quyết của luận án.

Chương 2: phân lớp dữ liệu bằng cây quyết định mờ theo phương pháp đối sánh điểm mờ dựa trên đại số gia tử. Chương này của luận án tập trung phân tích sự ảnh hưởng của tập mẫu huấn luyện đối với hiệu quả cây kết quả thu được, trình bày một phương pháp nhằm trích chọn được tập mẫu huấn luyện đặc trưng phục vụ cho quá trình huấn luyện; phân tích, đưa ra các khái niệm về tập mẫu không thuần nhất, giá trị ngoại lai và xây dựng thuật toán để có thể thuần nhất cho các thuộc tính có chứa các giá trị này. Đề xuất các thuật toán MixC4.5 và FMixC4.5 phục vụ quá trình học cây quyết định trên tập mẫu không thuần nhất; thử nghiệm trên các cơ sở dữ liệu không chứa dữ liệu mờ Northwind và có chứa thông tin mờ Mushroom để đối sánh về khả năng dự đoán của cây kết quả sau khi huấn luyện.

Chương 3: phương pháp huấn luyện cây quyết định mờ cho bài toán phân lớp dữ liệu dựa trên đối sánh khoảng mờ. Chương này của luận án tập trung nghiên cứu quá trình học cây quyết định mờ nhằm đạt hai mục tiêu đã đề ra là $f_h(S) \rightarrow max$ và $f_n(S) \rightarrow min$. Trên cơ sở nghiên cứu mối tương quan của các khoảng mờ, luận án đề xuất phương pháp đối sánh dựa trên khoảng mờ, xây dựng phương pháp nhằm có thể định lượng cho các giá trị của thuộc tính không thuần nhất, chưa xác định *Min-Max* của tập huấn luyện và xây dựng thuật toán học phân lớp bằng cây quyết định dựa trên khoảng mờ HAC4.5 nhằm đạt được mục tiêu $f_h(S) \rightarrow max$. Cùng với mục tiêu cần đạt được $f_n(S) \rightarrow min$, luận án cũng đề xuất khái niệm khoảng mờ lớn nhất, đưa ra thuật toán HAC4.5* nhằm đồng thời đạt được hai mục tiêu đề ra, đó là tính hiệu quả của quá trình phân lớp và tính đơn giản và dễ hiểu đối với người dùng. Các kết quả của luận án được phân tích, đánh giá và cài đặt thử nghiệm trên các cơ sở dữ liệu có chứa thông tin

mờ Mushroom và Adult nhằm thể hiện tính hiệu quả của các phương pháp đã đề xuất.

Các kết quả chính của luận án đã được báo cáo tại các hội nghị khoa học và senimar, được công bố trong 7 công trình khoa học được đăng trong các hội nghị, tạp chí chuyên ngành trong và ngoài nước:

- 01 bài đăng ở tạp chí Khoa học và Công nghệ trường Đại học Khoa học Huế.
- 01 bài đăng ở tạp chí Khoa học Đại học Huế.
- 01 bài đăng ở kỷ yếu Hội thảo quốc gia Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR).
- 02 bài đăng ở Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ thông tin và Truyền thông, Tạp chí Thông tin, Khoa học và Công nghệ, Bộ Thông tin và Truyền thông.
- 01 bài đăng ở tạp chí chuyên ngành Tin học và Điều khiển (Journal of Computer Science and Cybernetics).
- 01 bài đăng ở tạp chí quốc tế International Journal of Research in Engineering and Science (IJRES).

Chương 1.

CƠ SỞ LÝ THUYẾT VỀ ĐẠI SỐ GIA TỬ VÀ TỔNG QUAN PHÂN LỚP DỮ LIỆU BẰNG CÂY QUYẾT ĐỊNH

Với mục tiêu nhằm giải quyết các vấn đề của bài toán phân lớp dữ liệu bằng cây quyết định mờ, Chương 1 của luận án trình bày một số vấn đề liên quan đến bài toán phân lớp dữ liệu bằng cây quyết định, cây quyết định mờ và các kiến thức cơ bản của đại số gia tử dùng để nghiên cứu trong quá trình học phân lớp dữ liệu bằng cây quyết định. Nội dung của chương này bao gồm: tập mờ, đại số gia tử và các phương pháp học phân lớp dữ liệu bằng cây quyết định.

1.1. Lý thuyết tập mờ

1.1.1.Tập mờ và thông tin không chắc chắn

Thực tế đã chứng minh khái niệm mờ luôn tồn tại, hiện hữu trong các bài toán ứng dụng, trong cách suy luận của con người, ví dụ như *trẻ, rất trẻ, hơi già, quá già*,... Vì thế, với việc quan niệm các đối tượng được sử dụng phải luôn rõ ràng ở trong logic cổ điển sẽ không không đủ tốt cho việc miêu tả các vấn đề của bài toán thế giới thực. Như vậy, rất cần một tiếp cận nghiên cứu mới so với logic cổ điển.

Năm 1965, L. A. Zadeh đã đề xuất hình thức hóa toán học của khái niệm mờ [79], từ đó lý thuyết tập mờ được hình thành và ngày càng thu hút sự nghiên cứu của nhiều tác giả. Bằng các phương pháp tiếp cận khác nhau, các nhà nghiên cứu như Dubois, Prade, Mariana, Ishibuchi, Herrera, Yakun Hu,… đã đưa ra những kết quả cả về lý thuyết và ứng dụng cho nhiều lĩnh vực như: điều khiển mờ, cơ sở dữ liệu mờ, khai phá dữ liệu mờ,... [11], [23], [50], [61], [76], [77].

Ý tưởng nổi bật của khái niệm tập mờ của Zadeh là từ những khái niệm trừu tượng về ngữ nghĩa của thông tin mờ, không chắc chắn như *trẻ-già, nhanh-chậm, cao-thấp, xấu-đẹp*,… ông đã tìm cách biểu diễn chúng bằng một khái

niệm toán học, được gọi là tập mờ, như là một sự khái quát trực tiếp của khái niệm tập hợp kinh điển.

Định nghĩa 1.1. [80] Cho một tập vũ trụ V khác rỗng. Một tập mờ A trên tập vũ trụ V được đặc trưng bởi hàm thuộc:

$$\mu_A(x): V \rightarrow [0,1] \quad (1.1)$$

với $\mu_A(x)$ là độ thuộc của phần tử x trong tập mờ A.

Một tập mờ hữu hạn được ký hiệu bởi:

$$A = \frac{\mu_{A(x_1)}}{x_1} + \frac{\mu_{A(x_2)}}{x_2} + \cdots + \frac{\mu_{A(x_n)}}{x_n} \quad (1.2)$$

Một tập mờ vô hạn được ký hiệu bởi:

$$A = \int \mu_A(x)/x \quad (1.3)$$

Ví dụ 1.1. Xét tập A gồm 5 người x_1, x_2, …, x_5 tương ứng có tuổi là 10, 15, 50, 55, 70. Gọi $A^{\sim}$ là tập hợp các người có tuổi là "*Trẻ*". Khi đó ta có thể xây dựng hàm thuộc như sau: $\mu_{Trẻ}(10) = 0.95$; $\mu_{Trẻ}(15) = 0.75$; $\mu_{Trẻ}(50) = 0.35$; $\mu_{Trẻ}(55) = 0.30$; $\mu_{Trẻ}(70) = 0.05$ và tập mờ $A^{\sim} = \frac{0.95}{x_1} + \frac{0.75}{x_2} + \frac{0.50}{x_3} + \frac{0.35}{x_4} + \frac{0.05}{x_5}$.

Giá trị hàm $\mu_A(x)$ càng gần tới 1 thì mức độ thuộc của x trong A càng cao. Tập mờ là sự mở rộng của khái niệm tập hợp kinh điển. Thật vậy, khi A là một tập hợp kinh điển, hàm thuộc của nó, $\mu_A(x)$, chỉ nhận 2 giá trị 1 hoặc 0, tương ứng với x có nằm trong A hay không.

Một khái niệm quan trọng trong việc tiếp cận giải bài toán phân lớp đó là phân hoạch mờ. Về hình thức, chúng ta có định nghĩa như sau:

Định nghĩa 1.2. [63] Cho p điểm cố định m_1, m_2, ..., m_p với $m_1 < m_2 < ... < m_p$ ở trong tập $V = [a, b] \subset R$. Khi đó tập Φ gồm p tập mờ A_1, A_2, ..., A_p (với μ_{A_1}, μ_{A_2},…, μ_{A_p} là các hàm thuộc tương ứng) trên V được gọi là một phân hoạch mờ của V nếu các điều kiện sau thỏa mãn, $\forall k = 1, ..., p$.

1. $\mu_{A_k}(m_k) = 1$; m_k được gọi là một điểm trong nhân của A_k.

2. Nếu $x \notin [m_{k-1}, m_{k+1}]$, $\mu_{A_k}(x) = 0$; trong đó $m_0 = m_1 = a$ và $m_{p+1} = m_p = b$.

3. $\mu_{A_k}(x)$ liên tục.

4. $\mu_{A_k}(x)$ đơn điệu tăng trên [m_{k-1}, m_k] và đơn điệu giảm trên [m_k, m_{k+1}].

5. $\forall x \in V, \exists k : \mu_{A_k}(x) > 0$. Tất cả mọi điểm trong V đều thuộc một lớp của phân hoạch này với độ thuộc nào đó khác không.

Thực tế các khái niệm mờ trong các bài toán ứng dụng rất đa dạng và khó để xác định được các hàm thuộc của chúng một cách chính xác, thông thường dựa trên ngữ cảnh mà khái niệm mờ đó đang được sử dụng. Ngược lại một khái niệm mờ có thể được mô hình hóa bởi các tập mờ. Trên cơ sở mối quan hệ này, L. A. Zadeh đã đưa ra khái niệm biến ngôn ngữ.

1.1.2. Biến ngôn ngữ

Khái niệm biến ngôn ngữ đã được L. A. Zadeh giới thiệu, là một công cụ quan trọng để phát triển phương pháp lập luận xấp xỉ dựa trên logic mờ [79], [81]. Ông đã viết: "*Khi thiếu hụt tính chính xác bề ngoài của những vấn đề phức tạp cố hữu, một cách tự nhiên là tìm cách sử dụng các biến gọi là biến ngôn ngữ; đó là các biến mà các giá trị của chúng không phải là các số mà là các từ hoặc các câu trong một ngôn ngữ tự nhiên hoặc nhân tạo. Động lực cho việc sử dụng các từ, các câu hơn các số là đặc trưng ngôn ngữ của các từ, các câu thường là ít xác định hơn của số*".

Định nghĩa 1.3. [81] Biến ngôn ngữ là một bộ gồm năm thành phần (*X*, *T*(*X*), *U*, *R*, *M*), trong đó *X* là tên biến, *T*(*X*) là tập các giá trị ngôn ngữ của biến *X*, *U* là không gian tham chiếu của biến cơ sở *u*, mỗi giá trị ngôn ngữ xem như là một biến mờ trên *U* kết hợp với biến cơ sở *u*, *R* là một qui tắc cú pháp sinh các giá trị ngôn ngữ cho tập *T*(*X*), *M* là qui tắc ngữ nghĩa gán mỗi giá trị ngôn ngữ trong *T*(*X*) với một tập mờ trên *U*.

Ví dụ 1.2. Xét biến ngôn ngữ *Age*, tức là *X* = *Age*, biến cơ sở *u* thể hiện tuổi con người có miền xác định *U* = [1, 100]. Khi đó, các giá trị ngôn ngữ tương ứng của biến *Age* là *T*(*Age*) có thể bao gồm các giá trị: {*young, old, very old, old, possible old, less old, less young, quite young, more young*,...}. Các giá trị ngôn ngữ *young* và *old* được gọi là các giá trị nguyên thủy. Mỗi giá trị ngôn ngữ trong *T*(*Age*) là tên của một biến mờ trên *U*, tức là biến có thể nhận giá trị trên *U* với một mức độ tương thích trong đoạn [0, 1], ràng buộc trên mỗi giá trị ngôn ngữ

hình thành ngữ nghĩa cho giá trị ngôn ngữ đó. Chẳng hạn với giá trị nguyên thủy *old*, quy tắc gán ngữ nghĩa *M* cho *old* bằng tập mờ sau:

$$M(old) = \{(u, \mu_{old}(u)) : u \in [0, 100]\}$$

trong đó $\mu_{old}(u) = max(min(1, \frac{(u-60)}{20}), 0)$ là một cách chọn hàm thuộc cho khái niệm mờ *old*. Ngữ nghĩa các giá trị ngôn ngữ khác trong *T*(*Age*) có thể tính thông qua tập mờ của các giá trị nguyên thủy bởi các phép toán tương ứng với các gia tử tác động, chẳng hạn như gia tử *very*, *more* or *less*,...[2-3], [15], [66].

Vấn đề mô hình các hóa các gia tử ngôn ngữ sử dụng tập mờ đã được nhiều nhà nghiên cứu quan tâm, chẳng hạn L. A. Zadeh [79], [81], Mingsheng Ying và Bernadette BouchonMeunier [51]. Mặt khác, chúng ta thấy việc gán ngữ nghĩa cho biến ngôn ngữ không có quy tắc ràng buộc nhất định như cách chọn hàm thuộc $\mu_{old}(u)$ ở trên, hơn nữa các phép toán trên tập mờ nói chung không đóng. Vì vậy trong các nghiên cứu của mình về biến ngôn ngữ và lập luận xấp xỉ, L. A. Zadeh luôn nhấn mạnh hai đặc trưng quan trọng sau đây của biến ngôn ngữ:

1. *Tính phổ quát:* miền giá trị của hầu hết các biến ngôn ngữ có cùng cấu trúc cơ sở theo nghĩa các giá trị ngôn ngữ tương ứng là giống nhau ngoại trừ phần tử sinh nguyên thủy.

2. *Tính độc lập ngữ cảnh của gia tử và liên từ:* ngữ nghĩa của các gia tử và liên từ hoàn toàn độc lập với với ngữ cảnh, khác với giá trị nguyên thủy của các biến ngôn ngữ lại phụ thuộc vào ngữ cảnh. Do đó khi tìm kiếm mô hình cho các gia tử và liên từ chúng ta không phải quan tâm đến giá trị nguyên thủy của biến ngôn ngữ đang xét.

Các đặc trưng này cho phép chúng ta sử dụng cùng một tập gia tử và xây dựng một cấu trúc toán học duy nhất cho miền giá trị của các biến ngôn ngữ khác nhau. Dựa trên khái niệm của biến ngôn ngữ, lý thuyết lập luận xấp xỉ nhằm mô hình hóa quá trình suy luận của con người đã được L. A. Zadeh đề xuất và nghiên cứu [80].

Vấn đề sử dụng tập mờ để biểu diễn các giá trị ngôn ngữ và dùng các phép toán trên tập mờ để biểu thị các gia tử ngôn ngữ đã cho phép thực hiện các thao tác dữ liệu mờ, một phần nào đã đáp ứng được nhu cầu thực tế của con

người. Tuy nhiên, theo cách sử dụng tập mờ cho thấy vẫn có nhiều hạn chế do việc xây dựng các hàm thuộc và xấp xỉ các giá trị ngôn ngữ bởi các tập mờ còn mang tính chủ quan, phụ thuộc nhiều vào ý kiến chuyên gia cho nên dễ mất mát thông tin. Mặc khác, bản thân các giá trị ngôn ngữ có một cấu trúc thứ tự nhưng ánh xạ gán nghĩa sang tập mờ, không bảo toàn cấu trúc đó nữa. Do đó, vấn đề đặt ra là cần có một cấu trúc toán học mô phỏng chính xác hơn cấu trúc ngữ nghĩa của một khái niệm mờ.

1.2. Đại số gia tử

Nhằm để giải quyết các hạn chế của tập mờ, đại số gia tử được ra đời do đề xuất của N. C. Ho và W. Wechler vào năm 1990 [29]. Việc sử dụng khái niệm đại số gia tử tuyến tính, đầy đủ đã giải đáp tốt các hạn chế đã nêu. Đến nay, đã có nhiều nghiên cứu về lý thuyết cũng như ứng dụng của nhiều tác giả trong và ngoài nước, đã cho chúng ta nhiều kết quả rất khả quan, có khả năng ứng dụng lớn. Các kết quả nổi bật như: điều khiển mờ và lập luận mờ [3], [4], [5], cơ sở dữ liệu mờ [1], [2], [63], phân lớp mờ [28], [31],…

1.2.1. Khái niệm đại số gia tử

Xét một ví dụ có miền ngôn ngữ của biến chân lý *TRUTH* gồm các từ sau: *Dom*(*TRUTH*) = {*đúng, sai, rất đúng, rất sai, ít nhiều đúng, ít nhiều sai, khả năng đúng, khả năng sai, xấp xỉ đúng, xấp xỉ sai, ít đúng, ít sai, rất khả năng đúng, rất khả năng sai*,...}, trong đó *đúng, sai* là các từ nguyên thuỷ, các từ nhấn *rất, ít nhiều, khả năng, xấp xỉ, ít,…* được gọi là các gia tử. Khi đó, miền ngôn ngữ $T = Dom(TRUTH)$ có thể biểu thị như một đại số $\underline{X} = (X, G, H, \leq)$, trong đó G là tập các từ nguyên thuỷ {*thấp, cao*} được xem là các phần tử sinh. $H = H^+ \cup H^-$ là tập các gia tử dương, âm và được xem như là các phép toán một ngôi, quan hệ $\leq$ trên các từ là quan hệ thứ tự được "*cảm sinh*" từ ngữ nghĩa tự nhiên. Tập X được sinh ra từ G bởi các phép tính trong H.

Như vậy, mỗi phần tử của X sẽ có dạng biểu diễn $x = h_n h_{n-1}.....h_1 c$, $c \in G$. Tập tất cả các phần tử được sinh ra từ một phần tử x được ký hiệu là $H(x)$. Nếu G có đúng hai từ nguyên thuỷ mờ, thì một được gọi là phần tử sinh dương ký hiệu là c^+, một gọi là phần tử sinh âm ký hiệu là c^- và ta có $c^- < c^+$. Trong ví dụ trên *đúng* là phần tử sinh dương còn *sai* là phần tử sinh âm.

Như vậy, một cách tổng quát, cho ĐSGT $\underline{X} = (X, G, H, \leq)$, với $G = \{0, c^-, W, c^+, 1\}$, trong đó c^+ và c^- tương ứng là phần tử sinh dương và âm, X là tập nền. $H = H^+ \cup H^-$ với giả thiết $H^+ = \{h_1, h_2, .., h_p\}$, $H^- = \{h_{-q}, ..., h_{-1}\}$, $h_1 < h_2 < ... < h_p$ và $h_{-q} < ... < h_{-1}$ là dãy các gia tử.

Trong ĐSGT tuyến tính, chúng ta bổ sung thêm vào hai phép tính Σ và Φ với ngữ nghĩa là cận trên đúng và cận dưới đúng của tập $H(x)$, tức là Σx = *supremum* $H(x)$ và Φx = *infimum* $H(x)$, khi đó ĐSGT tuyến tính được gọi là ĐSGT tuyến tính đầy đủ và được ký hiệu $\underline{X} = (X, G, H, \Sigma, \Phi, \leq)$.

Định nghĩa 1.4. [29] Cho hai gia tử $h, k \in H$ và $x \in X$.

1. Gia tử k được gọi là dương đối với gia tử h nếu $hx \geq x$ thì $khx \geq hx$ hoặc nếu $hx \leq x$ thì $khx \leq hx$.

2. Gia tử k được gọi là âm đối với gia tử h nếu $hx \geq x$ thì $khx \leq hx$ hoặc nếu $hx \leq x$ thì $khx \geq hx$.

Định nghĩa 1.5. [29] Cho hai gia tử $h, k \in H$ và $x \in X$.

1. Gia tử h được gọi là ngược với gia tử k nếu ta có: $hx \geq x$ và $kx \leq x$ hoặc $hx \leq x$ và $kx \geq x$.

2. Gia tử h được gọi là tương thích với gia tử k nếu ta có $hx \geq x$ và $kx \geq x$ hoặc $hx \leq x$ và $kx \leq x$.

Định nghĩa 1.6. [29] Giả sử trong H^+ có phần tử lớn nhất là V *và trong* H^- có phần tử lớn nhất là L.

1. Phần tử sinh $c \in G$ gọi là phần tử sinh dương (hoặc âm) nếu: $c \leq Vc$ (hoặc $c \geq Lc$)

2. Với $c^+, c^- \in G, h \in H^+$ ta có: $c^+ < hc^+ < Vc^+$ hoặc $Vc^- < hc^- < c^-$

3. Với $c^+, c^- \in G, h \in H^-$ ta có: $c^- < hc^- < Lc^-$ hoặc $Lc^+ < hc^+ < c^+$

Định nghĩa 1.7. [29] Cho $\underline{X} = (X, G, H, \leq)$ là một ĐSGT, với mỗi $x \in X$, độ dài của x được ký hiệu $|x|$ và xác định như sau:

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

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

Như vậy, độ dài của một hạng từ x là số gia tử trong biểu diễn chính tắc của nó đối với phần tử sinh cộng thêm 1, ký hiệu $l(x)$.

Bây giờ chúng ta xét một số tính chất của ĐSGT tuyến tính. Định lý sau cho thấy tính thứ tự ngữ nghĩa của các hạng từ trong ĐSGT.

Định lý 1.1. [29] Cho ĐSGT $\underline{X} = (X, G, H, \leq)$. Khi đó ta có các khẳng định sau:

1. Với mỗi $x \in X$ thì $H(x)$ là tập sắp thứ tự tuyến tính.

2. Nếu X được sinh từ G bởi các gia tử và G là tập sắp thứ tự tuyến tính thì X cũng là tập sắp thứ tự tuyến tính. Hơn nữa nếu $x < x'$, và x, x' là độc lập với nhau, tức là $x \notin H(x')$ và $x' \notin H(x)$, thì $H(x) \leq H(x')$.

Định lý tiếp theo xem xét sự so sánh của hai hạng từ trong miền ngôn ngữ của biến X.

Định lý 1.2. [30] Cho $x = h_n \ldots h_1 u$ và $y = k_m \ldots k_1 u$ là hai biểu diễn chính tắc của x và y đối với u. Khi đó tồn tại chỉ số $j \leq min\{n, m\} + 1$ sao cho với mọi $i < j$ ta có $h_i = k_i$ và:

1. $x < y$ khi và chỉ khi $h_j x_j < k_j x_j$, trong đó $x_j = h_{j-1} \ldots h_1 u$.

2. $x = y$ khi và chỉ khi $m = n$ và $h_j x_j = k_j x_j$.

3. x và y là không sánh được với nhau khi và chỉ khi $h_j x_j$ và $k_j x_j$ là không sánh được.

1.2.2. Các hàm đo của đại số gia tử

Định nghĩa 1.8. [27] Hàm $f : X \to [0, 1]$ gọi là hàm định lượng ngữ nghĩa của X nếu $h, k \in H^+$ hoặc $h, k \in H^-$ và $x, y \in X$, ta có: $\left|\frac{f(hx)-f(x)}{f(kx)-f(x)}\right| = \left|\frac{f(hy)-f(y)}{f(ky)-f(y)}\right|$.

Định nghĩa 1.9. [27] Cho hàm định lượng ngữ nghĩa f của X. Với bất kỳ $x \in X$, tính mờ của x, được ký hiệu là $I(x)$ và được đo bằng đường kính của tập $f(H(x)) \subseteq [0, 1]$.

Định nghĩa 1.10. [27] Hàm fm: $X \to [0,1]$ được gọi là *độ đo tính mờ* trên X nếu thoả mãn các điều kiện sau:

1. $fm(c^-) = W > 0$ *và* $fm(c^+) = 1 - W > 0$

2. Với $c \in \{c^-, c^+\}$ thì $\sum_{i=1}^{p+q} fm(h_i c) = fm(c)$.

3. Với $\forall x, y \in X$, $\forall h \in H$, $\frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)} = \frac{fm(hc)}{fm(c)}$, với $c \in \{c^-, c^+\}$, nghĩa là tỉ số này không phụ thuộc vào x và y, được kí hiệu là $\mu(h)$ gọi là độ đo tính mờ

của gia tử *h*.

1.2.3. Một số tính chất của các hàm đo

Mệnh đề 1.1. [28] Cho $\underline{X} = (X, G, H, \leq)$ là một ĐSGT, ta có:

1. $fm(hx) = \mu(h) \times fm(x)$, với $\forall x \in X$
2. $fm(c^-) + fm(c^+) = 1$
3. $\sum_{i=1}^{p+q} fm(h_i c) = fm(c)$, trong đó $c \in \{c-, c^+\}$
4. $\sum_{i=1}^{p+q} fm(h_i x) = fm(x)$, $\forall x \in X$
5. $\sum_{i=1}^{p} \mu(h_i) = \alpha$ và $\sum_{i=p+1}^{p+q} \mu(h_i) = \beta$, với $\alpha, \beta > 0$ và $\alpha + \beta = 1$

Trong ĐSGT, mỗi phần tử $x \in X$ đều mang dấu âm hay dương và được định nghĩa đệ quy như sau:

Định nghĩa 1.11. [27] Cho $\underline{X} = (X, G, H, \leq)$ là một ĐSGT. Hàm *Sign*: X $\rightarrow$ {-1, 0, 1} là một ánh xạ được định nghĩa một cách đệ qui như sau, với $h, h' \in H$, $c \in \{c^+, c^-\}$:

1. $Sign(c^-) = -1$ và $Sign(c^+) = +1$
2. $Sign(h'hx) = -Sign(hx)$ nếu *h'* là *negative* với *h* và $h'hx \neq hx$
3. $Sign(h'hx) = Sign(hx)$ nếu *h'* là *positive* với *h* và $h'hx \neq hx$
4. $Sign(h'hx) = 0$ nếu $h'hx = hx$

Mệnh đề 1.2. [27] Với $\forall x \in X$, ta có: $\forall h \in H$, nếu $Sign(hx) = +1$ thì $hx > x$, nếu $Sign(hx) = -1$ thì $hx < x$ và nếu $Sign(hx) = 0$ thì $hx = x$.

Định nghĩa 1.12. [27] Giả sử cho trước độ đo tính mờ của các gia tử $\mu(h)$, và các giá trị độ đo tính mờ của các phần tử sinh $fm(c^-)$, $fm(c^+)$ và *w* là phần tử trung hòa. Hàm $\nu : X \rightarrow [0,1]$ được gọi là hàm định lượng ngữ nghĩa của *X* được định nghĩa như sau với $x = h_m...h_2h_1c$:

1. Với *x* là phần tử sinh tức là $x = c^+$ hoặc $x = c^-$, lúc này:

$$\nu(c^-) = W - \alpha \times fm(c^-)$$

$$\nu(c^+) = W + \alpha \times fm(c^+)$$

2. Với *x* không phải là phần tử sinh và với $1 \leq j \leq p$:

$$v(h_j x) = v(x) + Sign(h_j x) \times \left[\sum_{i=1}^{j} fm(h_i x) - \omega(h_j x) \times fm(h_j x)\right]$$

ngược lại, tức với $-q \le j \le -1$:

$$v(h_j x) = v(x) + Sign(h_j x) \times \left[\sum_{i=j}^{-1} fm(h_i x) - \omega(h_j x) \times fm(h_j x)\right]$$

trong đó $\omega(h_j x) = \frac{1}{2}\left[1 + Sign(h_j x) \times Sign(h_q h_j x) \times (\beta - \alpha)\right] \in \{\alpha, \beta\}$

Ví dụ 1.3. Cho ĐSGT $\underline{X} = (X, G, H, \le)$, với $G = \{0, nhỏ, W, lớn, 1\}$, $H^+ = \{hơn, rất\}$, $H^- = \{ít, khả\ năng\}$. Giả sử cho $W = 0.5$, $\mu(ít) = 0.4$, $\mu(khả\ năng) = 0.1$, $\mu(hơn) = 0.1$, $\mu(rất) = 4$. Khi đó giá trị định lượng ngữ nghĩa như sau:

1. Với $x = c^+ = lớn$, ta có $v(lớn) = W + \alpha \times fm(lớn) = 0.5 + 0.5 \times 0.5 = 0.75$. Với $x = c^- = nhỏ$, ta có $v(nhỏ) = W - \alpha \times fm(nhỏ) = 0.5 - 0.5 \times 0.5 = 0.25$

2. Với $h_j x = khả\ năng\ lớn$, tức là $j = -1$, $x = lớn$, ta có $Sign(h_j x) = -Sign(lớn) = -1$, $fm(h_{-1}x) = fm(khả\ năng\ lớn) = \mu(khả\ năng) \times fm(lớn) = 0.1 \times 0.5 = 0.05$. Vậy $v(khả\ năng\ lớn) = 0.75 - (0.05 - 0.5 \times 0.05) = 0.725$

3. Với $h_j x$ = rất *khả năng lớn*, tức là $j = p = 2$, $x = khả\ năng\ lớn$, ta có $Sign(h_j x) = -Sign(khả\ năng\ lớn) = 1$, $fm(h_1) = \mu(h_1) \times fm(x) = \mu(hơn) \times \mu(khả\ năng) \times fm(lớn) = 0.1 \times 0.1 \times 0.5 = 0.005$

Tương tự $fm(h_2) = \mu(h_2) \times fm(x) = \mu(rất) \times \mu(khả\ năng) \times fm(lớn) = 0.4 \times 0.1 \times 0.5 = 0.02$

Vậy $v(rất\ khả\ năng\ lớn) = 0.725 + ((0.005 + 0.02) - (0.5 \times 0.02) = 0.74$.

Định nghĩa 1.13. Xét $P^k = \{I(x): x \in X^k\}$ với $X^k = \{x \in X : |x| = k\}$ là một phân hoạch. Ta nói rằng u xấp xỉ v theo mức k trong P^k khi và chỉ khi $I(u)$ và $I(v)$ cùng thuộc một khoảng trong P^k.

Ví dụ 1.4. Cho ĐSGT $\underline{X} = (X, G, H, \le)$, Trong đó $H = H^+ \cup H^-$, $H^+ = \{hơn, rất\}$, $hơn < rất$, $H^- = \{ít, khả\ năng\}$, $ít > khả\ năng$, $G = \{trẻ, già\}$. Ta có $P^1 = \{I(trẻ), I(già)\}$ là một phân hoạch của [0,1]. Tương tự, $P^2 = \{I(hơn\ trẻ), I(rất\ trẻ), I(ít\ trẻ), I(khả\ năng\ trẻ), I(hơn\ già), I(rất\ già), I(ít\ già), I(khả\ năng\ già)\}$ là phân hoạch của [0,1].

Định nghĩa 1.14. Cho ĐSGT $\underline{X} = (X, G, H, \le)$, v là hàm định lượng ngữ nghĩa

của X. $X_k = \{x \in X : |x| = k\}$. $\Phi_k : [0, 1] \to X$ gọi là hàm ngược của hàm ν theo mức k được xác định: $\forall a \in [0, 1]$, $\Phi_k(a) = x^k$ khi và chỉ khi $a \in I(x^k)$, với $x^k \in X_k$.

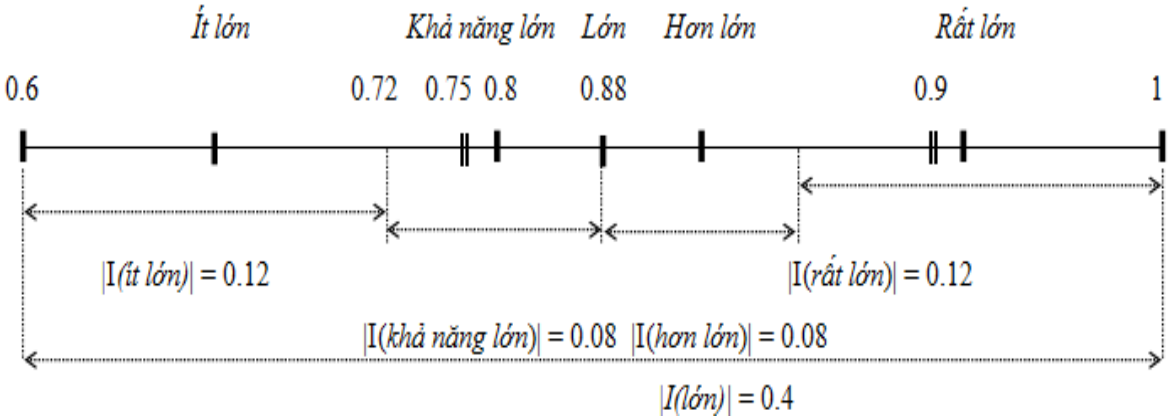
Ví dụ 1.5. Cho ĐSGT $\underline{X} = (X, G, H, \leq)$, trong đó $H^+ = \{hơn, rất\}$ với *hơn* < *rất* và $H^- = \{ít, khả\ năng\}$ với *ít* > *khả năng*, $G = \{nhỏ, lớn\}$. Cho $W = 0.6$, $\mu(hơn) = 0.2$, $\mu(rất) = 0.3$, $\mu(ít) = 0.3$, $\mu(khả\ năng) = 0.2$.

Ta có $P^2 = \{I(hơn\ lớn), I(rất\ lớn), I(ít\ lớn), I(khả\ năng\ lớn), I(hơn\ nhỏ), I(rất\ nhỏ), I(ít\ nhỏ), I(khả\ năng\ nhỏ)\}$ là phân hoạch của [0, 1].

$fm(nhỏ) = 0.6$, $fm(lớn) = 0.4$, $fm(rất\ lớn) = 0.12$, $fm(khả\ năng\ lớn) = 0.08$

Ta có $|I(rất\ lớn)| = fm(rất\ lớn) = 0.12$, hay $I(rất\ lớn) = [0.88, 1]$

Do đó theo định nghĩa $\Phi_2(0.9) = rất\ lớn$ vì $0.9 \in I(rất\ lớn)$, như Hình 1.1.



Hình 1.1. Tính mờ của phần tử sinh *lớn*

Tương tự ta có $|I(khả\ năng\ lớn)| = fm(khả\ năng\ lớn) = 0.08$, hay $I(khả\ năng\ lớn) = [0.72, 0.8]$.

Do đó theo định nghĩa $\Phi_2(0.75) = khả\ năng\ lớn$ vì $0.75 \in I(khả\ năng\ lớn)$.

Định lý 1.3. Cho ĐSGT $\underline{X} = (X, G, H, \leq)$, ν là hàm định lượng ngữ nghĩa của X, Φ_k là hàm ngược của ν, ta có:

1. $\forall x^k \in X^k$, $\Phi_k(\nu(x^k)) = x^k$

2. $\forall a \in I(x^k)$, $\forall b \in I(y^k)$, $x^k \neq_k y^k$, nếu $a < b$ thì $\Phi_k(a) <_k \Phi_k(b)$

Thật vậy:

1. Đặt $a = \nu(x^k) \in [0, 1]$. Vì $\nu(x^k) \in I(x^k)$ nên $a \in I(x^k)$. Theo định nghĩa ta có $\Phi_k(\nu(x^k)) = x^k$.

2. Vì $x^k \neq_k y^k$ nên theo định nghĩa ta có $x^k <_k y^k$ hoặc $y^k <_k x^k$, suy ra $\nu(x^k) < \nu(y^k)$ hoặc $\nu(y^k) < \nu(x^k)$. Mặt khác ta có $\nu(x^k) \in I(x^k)$ và $\nu(y^k) \in I(y^k)$, theo giả thiết $a < b$ do đó $x^k <_k y^k$, tức là $\Phi_k(a) <_k \Phi_k(b)$.

■

1.2.4. Khoảng mờ và các mối tương quan của khoảng mờ

Định nghĩa 1.15. [27] Khoảng mờ $I(x)$ của một phần tử x là một đoạn con của [0, 1], được xác định bằng cách quy nạp theo độ dài của x như sau:

1. Với độ dài x bằng 1($l(x) = 1$), tức là $x \in \{c^+, c^-\}$, $I_{fm}(c^-)$ và $I_{fm}(c^+)$ là các khoảng con và tạo thành một phân hoạch của [0, 1], thỏa $I_{fm}(c^-) \leq I_{fm}(c^+)$. Tức là $\forall u \in I_{fm}(c^-)$ và $\forall u \in I_{fm}(c^+)$: $u \leq v$. Điều này hoàn toàn phù hợp với thứ tự ngữ nghĩa của c^- và c^+.

Ký hiệu độ dài của $I_{fm}(x)$ là $|I_{fm}(x)|$. Ta có $|I_{fm}(c^-)| = I_{fm}(c^-)$ và $|I_{fm}(c^+)| = I_{fm}(c^+)$.

2. Giả sử $\forall x \in X$ độ dài bằng k ($l(x) = k$) có khoảng mờ là $I_{fm}(x)$ và $|I_{fm}(x)| = fm(x)$. Các khoảng mờ của $y = h_i x$, $\forall i \in [-q, -q+1,\ldots,-1, 1, 2,..., p]$, lúc này $l(y) = k + 1$, là tập $\{I_{fm}(h_i x)\}$ thoả mãn một phân hoạch của $I_{fm}(x)$, $|I_{fm}(h_i x)| = I_{fm}(h_i x)$ và có thứ tự tuyến tính tương ứng với thứ tự của tập $\{h_{-q}x, h_{-q+1}x, \ldots, h_p x\}$.

Khi $l(x) = k$, ta ký hiệu $I(x)$ thay cho $I_{fm}(x)$, $X_k = \{\forall x \in X: l(x) = k\}$ là tập các phần tử trong X có độ dài đúng bằng k, $I_k = \{I_k(x) : x \in X_k\}$ là tập tất cả các khoảng mờ mức k.

Định nghĩa 1.16. Hai khoảng mờ được gọi là bằng nhau, ký hiệu $I(x) = I(y)$ khi chúng được xác định bởi cùng một giá trị ($x = y$), tức là ta có $I_L(x) = I_L(y)$ và $I_R(x) = I_R(y)$. Trong đó ký hiệu $I_L(x)$ và $I_R(x)$ là điểm mút trái và phải của khoảng mờ $I(x)$. Ngược lại, ta gọi chúng là hai khoảng mờ khác nhau và ký hiệu là $I(x) \neq I(y)$.

Định lý 1.4. [27] Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$, ta có:

1. Nếu $sign(h_p x) = +1$, thì

$$I(h_{-q}x) \leq I(h_{-q+1}x) \leq \ldots \leq I(h_{-1}x) \leq I(h_1 x) \leq I(h_2 x) \leq \ldots \leq I(h_p x)$$

và nếu $sign(h_p x) = -1$, thì

$$I(h_{-q}x) \leq I(h_{-q+1}x) \geq \ldots \geq I(h_{-1}x) \geq I(h_1 x) \geq I(h_2 x) \geq \ldots \geq I(h_p x)\ .$$

2. Tập $I_k = \{I_k(x) : x \in X_k\}$ là một phân hoạch của đoạn [0, 1].

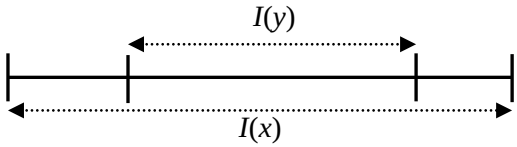
3. Cho một số m, $\{I(y) : y = h_m..h_1x \ \forall \ h_m..h_1 \in H\}$ là một phân hoạch của khoảng mờ I(x).

4. Tập $I_k = \{I_k(x) : x \in X_k\}$ mịn hơn tập $I_{k-1} = \{I_k(x) : x \in X_{k-1}\}$ tức là mọi khoảng trong I_k đều được chứa trong I_{k-1}.

5. Nếu $x < y$ và $l(x) = l(y) = k$ thì $I_k(x) \preceq I_k(y)$ và $I_k(x) \neq I_k(y)$.

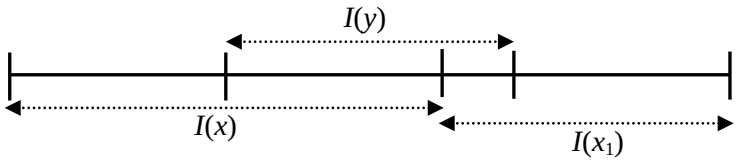
Định nghĩa 1.17. Cho một ĐSGT $\underline{X} = (X, G, H, \preceq)$, với $x, y \in X$ ta có:

1. Nếu $I_L(x) \leq I_L(y)$ và $I_R(x) \geq I_L(y)$ thì ta nói giữa y và x có mối tương quan $I(y) \subseteq I(x)$, ngược lại ta nói $I(y) \not\subset I(x)$.



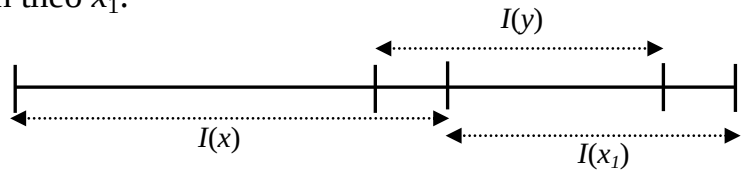
Hình 1.2. Mối tương quan $I(y) \subseteq I(x)$

2. Khi $I(y) \not\subset I(x)$, với $x_1 \in X$ và giả sử $x < x_1$, nếu $|I(y) \cap I(x)| \geq |I(y)|/£$ với £ là số đoạn $I(x_i) \subseteq [0, 1]$ sao cho $I(y) \cap I(x_i) \neq \varnothing$ thì ta nói ta nói y có mối tương quan được đối sánh theo x.



Hình 1.3. Mối tương quan của y được đối sánh theo x, khi $I(y) \not\subset I(x)$

Ngược lại, nếu $|I(y) \cap I(x_1)| \geq |I(y)|/£$ thì ta nói ta nói y có mối tương quan được đối sánh theo x_1.



Hình 1.4. Mối tương quan của y được đối sánh theo x_1, khi $I(y) \not\subset I(x)$

1.3. Phân lớp dữ liệu bằng cây quyết định

1.3.1. Bài toán phân lớp trong khai phá dữ liệu

Mục đích của khai phá dữ liệu nhằm phát hiện các tri thức mà mỗi tri thức được khai phá đó sẽ được mô tả bằng các mẫu dữ liệu. Sự phân lớp là quá trình

quan trọng trong khai phá dữ liệu, nó chính là việc đi tìm những đặc tính của đối tượng, nhằm mô tả một cách rõ ràng phạm trù mà các đối tượng đó thuộc về một lớp nào đó [44], [74]. Quá trình phân lớp gồm có 02 tiến trình:

1. *Xây dựng mô hình:* với tập các lớp đã được định nghĩa trước, mỗi bộ mẫu phải được quyết định để thừa nhận vào một nhãn lớp. Tập các bộ dùng cho việc xây dựng mô hình gọi là tập dữ liệu huấn luyện, tập huấn luyện có thể được lấy ngẫu nhiên từ các cơ sở dữ liệu nghiệp vụ được lưu trữ.

2. *Sử dụng mô hình:* ước lượng độ chính xác của mô hình. Dùng một tập dữ liệu kiểm tra có nhãn lớp được xác định hoàn toàn độc lập với tập dữ liệu huấn luyện để đánh giá độ chính xác của mô hình. Khi độ chính xác của mô hình được chấp nhận, ta sẽ dùng mô hình để phân lớp các bộ hoặc các đối tượng trong tương lai mà nhãn lớp của nó chưa được xác định từ tập dữ liệu chưa biết.

Vậy, bài toán phân lớp có thể được phát biểu tổng quát như sau:

Cho $U = \{A_1, A_2,\ldots, A_m\}$ là tập có m thuộc tính, $Y = \{y_1, ..., y_n\}$ là tập các nhãn của các lớp; với $D = A_1 \times ... \times A_m$ là tích Đề-các của các miền của m thuộc tính tương ứng, có n số lớp và N là số mẫu dữ liệu. Mỗi dữ liệu $d_i \in D$ thuộc một lớp $y_i \in Y$ tương ứng tạo thành từng cặp $(d_i , y_i) \in (D, Y)$.

Cách thức xây dựng mô hình quyết định tính hiệu quả của mô hình thu được. Nhiều tác giả đã nghiên cứu về lý thuyết nhằm xây dựng mô hình và triển khai ứng dụng như:

- Abonyi J., Roubos J.A. [6], Alberto Fernández, María Calderón [8], Fernandez A., Calderon M., Barrenechea E. [22] với hệ luật và hệ luật mờ;

- José A. Sanz, Alberto F., Humberto B. [40] với hệ luận ngôn ngữ mờ;

- Adler D. [7], Hou Yuan-long, Chen Ji-lin, Xing Zong-yi [33], Ishibuchi H., Nojima Y., Kuwajima I. [36] với giải pháp di truyền truyền học;

- Fuller R [23], Lee C. S. George, Lin C. T [45], Zahra Mirzamomen, Mohammadreza K. [82] với phương pháp mạng nơ-ron và mạng nơ-ron mờ;

- Shou-Hsiung Cheng [68], Ziarko W. [88] với lý thuyết tập thô;

- Prade H., Djouadi Y., Alouane B [59], Rolly Intan, Oviliani Yenty Yuliana, Andreas Handojo [65] với phương pháp phân cụm và luật kết hợp,…

Trong các phương pháp đã được nghiên cứu, mô hình cây quyết định là một trong những giải pháp trực quan và hữu hiệu để mô tả quá trình khai phá dữ liệu nên nó được coi là công cụ mạnh, hữu ích và phổ dụng Kishor Kumar Reddy, Vijaya Babu [43], Mariana V. Ribeiro, Luiz Manoel S. Cunha, Heloisa A. Camargo [50], Quinlan J. R. [60], [61], Yakun Hu, Dapeng Wu, Antonio Nucci [77],…

1.3.2. Cây quyết định

Một cây quyết định là một mô hình logic được biểu diễn như một cây, cho biết giá trị của một biến mục tiêu có thể được dự đoán bằng cách dùng các giá trị của một tập các biến dự đoán. Trên mô hình cây quyết định, mỗi một nút trong tương ứng với một biến dự đoán, đường nối giữa nó với nút con của nó thể hiện một giá trị cụ thể cho biến đó. Mỗi nút lá đại diện cho giá trị dự đoán của biến mục tiêu, được biểu diễn bởi đường đi từ nút gốc tới nút lá đó. Nó có thể hiểu như là một cách biểu diễn các quy tắc để đưa về kết quả là một giá trị cụ thể hay thuộc một lớp nào đó.

Giải bài toán phân lớp dựa trên mô hình cây quyết định chính là xây dựng một cây quyết định, ký hiệu *S*, để phân lớp. S đóng vai trò như một ánh xạ từ tập dữ liệu vào tập nhãn:

$$S : D \rightarrow Y \qquad (1.4)$$

Cây quyết định biểu diễn cho tri thức về bài toán, nó không chỉ phản ánh đúng với tập dữ liệu mẫu huấn luyện mà còn phải có khả năng dự đoán và cung cấp giúp cho người dùng phán đoán, ra quyết định đối với đối tượng trong tương lai mà nhãn lớp của nó chưa được xác định từ tập dữ liệu chưa biết. Quá trình học cây quyết định gồm có 3 giai đoạn:

1. Tạo cây. Sử dụng các thuật toán phân lớp để phân chia tập dữ liệu huấn luyện một cách đệ quy cho đến khi mọi nút lá đều thuần khiết, tức là nút mà tại đó tập mẫu tương ứng có cùng một giá trị trên thuộc tính quyết định Y. Sự lựa chọn các thuộc tính trong quá trình xây dựng cây được dựa trên việc đánh giá lượng lợi ích thông tin tại mỗi thuộc tính đang xét.

2. Cắt tỉa cây. Sau khi tạo cây, cắt tỉa cây quyết định là việc làm rất cần thiết để khắc phục những khiếm khuyết của cây. Cắt tỉa cây là cố gắng loại bỏ những nhánh không phù hợp hay những nhánh gây ra lỗi.

3. Kiểm định cây kết quả. Để bảo đảm độ chính xác của cây trước khi đưa vào ứng dụng trong thực tế, ta cần phải đánh giá độ chính xác của cây từ đó đưa ra tiêu chí đánh giá độ tin cậy theo tỷ lệ phần trăm được dự đoán chính xác.

Việc tạo cây là giai đoạn quan trọng nhất, nó chính là quá trình tạo ra mô hình logic cho cây. Để xây dựng cây quyết định, tại mỗi nút trong cần xác định một thuộc tính thích hợp để kiểm tra, phân chia dữ liệu thành các tập con.

Cho tập mẫu huấn luyện D gồm có m thuộc tính, n bộ. Mỗi thuộc tính bất kỳ $A_i \in D$, ta ký hiệu $|A_i|$ là số các giá trị khác nhau của nó và gọi là lực lượng của A_i. Số lần xuất hiện mỗi một giá trị a_{i_j} trong A_i ký hiệu là $|a_{i_j}|$. Với thuộc tính quyết định Y, số lớp cần phân hoạch trong Y chính là lực lượng của Y và ta viết $|Y|$. Như vậy khi $|Y| = 1$ thì tất cả các đối tượng trong tập mẫu thuộc cùng một lớp và ta nói chúng là thuần nhất trên Y.

Trên mỗi tập mẫu huấn luyện, về cơ bản các thuật toán phân lớp dữ liệu bằng cây quyết định phải thực hiện 2 bước sau [53], [60]:

Bước 1: Chọn thuộc tính A_i có các giá trị $a_{i_1}, a_{i_2}, \dots, a_{i_n}$

Bước 2: Với thuộc tính A_i được chọn, ta tạo một nút của cây và sau đó chia tập mẫu này thành k tập mẫu $D_1, D_2, \dots, D_k$ tương ứng với k nút được tạo và sau đó lại tiếp tục.

Bước 2 là bước phân chia với kết quả nhận được từ *Bước 1*, điều này có nghĩa là chất lượng của cây kết quả phụ thuộc phần lớn vào cách chọn thuộc tính và cách phân chia tập mẫu tại mỗi nút. Chính vì điều này, các thuật toán đều phải tính lợi ích thông tin nhận được trên các thuộc tính và chọn thuộc tính tương ứng có lợi ích thông tin tốt nhất để làm nút phân tách trên cây, nhằm để đạt được cây có ít nút nhưng có khả năng dự đoán cao.

1.3.3. Lợi ích thông tin và tỷ lệ lợi ích thông tin

a. Entropy

Một bít là một chữ số nhị phân nên ta sử dụng một bít để đại diện cho đối tượng thì ta chỉ phân biệt được hai đối tượng, với n bít sẽ phân biệt được 2^n đối tượng khác nhau. Theo đó chúng ta có thể phân biệt n đối tượng bằng $log_2(n)$ bít.

Một bộ mã P thiết kế để phân biệt các phần tử của tập $\{x\}$, để nhận diện

được $\{x\}$, chúng ta cần $-log_2P(x)$ bít. Nếu muốn xác định một phân phối thì ít nhất ta cần phải dùng số bít kỳ vọng để nhận diện một phần tử là:

$$\sum_x P(x)\ log\ P(x) \tag{1.5}$$

gọi là nội dung thông tin hay *Entropy* của một phân phối [54], [60].

b. Lợi ích thông tin

Lợi ích thông tin được tính theo *Entropy*, nó đại diện cho giá trị thông tin của thuộc tính được chọn trong tập mẫu. Với thuộc tính quyết định Y của tập D chưa thuần nhất, được phân phối trong n lớp và giả sử tỉ lệ của các lớp của Y trong D là $p_1, p_2, \ldots p_n$. Khi đó, *Entropy* của Y trong D là:

$$E(Y,D)\ =\ \sum_{i=1}^{n} -p_i \log_2 p_i \tag{1.6}$$

Giả sử thuộc tính $A_i \in D$ có m giá trị được chọn làm thuộc tính phân lớp và giả thiết A_i sẽ chia tập huấn luyện D thành m tập con $D_1, D_2, \ldots D_n$,. Lúc này, *Entropy* mà ta nhận được khi phân lớp trên thuộc tính A_i là:

$$E(A_i,D)\ =\sum_{j=1}^{m} \frac{|D_j|}{|D|} Entropy\ (D_j) \tag{1.7}$$

Lợi ích thông tin của thuộc tính A_i trong D được tính [54], [60]:

$$Gain(A_i, D) = E(Y, D) - E(A_i, D) \tag{1.8}$$

c. Tỷ lệ lợi ích thông tin

Với cách tính ở trên, khi thuộc tính A_i có giá trị liên tục với số lượng phần tử lớn, khi đó, mỗi giá trị sẽ là một lớp, $E(A_i, D) = 0$ và lợi nhuận thông tin $Gain(A_i, D) = E(Y, D)$. Do đó, ta tính tỉ lệ lợi nhuận thông tin bằng cách sử dụng thêm hệ số phân chia.

Giả sử thuộc tính A_i trong tập D có k giá trị, được làm k tập $D_1, D_2, \ldots, D_k$. Hệ số phân chia của thuộc tính A_i trong tập D ký hiệu là $SplitInfo(A_i, D)$ được cho bởi công thức (1.9).

$$SplitInfo(A_i,\ D)\ =\ -\sum_{j=1}^{k} \frac{|D_j|}{|D|}\ Log_2 \frac{|D_j|}{|D|} \tag{1.9}$$

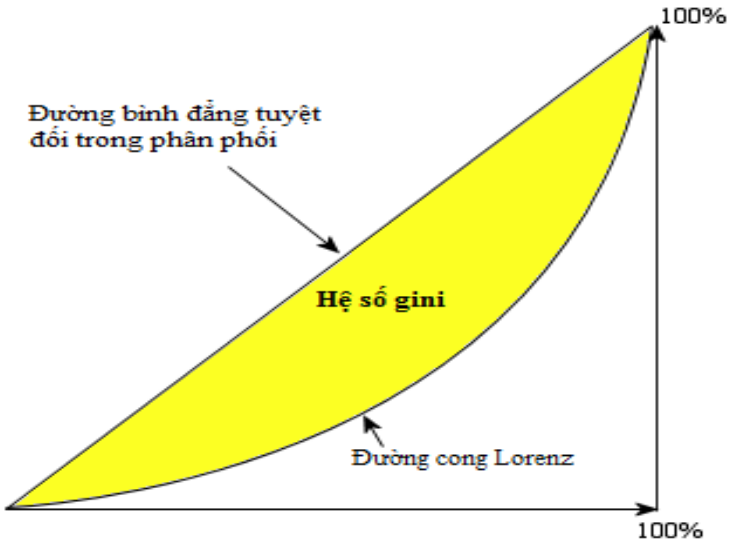
Như vậy, tỷ lệ lợi ích thông tin của thuộc tính A_i là:

$$GainRatio(A_i,\ D)\ =\ \frac{Gain\ (A_i,\ D)}{SplitInfo\ (A_i,\ D)} \tag{1.10}$$

Trên cơ sở tính toán tỷ lệ lợi ích thông tin cho các thuộc tính trong D, thuộc tính nào có tỷ lệ lợi ích thông tin lớn nhất được chọn để phân lớp [54], [60].

d. Hệ số *Gini* và tỷ lệ hệ số *Gini*

Hệ số *Gini* là tỷ lệ phần trăm giữa diện tích của vùng nằm giữa đường bình đẳng tuyệt đối và đường cong Lorenz với diện tích của vùng nằm giữa đường bình đẳng tuyệt đối và đường bất bình đẳng tuyệt đối. Hệ số *Gini* được đưa ra dựa vào hàm phân bố xác xuất, nó dựa trên việc tính bình phương các xác suất thành viên cho mỗi thể loại đích trong nút.



Hình 1.5. Minh họa hình học về chỉ số Gini

Giả sử tập *D* được chia làm *n* lớp khác nhau, tần suất xuất hiện của lớp *i* trong *D* là p_i, chỉ số *Gini* của tập *D* được ký hiệu là *Gini*(*D*), được cho bởi công thức (1.11) [47], [48].

$$\text{Gini}(D) = 1 - \sum_{i=1}^{n} p_i^2 \qquad (1.11)$$

Nếu tập *D* được tách thành 2 tập con D_1, D_2 thì hệ số *Gini* của tập D khi được chia tách được gọi là tỷ lệ hệ số Gini (*GiniSplitIndex)* ký hiệu là $Gini(D)_{Split}$ được xác định như công thức (1.12).

$$Gini(D)_{Split} = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \qquad (1.12)$$

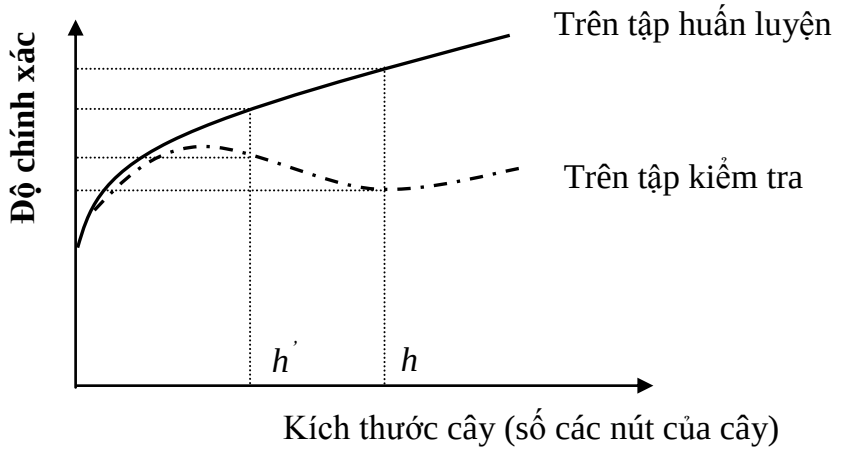
1.3.4. Vấn đề quá khớp trong mô hình cây quyết định

Trong quá trình học cây quyết định, mỗi nhánh của cây vừa đủ sâu để phân lớp hoàn hảo các mẫu huấn luyện, điều này chính là chiến lược phù hợp. Song trong thực tế nó có thể dẫn đến nhiều khó khăn khi có độ nhiễu của dữ liệu

huấn luyện hoặc số mẫu huấn luyện là quá nhỏ để đem lại một mô hình quá lý tưởng [38], [41], [54], [64], .

Trong qui nạp, việc khái quát hoá không thể không có thiên lệch qui nạp (Inductive Bias), đó là việc chọn một số tiêu chuẩn để ràng buộc cho một lớp khái niệm nào đó. Và cũng nói thêm rằng, việc học tập mà không có thiên lệch sẽ dẫn đến một tập rất lớn các lớp khái niệm cần tính. Vì vậy, đôi lúc các mẫu dữ liệu cho ta một khái niệm trong quá trình học nhưng điều này chưa hẳn là có thể dự đoán tốt đối với các mẫu chưa gặp. Hơn thế nữa, khi số lượng các mẫu của tập huấn luyện tăng lên thì cũng không bảo đảm được rằng chương trình học sẽ hội tụ đến khả năng đúng khi dự đoán, ta gọi là "*quá khớp*" trong quá trình huấn luyện. Trong thực tế, khó có câu trả lời cho câu hỏi: "cần bao nhiêu mẫu để nhận ra một khái niệm đúng". Như vậy, "*quá khớp*" là một vấn đề khó khăn đáng kể trên thực tế đối với việc học phân lớp bằng cây quyết định [54].

Định nghĩa 1.18. Cho một giả thiết *h* ứng với mô hình của một cây quyết định, ta nói nó là "*quá khớp"* với tập dữ liệu huấn luyện, nếu tồn tại một giả thiết h' với *h* có sai số nhỏ hơn tức độ chính xác lớn hơn h' trên tập dữ liệu huấn luyện, nhưng h' có sai số nhỏ hơn *h* trên tập dữ liệu kiểm tra, minh họa ở Hình 1.6.



Hình 1.6. Vấn đề "*quá khớp*" trong cây quyết định

Định nghĩa 1.19. Một cây quyết định được gọi là cây dàn trải nếu tồn tại nút có số nhánh phân chia tại đó lớn hơn chiều cao của cây.

Khi một cây quyết định được xây dựng dựa trên tập mẫu huấn luyện xảy ra tình trạng quá khớp, có khả năng xuất hiện dàn trải do ta có quá ít thông tin, bị nhiễu hoặc các ngoại lệ của dữ liệu. Lúc này, cây kết quả không phản ánh ý nghĩa

thực tiễn của mô hình được huấn luyện và chúng ta phải tiến hành cắt tỉa cây để thu gọn mô hình, Jothikumar R., Siva Balan R. V., [41]; Patil N., R.C. Barros và các cộng sự [54], [64],...

1.4. Phân lớp dữ liệu bằng cây quyết định mờ

1.4.1. Các hạn chế của phân lớp dữ liệu bằng cây quyết định rõ

Như chúng ta đã biết, do tính hữu hiệu của mô hình cây quyết định cho bài toán phân lớp dữ liệu nên hiện đã có nhiều nghiên cứu cho vấn đề này. Mục tiêu của cách tiếp cận này là dựa vào tập huấn luyện với các miền dữ liệu được xác định cụ thể (giá trị của mỗi thuộc tính trong tập mẫu là giá trị xác định liên tục hay rời rạc), chúng ta xây dựng một phương pháp học cây quyết định với sự phân chia rõ ràng theo các ngưỡng giá trị tại các nút phân chia. Các kết quả nổi bật như: CART [14], [24], [43], [74]; ID3, C45, C50 [54], [60-62], [67], [78]; SLIQ [47], [52] và SPRINT [48], [87],...

♦ **Hướng tiếp cận dựa vào việc tính lợi ích thông tin của thuộc tính:** Breiman L, Friedman J. [14], Guang-Bin Huang, Hongming Zhou [24], Kishor Kumar Reddy [43], Patil N. [54], Quinlan J. R. [60-62], Shou-Hsiung Cheng, Yi Yang [67], [78] và các cộng sự,… đã dựa vào khái niệm Entropy thông tin để tính lợi ích thông tin và tỷ lệ lợi ích thông tin của các thuộc tính tại thời điểm phân chia của tập mẫu huấn luyện, từ đó lựa chọn thuộc tính tương ứng có lợi ích thông tin lớn nhất làm điểm phân chia. Sau khi chọn được thuộc tính để phân lớp, nếu thuộc tính là kiểu rời rạc thì phân lớp theo giá trị phân biệt của chúng, nếu thuộc tính là liên tục thì ta phải tìm ngưỡng của phép tách để chia thành 2 tập con theo ngưỡng đó. Việc tìm ngưỡng cho phép tách cũng dựa theo tỷ lệ lợi ích thông tin của các ngưỡng trong tập huấn luyện tại nút đó. Với m là số thuộc tính, n là số thể hiện của tập huấn luyện thì độ phức tạp của các thuật toán là $O(m \times n \times \log n)$.

Tuy hướng tiếp cận này cho chúng ta các thuật toán có độ phức tạp thấp nhưng việc phân chia *k-phân* trên các thuộc tính rời rạc làm cho số nút của cây tại một cấp tăng lên nhanh, làm tăng chiều rộng của cây, dẫn đến việc cây dàn trải theo chiều ngang nên dễ xảy ra tình trạng quá khớp, khó để có thể dự đoán. Hơn nữa, cách chia này có khả năng dẫn đến lỗi - khi dữ liệu không thể đoán nhận được lớp - điều này dẫn đến việc dự đoán sẽ cho kết quả không chính xác.

♦ **Hướng tiếp cận dựa vào việc tính hệ số Gini của thuộc tính:** Manish Mehta, Jorma Rissanen, Rakesh Agrawal, Narasimha Prasad, Mannava Munirathnam Naidu, Zhihao Wang, Junfang Wang, Yonghua Huo, Hongze Qiu, Haitang Zhang và các cộng sự [32], [47], [48], [52], [87] dựa vào việc tính hệ số Gini và tỷ lệ hệ số Gini của các thuộc tính để lựa chọn điểm phân chia cho tập huấn luyện tại mỗi thời điểm. Theo cách tiếp cận này, chúng ta không cần đánh giá mỗi thuộc tính mà chỉ cần tìm điểm tách tốt nhất cho mỗi thuộc tính đó. Thêm vào đó, với việc sử dụng kỹ thuật tiền xử lý sắp xếp trước trên mỗi một thuộc tính, nên hướng tiếp cận này đã giải quyết được vấn đề thiếu bộ nhớ khi tập huấn luyện lớn.

Tuy nhiên, vì tại thời điểm phân chia với thuộc tính rời rạc, hoặc luôn lựa chọn cách phân chia theo *nhị phân tập hợp* của SLIQ (Manish Mehta, Jorma Rissanen, Narasimha Prasad, Mannava Munirathnam Naidu và các cộng sự [47], [52]) hoặc *nhị phân theo giá trị* của SPRINT (Manish Mehta, Jorma Rissanen, Zhihao Wang, Junfang Wang, Yonghua Huo và các cộng sự [48], [87]) nên cây kết quả mất cân xứng vì phát triển nhanh theo chiều sâu. Thêm vào đó, tại mỗi thời điểm chúng ta phải tính một số lượng lớn hệ số Gini cho các giá trị rời rạc nên chi phí về độ phức tạp tính toán cao.

Thêm vào đó, việc học phân lớp bằng cây quyết định theo các hướng tiếp cận đòi hỏi tập mẫu huấn luyện phải thuần nhất và chỉ chứa các dữ liệu kinh điển. Tuy nhiên, do bản chất luôn tồn tại các khái niệm mờ trong thế giới thực nên điều kiện này không đảm bảo trong các cơ sở dữ liệu hiên đại. Vì vậy, việc nghiên cứu bài toán phân lớp dữ liệu bằng cây quyết định mờ là vấn đề tất yếu.

1.4.2. Bài toán phân lớp dữ liệu bằng cây quyết định mờ

Như đã trình bày, cho $U = \{A_1, A_2,\ldots, A_m\}$ là tập có m thuộc tính, $Y = \{y_1, ..., y_n\}$ là tập các nhãn của các lớp; với $D = A_1 \times ... \times A_m$ là tích Đề-các của các miền của m thuộc tính tương ứng, có n số lớp và N là số mẫu dữ liệu. Mỗi dữ liệu $d_i \in D$ thuộc một lớp $y_i \in Y$ tương ứng tạo thành từng cặp $(d_i , y_i) \in (D, Y)$. Ta có bài toán phân lớp dữ liệu bằng cây quyết định là một ánh xạ từ tập dữ liệu vào tập nhãn:

$$S : D \rightarrow Y \qquad (1.4)$$

Trong thực tế, chúng ta có rất nhiều kho dữ liệu nghiệp vụ được lưu trữ

mờ nên cách tiếp cận phân lớp dữ liệu bằng cây quyết định rõ không thể giải quyết các yêu cầu của bài toán. Với mỗi thuộc tính A_i của tập mẫu huấn luyện được gắn với một miền trị thuộc tính, ký hiệu là $Dom(A_i)$, trong có một số thuộc tính cho phép nhận các giá trị ngôn ngữ trong lưu trữ hay trong các câu truy vấn và được gọi là thuộc tính mờ. Các thuộc tính còn lại được gọi là thuộc tính kinh điển. Với sự xuất hiện của các thuộc tính chứa giá trị ngôn ngữ, tức $\exists A_i \in D$ có miền trị $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, với D_{A_i} là tập các giá trị kinh điển của A_i và LD_{A_i} là tập các giá trị ngôn ngữ của A_i. Lúc này, bài toán phân lớp dữ liệu bằng cây quyết định $S : D \rightarrow Y$ tại (1.4) là bài toán phân lớp dữ liệu bằng cây quyết định mờ.

Như vậy, mô hình cây quyết định S phải đạt các mục tiêu như hiệu quả phân lớp cao, tức là sai số phân lớp cho các dữ liệu dự đoán ít nhất có thể và cây có ít nút để thuận tiện cho việc biểu diễn và duyệt cây. Mục tiêu về hiệu quả phân lớp nhằm đáp ứng tính đúng đắn của của mô hình đối với tập dữ liệu mẫu được cho của bài toán, còn mục tiêu sau với mong muốn mô hình cây quyết định nhận được phải đơn giản đối với người dùng.

Ta ký hiệu $\underline{S}$ là tập tất cả các cây có thể được tạo ra từ tập huấn luyện S trên thuộc tính quyết định Y. Gọi $f_h(S) : \underline{S} \rightarrow \mathbb{R}$ là hàm đánh giá khả năng dự đoán của cây quyết định S và $f_n(S) : \underline{S} \rightarrow \mathbb{N}$ là hàm thể hiện số nút của cây kết quả nhằm đánh giá tính đơn giản của cây đối với người dùng. Lúc này, mục tiêu của bài toán phân lớp dữ liệu bằng cây quyết định mờ:

$$S : D \rightarrow Y$$

nhằm đạt được:

$$f_h(S) \rightarrow max \text{ và } f_n(S) \rightarrow min \qquad (1.13)$$

Hai mục tiêu trên khó có thể đạt được đồng thời. Khi số nút của cây giảm đồng nghĩa với lượng tri thức về bài toán giảm nên nguy cơ phân lớp sai sẽ tăng lên, nhưng khi có quá nhiều nút cũng có thể gây ra sự quá khớp thông tin trong quá trình phân lớp.

Bên cạnh đó, sự phân chia tại mỗi nút ảnh hưởng đến tính phổ quát hay cá thể tại nút đó. Nếu sự phân chia tại một nút là nhỏ sẽ làm tăng tính phổ quát và ngược lại nếu sự phân chia lớn sẽ làm tăng tính cá thể của nút đó. Tính phổ quát

của nút trên cây sẽ làm tăng khả năng dự đoán nhưng nguy cơ gây sai số lớn, trong khi tính cá thể giảm khả năng dự đoán nhưng lại tăng tính đúng đắn nhưng nó cũng là nguyên nhân của tình trạng quá khớp trên cây.

Các phương pháp giải quyết bài toán mô hình cây quyết định đều phải thỏa hiệp giữa các mục tiêu này để đạt được kết quả cuối cùng.

1.4.3. Một số vấn đề của bài toán phân lớp dữ liệu bằng cây quyết định mờ

Quá trình xây dựng một mô hình cây quyết định mờ từ tập huấn luyện mờ đã được nhiều nhà khoa học nghiên cứu với nhiều hướng tiếp cận khác nhau. Tuy nhiên, chúng ta có thể tổng hợp quá trình học gồm hai bước:

1. Phân hoạch mờ trên miền của các thuộc tính mờ bằng tập các giá trị ngôn ngữ của các biến ngôn ngữ, mỗi giá trị ngôn ngữ được gán một hàm thuộc tương ứng.

2. Xác định cách phân chia mờ tại các nút tương ứng với thuộc tính mờ để tạo cây quyết định mờ.

Tùy thuộc vào mục đích của mô hình cây quyết định mờ, hiện có nhiều phương pháp học khả quan đã được nghiên cứu và công bố [9-13], [16], [19], [26], [32], [35], [40], [49], [51], [55], [56], [69], [73], [83-86] và chúng ta có thể tổng hợp theo các cách tiếp cận sau:

♦ **Hướng tiếp cận dựa vào lý thuyết tập mờ:** Các nhà khoa học theo cách tiếp cận này đã đưa ra nhiều giải pháp kết hợp khác nhau dựa trên nền tảng của lý thuyết tập mờ. Các nghiên cứu của A. K. Bikas, E. M. Voumvoulakis, Bhatt R. B., [9], [12] và các cộng sự chỉ ra hướng tiếp cận với sự kết hợp của mạng nơ-ron; James F. Smith, N. H. T. Vu [37] với giải thuật di truyền; Moustakidis S., Mallinis G., Koutsias N. [46] với phương pháp máy véc-tơ hỗ trợ; hay cải tiến từ các cách tiếp cận học cây quyết định rõ thông qua lý thuyết tập mờ để tính lợi ích thông tin cho các thuộc tính mờ từ các nghiên cứu của B. Chandra [11], Chida A. [16], Daveedu Raju Adidela, Jaya Suma. G, Lavanya Devi. G [19], Hesham A. Hefny, Ahmed S. Ghiduk [26], Hou Yuan-long, Chen Ji-lin, Xing Zong-yi [32], Marcos E. Cintra, Maria C. Monard [49], Zeinalkhani M., Eftekhari M. [83] và các cộng sự,... với các thuật toán nổi bậc như: Fuzzy ID3, Fuzzy SLIQ, Fuzzy HSM.

Hướng tiếp cận này đã giải quyết được các giá trị mờ trong tập huấn luyện thông qua việc xác định các hàm thuộc, từ đó các bộ giá trị này có thể tham gia vào quá trình huấn luyện nên đã giải quyết được hạn chế của cách tiếp phân lớp rõ là bỏ qua các giá trị dữ liệu mờ trong huấn luyện. Tuy vậy, hiện vẫn còn gặp phải những hạn chế xuất phát từ bản thân nội tại của lý thuyết tập mờ:

- Rất khó để mô phỏng hoàn chỉnh cấu trúc ngôn ngữ mà con người sử dụng để suy luận. Cấu trúc thứ tự cảm sinh trên các khái niệm mờ biểu thị bằng các giá trị ngôn ngữ không được thể hiện trên các tập mờ vì hàm thuộc của chúng lại không sánh được với nhau.

- Trong quá trình lập luận, nhiều khi ta cần phải xấp xỉ ngôn ngữ tức là phải tìm một giá trị ngôn ngữ mà ý nghĩa của nó xấp xỉ với một tập mờ cho trước, điều này gây nên sự phức tạp và sai số lớn cho quá trình xấp xỉ và phụ thuộc rất lớn vào sự chủ quan.

- Một hệ suy diễn xây dựng trên một ngôn ngữ hình thức đều xác định trên tập các lớp công thức, tương đương một cấu trúc đại số thuộc lớp các đại số trừu tượng; trong khi lôgíc mờ, giá trị ngôn ngữ còn thiếu một cơ sở đại số làm nền tảng.

♦ **Hướng tiếp cận xây dựng cây quyết định ngôn ngữ**: Suzan Kantarci-Savas, Efendi Nasibov, Zengchang Qin, Jonathan Lawry, Yongchuan Tang,... [69], [84], [85] và các cộng sự đã xác định các giá trị ngôn ngữ cho tập dữ liệu mờ và xây dựng cây quyết định ngôn ngữ (Linguistic Decision Tree - LDT) bằng cách sử dụng tư tưởng của thuật toán ID3 của cây quyết định rõ cho các nút ứng với các thuộc tính ngôn ngữ (LID3) với các kết quả nổi bật của các thuật toán như: LID3 Uniform, LID3 Entropy, LID3 Percentile... Việc xây dựng các nhãn ngôn ngữ cho các giá trị mờ dựa vào xác suất của các nhãn liên kết trong khi vẫn giữ được các giá trị rõ đã biết, hướng tiếp cận này đã làm giảm sai số đáng kể cho quá trình huấn luyện.

Tuy vậy, hướng tiếp cận này làm này sẽ làm phát sinh cây đa phân do có sự phân chia lớn theo chiều ngang tại các nút ngôn ngữ khi tập giá trị ngôn ngữ của thuộc tính mờ lớn (Hình 1.7). Thêm vào đó, tại nút này, ta không thể sử dụng cách phân chia nhị phân của thuật toán C45/C50 (thuật toán hiện là hữu hiệu nhất cho quá trình học cây quyết định) vì không có thứ tự giữa các giá trị

ngôn ngữ. Do vậy dễ dẫn đến tình trạng quá khớp trên cây kết quả nhận được sau quá trình huấn luyện.

Hình 1.7. Điểm phân chia đa phân theo giá trị ngôn ngữ tại thuộc tính mờ

♦ **Hướng tiếp cận dựa vào ĐSGT:** Bài toán phân lớp dữ liệu mờ nói chung và phân lớp dữ liệu bằng cây quyết định mờ nói riêng, khi tập mẫu liệu huấn luyện có thuộc tính không thuần nhất tức thuộc tính chứa cả dữ liệu rõ và mờ thì bài toán trở nên phức tạp và khó giải quyết. ĐSGT do N. C. Ho & W. Wechler khởi xướng từ 1990 [29] có nhiều ưu điểm. Theo cách tiếp cận này, mỗi giá trị ngôn ngữ của một biến ngôn ngữ nằm trong một cấu trúc đại số nên ta có thể đối sánh giữa các giá trị ngôn ngữ nên đã giải quyết được vấn đề khó khăn của các hướng tiếp cận trước.

Theo hướng tiếp cận này, các nghiên cứu của N. C. Ho, N. C. Hao, L. A. Phuong, L. X. Viet, L. X. Vinh, Long N. V., Lan V. N. [1-5], [27], [28], [29], [30], [31] và các cộng sự đã chỉ ra phương pháp định lượng ngữ nghĩa theo điểm nhằm thuần nhất dữ liệu về các giá trị số hay giá trị ngôn ngữ và cách thức truy vấn dữ liệu trên thuộc tính này. Bài toán xây dựng cây quyết định mờ lúc này có thể sử dụng các thuật toán học theo cách tiếp cận cây quyết định rõ với các nút phân chia nhị phân được tính theo dựa theo điểm phân chia với các giá trị ngôn ngữ đã có thứ tự và hoàn toàn xác định tương ứng với một giá trị số trong ĐSGT đã xây dựng.

Tuy vậy, hướng tiếp cận dựa trên phương pháp định lượng ngữ nghĩa theo điểm làm nền tảng vẫn còn một số vấn đề như:

- Sử dụng khái niệm độ đo tính mờ các giá trị ngôn ngữ để định nghĩa khoảng tính mờ và biểu diễn cho một miền dữ liệu ta thường chỉ áp dụng ở một mức tức là cho các giá trị ngôn ngữ có số lượng gia tử giống nhau, nên sẽ bỏ qua các giá trị ngôn ngữ khác mức có số lượng gia tử ít hơn, hay thậm chí không có gia tử. Điều này rất không phù hợp, bởi các giá trị ngôn ngữ có vai trò bình đẳng trong việc biểu diễn ngữ nghĩa cho một miền dữ liệu nào đó. Vì vậy vẫn còn sai

số trong cây kết quả vì một đoạn con các giá trị rõ của tập huấn luyện sẽ được quy về một điểm tức là một giá trị ngôn ngữ tương ứng, điều này cũng làm xuất hiện các giá trị gần nhau có thể được phân hoạch ở hai đoạn con khác nhau nên kết quả phân lớp khác nhau.

- Cây kết quả thu được theo hướng tiếp cận này trong nhiều trường hợp khó đưa ra dự đoán khi có sự đan xen ở điểm phân chia mờ. Ví dụ ta cần dự đoán cho trường hợp đoạn con $[x_1, x_2]$, với $x_1 < x$ và $x_2 > x$ tại nút phân chia mờ trên cây, Hình 1.8.

Hình 1.8. Điểm phân chia nhị phân theo giá trị ngôn ngữ hoặc giá trị số tại thuộc tính mờ, dựa trên phương pháp định lượng ngữ nghĩa theo điểm trong ĐSGT.

- Việc sử dụng ĐSGT để định lượng cho các giá trị ngôn ngữ trong tập mẫu huấn luyện theo cách tiếp cận này phải dựa vào miền giá trị rõ của chính thuộc tính đang xét đó. Do vậy, ta phải tìm thấy miền trị $[\psi_{min}, \psi_{max}]$ từ miền giá trị rõ của thuộc tính để từ đó sẽ định lượng cho các giá trị ngôn ngữ từ miền trị này. Tuy vậy, việc tìm miền trị $[\psi_{min}, \psi_{max}]$ không phải lúc nào cũng thuận lợi vì có thể xuất hiện các giá trị ngôn ngữ mà giá trị thật sự của nó nằm ngoài miền dữ liệu rõ đang có trong thuộc tính đang xét.

Hiện nay, học phân lớp dữ liệu bằng cây quyết định là một vấn đề quan trọng của bài toán phân lớp trong lĩnh vực khai phá dữ liệu. Việc xây dựng một giải pháp học nhằm thu được cây quyết định hiệu quả để đáp ứng yêu cầu người dùng là một thách thức lớn. Các hướng tiếp cận nhằm mục đích xây dựng mô hình cây quyết định hiệu quả dựa trên tập huấn luyện hiện vẫn còn gặp các khó khăn cần khắc phục. Để giải quyết vấn đề này, luận án tập trung nghiên cứu các lý thuyết về tính chất và đặc trưng của các giá trị ngôn ngữ của tập huấn luyện dựa trên bản chất của ĐSGT, nghiên cứu các mô hình học bằng cây quyết định và các giải pháp học nhằm xây dựng cây quyết định hiệu quả trong phân lớp và

đơn giản với người dùng.

1.5. Kết luận chương 1

Với mục tiêu nghiên cứu bài toán phân lớp dữ liệu bằng cây quyết định mờ dựa trên ĐSGT, chương này tập trung nghiên cứu, phân tích và đánh giá các vấn đề liên quan mật thiết đến luận án.

Đầu tiên luận án đã trình bày về khái niệm mờ, vấn đề mô hình hóa toán học cho khái niệm mờ chính là các tập mờ và khái niệm biến ngôn ngữ. Tiếp theo là phương pháp lập luận xấp xỉ trực tiếp trên ngôn ngữ, ở phần này những khái niệm và tính chất về ĐSGT lần lượt được nêu ra, đây là những kiến thức cơ sở cần thiết cho việc nghiên cứu các chương tiếp theo của luận án.

Luận án cũng đã trình bày các vấn đề cơ bản của bài toán phân lớp dữ liệu bằng cây quyết định, các hạn chế trên cây quyết định truyền thống và sự cần thiết của bài toán phân lớp dữ liệu bằng cây quyết định mờ. Ở đây, luận án đã phát biểu hình thức bài toán phân lớp dữ liệu bằng cây quyết định và cũng tập trung nghiên cứu, phân tích và đánh giá các công trình nghiên cứu đã công bố gần đây, chỉ ra các vấn đề còn tồn tại để định hướng cho mục tiêu và nội dung cần giải quyết cho luận án.

Chương 2.
PHÂN LỚP DỮ LIỆU BẰNG CÂY QUYẾT ĐỊNH MỜ THEO PHƯƠNG PHÁP ĐỐI SÁNH ĐIỂM MỜ DỰA TRÊN ĐẠI SỐ GIA TỬ

2.1. Giới thiệu

Trong bài toán học phân lớp dữ liệu bằng cây quyết định mờ: $S : D \rightarrow Y$, $Y = \{y_1, ..., y_n\}$ là tập các nhãn của các lớp, $D = A_1 \times ... \times A_m$ là tích Đề-các của các miền của m thuộc tính tương ứng, n là số lớp và N là số mẫu dữ liệu. Với $f_h(S)$ là hàm đánh giá khả năng dự đoán, $f_n(S)$ là hàm cho biết số nút của cây quyết định S, mục tiêu của bài toán phân lớp dữ liệu bằng cây quyết định mờ nhằm đạt được cây có ít nút nhưng có khả năng dự đoán cao, không xảy ra tình trạng quá khớp, tức cần đạt được:

$$f_h(S) \rightarrow max \text{ và } f_n(S) \rightarrow min$$

Như chúng ta đã biết, trên tập mẫu huấn luyện D, về cơ bản, các thuật toán phân lớp dữ liệu bằng cây quyết định phải thực hiện 2 bước:

Bước 1: Chọn thuộc tính có lợi ích thông tin tốt nhất A_i với các giá trị $\{a_{i_1}, a_{i_2}, \dots, a_{i_n}\}$.

Bước 2: Với thuộc tính được chọn A_i, tạo một nút của cây và sau đó chia tập D thành k tập mẫu $D_1, D_2, \dots, D_k$ tương ứng và sau đó lại tiếp tục.

Tuy vậy, tại các bước của quá trình huấn luyện, hiện vẫn còn gặp một số vấn đề, cụ thể:

1. Trong các kho dữ liệu, dữ liệu được lưu trữ rất đa dạng vì chúng phục vụ nhiều công việc khác nhau. Nhiều thuộc tính cung cấp các thông tin có khả năng dự đoán sự việc nên rất có ý nghĩa trong quá trình học nhưng cũng có nhiều thuộc tính không có khả năng phản ánh thông tin dự đoán mà chỉ có ý nghĩa lưu trữ, thống kê bình thường. Vì vậy, khi chúng ta chọn tập mẫu không đặc trưng

thì mô hình cây quyết định có được sau khi huấn luyện sẽ không có khả năng dự đoán [88].

2. Tất cả các phương pháp học quy nạp cây quyết định như CART, ID3, C4.5, SLIQ, SPRINT,… điều cần đến sự nhất quán của tập mẫu [24], [47], [48], [60], [62]. Tuy nhiên trong bài toán phân lớp dữ liệu bằng cây quyết định mờ, còn có sự xuất hiện của các thuộc tính chứa giá trị ngôn ngữ, tức $\exists A_i \in D$ có miền trị $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, với D_{A_i}là tập các giá trị kinh điển của A_i và LD_{A_i} là tập các giá trị ngôn ngữ của A_i. Trong trường hợp này, các thuật toán học quy nạp trên sẽ lựa chọn cách thức bỏ qua các bộ dữ liệu "*lỗi*" nằm ở miền giá trị LD_{A_i} này, hay người huấn luyện có thể nhờ ý kiến chuyên gia để xác định các giá trị "*lỗi*" trong quá trình học. Việc này sẽ làm mất dữ liệu hay phụ thuộc lớn vào trình độ của chuyên gia, ví dụ thân nhiệt của một người "*rất cao*" có giá trị được xác định là 40 nhưng tuổi thọ có giá trị 40 được lưu trữ thì lại có giá trị "*rất thấp*" [1] nên kết quả thu được không phải lúc nào cũng thật sự phù hợp. Thêm vào đó, các thuật toán học quy nạp này luôn cố định trước cách phân chia tại các điểm phân chia ứng với thuộc tính rời rạc, do đó mô hình của cây kết quả thu được không linh động tại các thuộc tính rời rạc khác nhau của tập mẫu huấn luyện.

3. Đại số gia tử với lợi thế của mình nên là một công cụ hữu hiệu nhằm thuần nhất các giá trị thực và các giá trị ngôn ngữ trong thuộc tính mờ về theo giá trị ngôn ngữ hay theo giá trị thực. Như thế, đây là một hướng tiếp cận phù hợp cho bài toán phân lớp dữ liệu bằng cây quyết định mờ. Tuy vậy, việc sử dụng đại số gia tử để định lượng cho các giá trị ngôn ngữ thông thường được dựa vào miền giá trị rõ của chính thuộc tính đang xét đó đó tức là ta có thể tìm thấy miền trị $[\psi_{min}, \psi_{max}]$ từ miền giá trị rõ và sau đó sẽ định lượng cho các giá trị ngôn ngữ từ miền trị này [1], [4]. Việc tìm miền trị $[\psi_{min}, \psi_{max}]$ không phải lúc nào cũng thuận lợi vì có thể xuất hiện các giá trị ngôn ngữ mà giá trị thật sự của nó nằm ngoài miền dữ liệu rõ đang có trong thuộc tính đang xét. Các giá trị ngôn ngữ này ta gọi là các giá trị (ngôn ngữ) "*ngoại lai*". Việc sử dụng đại số gia tử để định lượng cho các giá trị này hiện vẫn còn gặp nhiều khó khăn. Các phương pháp tiền xử lý dữ liệu truyền thống như sử dụng giá trị hằng toàn cục hay sử dụng giá trị trung bình của thuộc tính, phương pháp Binning, hồi quy,... [20],

[58], [71] khó có thể sử dụng cho việc định lượng các giá trị ngoại lai này. Ta có thể bỏ qua các trường hợp ngoại lai này hoặc xem chúng cùng lớp tương đương với các giá trị ngôn ngữ khác nhưng việc làm này sẽ làm mất thông tin.

Trong chương này, trên cơ sở phân tích mối tương quan giữa các thuật toán học cây quyết định nền tảng và phân tích sự ảnh hưởng tập mẫu huấn luyện đối với cây kết quả thu được, luận án trình bày một cách có hệ thống phương pháp lựa chọn tập mẫu huấn luyện và đề xuất thuật toán phục vụ việc học cây quyết định linh hoạt. Đồng thời, luận án cũng đưa ra mô hình học khi tập mẫu huấn luyện có chứa giá trị mờ, định nghĩa các giá trị ngôn ngữ ngoại lai và đề xuất thuật toán nhằm thuần nhất miền trị cho các thuộc tính theo tiếp cận đại số gia tử. Cuối cùng sẽ trình bày thuật toán FMixC4.5 phục vụ cho việc học cây quyết định trên tập huấn luyện mờ. Các thuật toán MixC4.5 và FMixC4.5 đề xuất trong luận án được cài đặt thử nghiệm, đánh giá dựa trên 2 bộ dữ liệu mẫu Northwind, Mushroom và công bố ở các công trình liên quan [CT1], [CT2], [CT3] và [CT6].

2.2. Phương pháp chọn tập mẫu huấn luyện đặc trưng cho bài toán phân lớp dữ liệu bằng cây quyết định

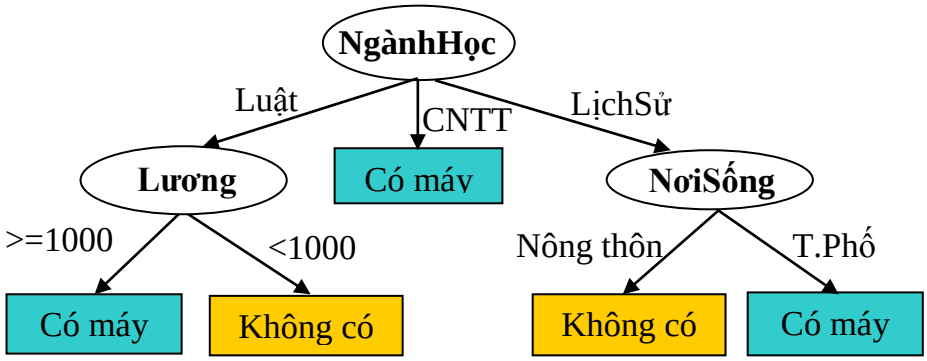
Ví dụ 2.1. Với dữ liệu DIEUTRA được khảo sát về tình hình có sử dụng máy tính xách tay của nhân viên như Bảng 2.1, cần chọn tập mẫu huấn luyện để huấn luyện cây quyết định cho bài toán dự đoán.

Bảng 2.1. Bảng dữ liệu DIEUTRA

ID	PhiếuĐT	HọVàTên	NơiSống	NgànhHọc	KinhTếGĐ	Lương	PhụCấp	MáyTính
750001	M01045	Nguyễn Văn An	T.Phố	Luật	Chưa tốt	450	45	Không
750002	M01087	Lê Văn Bình	NôngThôn	Luật	Chưa tốt	400	40	Không
750003	M02043	Nguyễn Thị Hoa	T.Phố	CNTT	Chưa tốt	520	52	Có
750004	M02081	Trần Bình	T.Phố	LịchSử	Trung bình	340	34	Có
750005	M02046	Trần Thị Hương	T.Phố	LịchSử	Khá	500	50	Có
750006	M03087	Nguyễn Thị Lài	NôngThôn	LịchSử	Khá	1000	100	Không
750007	M03025	Vũ Tuấn Hoa	NôngThôn	CNTT	Khá	2000	200	Có
750008	M03017	Lê Bá Linh	T.Phố	Luật	Trung bình	350	35	Không

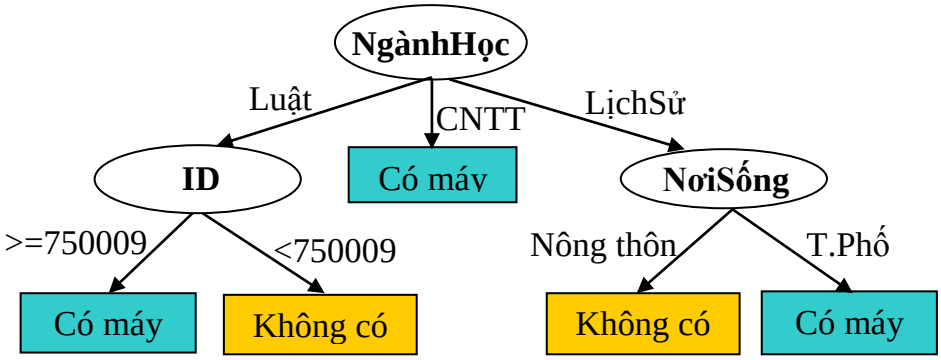
750009	M04036	Bạch Ân	T.Phố	Luật	Khá	1000	100	Có
750010	M04037	Lý Thị Hoa	T.Phố	LịchSử	Trung bình	500	50	Có
750011	M04042	Vũ Quang Bình	NôngThôn	Luật	Trung bình	1000	100	Có
750012	M04083	Nguyễn Hoa	NôngThôn	CNTT	Trung bình	400	40	Có
750013	M05041	Lê Xuân Hoa	T.Phố	CNTT	Chưa tốt	550	55	Có
750014	M05080	Trần Quế Chung	NôngThôn	LịchSử	Trung bình	500	50	Không

Giả sử chọn ***M1 = (NơiSống, NgànhHọc, KinhTếGĐ, Lương)*** làm tập mẫu huấn luyện, cây quyết định thu được áp dụng thuật toán C4.5 cho kết quả như ở Hình 2.1.



Hình 2.1. Cây quyết định được tạo từ tập mẫu huấn luyện M1

Tuy nhiên, nếu chọn ***M2 = (ID, NơiSống, NgànhHọc, KinhTếGĐ)*** làm tập mẫu huấn luyện, cũng áp dụng thuật toán C4.5 ta lại có cây kết quả thu được như ở Hình 2.2.



Hình 2.2. Cây quyết định được tạo từ tập mẫu huấn luyện M2

Như vậy, so sánh giữa cây Hình 2.1 và Hình 2.2, chúng ta dễ dàng nhận thấy cây ở Hình 2.2. là một cây không có khả năng dự đoán theo nhánh ***ID***, do không phản ánh được bản chất thực tế của dữ liệu cần học.

2.2.1. Tính chất thuộc tính của tập mẫu huấn luyện đối với quá trình huấn luyện

Như đã phân tích ở Mục 1.3.4 về tình trạng quá khớp của mô hình cây quyết định sau khi huấn luyện hay ở Ví dụ 2.1 về khả năng ứng dụng trong dự đoán, việc lựa chọn các thuộc tính tham gia vào tập mẫu trong quá trình huấn luyện là vấn đề đầu tiên phải quan tâm. Thông qua việc phân loại các thuộc tính rời rạc, luận án nhằm cung cấp sự lựa chọn xác đáng cho việc lựa chọn tập mẫu trong quá trình huấn luyện.

Trong tập mẫu huấn luyện có m thuộc tính, chiều cao tối đa của cây kết quả là $m - 1$, theo Định nghĩa 1.19, các thuộc tính rời rạc có lực lượng lớn hơn m rất có khả năng làm xuất hiện cây dàn trải theo chiều ngang tại nút đó vì có chiều rộng của nhánh lớn hơn chiều sâu của cây. Tuy vậy, sự phân nhánh của cây còn phụ thuộc vào lực lượng của thuộc tính huấn luyện Y tức là phụ thuộc vào giá trị $|Y|$, vì chúng ta cần phân chia đến các lớp thuần nhất theo mỗi giá trị của Y. Do vậy, khi xét tính dàn trải theo chiều ngang của cây, ta cần phải xét đến số lớp phân chia trên Y cùng với chiều cao của cây. Định nghĩa sau nhằm xác định ngưỡng có thể cho phép dàn trải đối với mỗi nút trên cây.

Định nghĩa 2.1. Thuộc tính $A_i \in D$ được gọi là thuộc tính có giá trị riêng biệt (gọi tắt là thuộc tính riêng biệt) nếu như nó là thuộc tính rời rạc và $|A_i| > (m - 1) \times |Y|$. Tập các thuộc tính có giá trị riêng biệt trong D ký hiệu là D^*.

Mệnh đề 2.1. Trong cây quyết định, nếu có một nút được tạo là ứng với một thuộc tính riêng biệt trong quá trình huấn luyện thì đó là một cây dàn trải.

Chứng minh: thật vậy, mẫu D có m thuộc tính nên có chiều cao tối đa có thể của cây sau khi huấn luyện là $m - 1$. Do vậy, tính đúng của mệnh đề được suy ra từ Định nghĩa 2.1.

■

Như chúng ta đã biết, trong các kho dữ liệu nghiệp vụ, do cần phải lưu trữ thông tin phản ảnh thế giới thực nên nhiều thuộc tính được lưu trữ không có khả năng dự đoán mà chỉ có ý nghĩa lưu trữ nhằm mục đính diễn giải thông tin. Các định nhĩa sau nhằm để phân loại các thuộc tính có khả năng tham gia trong quá trình huấn luyện hay không.

Định nghĩa 2.2. Cho tập mẫu *D*. Thuộc tính $A_i \in D$ mà giữa các phần tử a_{i_j}, a_{i_k} với $j \neq k$ là không sánh được thì ta gọi A_i là thuộc tính ghi nhớ trong tập mẫu. Tập các thuộc tính này trong *D* ký hiệu là D^G.

Mệnh đề 2.2. Cho tập mẫu *D*. Nếu thuộc tính $A_i \in D$ là thuộc tính ghi nhớ thì ta loại A_i ra khỏi mẫu *D* mà không làm thay đổi cây quyết định thu được.

Chứng minh: hiển nhiên, bởi ta không thể so sánh giữa các phần tử a_{i_j} với a_{i_k} của A_i để tính hàm *Gain*(A_i, *D*) nên không tồn tại lợi ích thông tin của mỗi bộ trên A_i. Vì thế A_i không thể xuất hiện trên cây kết quả nên ta loại A_i ra khỏi *D* mà không làm thay đổi cây quyết định thu được.

■

Mệnh đề 2.3. Nếu trong tập mẫu huấn luyện chứa thuộc tính A_i là khoá của tập *D* thì cây quyết định thu được là quá khớp tại nút A_i.

Chứng minh: thật vậy, với thuộc tính A_i có *Dom*(A_i) = $\{a_{i_1}, a_{i_2}, \dots, a_{i_n}\}$, do A_i là khoá nên ta có $a_{i_j} \neq a_{i_k}$, $\forall i \neq k$. Như thế, mẫu *D* được phân ra làm *n* phân hoạch, mà mỗi phân hoạch chỉ có *1* bộ nên $\forall a_{i_j} \in A_i$, hàm $E(A_i, D) = 0$. Hàm xác định lợi ích thông tin nhận được trên thuộc tính A_i ở Công thức 1.9 đạt giá trị cực đại, vì thế A_i được chọn làm điểm phân tách cây. Tại đây, cây được phân chia làm *n* nút, mỗi cạnh tương ứng được gán nhãn a_{i_j}, đây là một cây dàn trải theo chiều ngang tại nút ứng với A_i. Do tính duy nhất của khoá nên không có giá trị trùng khớp khi so sánh tại nút này trong quá trình dự đoán. Vậy cây kết quả thu được là quá khớp tại nút A_i, theo Định nghĩa 1.19.

■

Với dữ liệu xét ở Bảng 2.1, các thuộc tính sau là không hiệu quả khi chọn nó trong các tập mẫu huấn luyện:

- Thuộc tính *ID*, *PhiếuĐT* là thuộc tính *khoá*.
- Thuộc tính *HọVàTên* là thuộc tính *có giá trị riêng biệt*.

2.2.2. Ảnh hưởng từ phụ thuộc hàm giữa các thuộc tính trong tập huấn luyện

Mệnh đề 2.4. Cho mẫu *D* với thuộc tính quyết định *Y*. Nếu có phụ thuộc hàm $A_i \rightarrow A_j$ và ta đã chọn A_i làm nút phân tách trên cây thì mọi nút con của nó sẽ không

nhận A_j làm nút phân tách.

Chứng minh: thật vậy, giả sử $|A_i| = k$, khi chọn A_i làm nút phân tách trên cây thì tại nút này ta có k nhánh. Không mất tính tổng quát, các nhánh của cây lần lượt được gán các giá trị là a_{i_j}, $j = 1, \ldots, k$. Do $A_i \rightarrow A_j$ nên tại nhánh bất kỳ thì trên mẫu huấn luyện tương ứng D', lúc này trên thuộc tính A_j sẽ có cùng *1* giá trị. Như thế $Gain(A_j, D') = 0$ là nhỏ nhất nên A_j không thể chọn để làm nút phân tách cây.

■

Mệnh đề 2.5. Trên mẫu D với thuộc tính quyết định Y, nếu có phụ thuộc hàm $A_i \rightarrow A_j$ thì lượng thông tin nhận được trên A_i không nhỏ hơn lượng thông tin nhận được trên A_j.

Chứng minh: thật vậy, giả sử thuộc tính quyết định Y có k giá trị. Do $A_1 \rightarrow A_2$ nên $|A_1| \geq |A_2|$. Theo công thức (1.9), lượng thông tin nhận được trên thuộc tính A_i là $Gain(A_i, D)$. Nên nếu $|A_1| = |A_2|$ thì trên A_1 hay A_2 đều có k phân hoạch như nhau nên $Gain(A_1, D) = Gain(A_2, D)$. Ngược lại nếu $|A_1| > |A_2|$ tức tồn tại $a_{1_i}, a_{1_j} \in A_1$, $a_{1_i} \neq a_{1_j}$ mà trên tương ứng trên A_2 thì $a_{2_i} = a_{2_i}$. Lúc này *2* phân hoạch trên A_1 được gộp thành 1 phân hoạch trên A_2 nên Entropy tương ứng trên A_2 lớn hơn. Vậy $Gain(A_1, D) > Gain(A_2, D)$.

■

Từ Mệnh đề 2.5, ta dễ dàng suy ra hệ quả nhằm giới hạn các thuộc tính trong tập huấn luyện như sau:

Hệ quả 2.1. Nếu có phụ thuộc hàm $A_1 \rightarrow A_2$ mà A_1 không phải là thuộc tính khóa của mẫu D thì thuộc tính A_2 không được chọn làm nút phân tách cây.

Với dữ liệu được xét ở Bảng 2.1, thuộc tính *PhụCấp* là phụ thuộc hàm vào thuộc tính không khóa *Lương* nên không hiệu quả khi chọn trong các tập mẫu huấn luyện cây.

Như vậy, với các kho dữ liệu nghiệp vụ được lưu trữ bao gồm các thông tin mô tả về phân loại dữ liệu các thuộc tính và các phụ thuộc hàm của chúng, chúng ta có thể chọn được tập mẫu huấn luyện phù hợp thông qua thuật toán "*chọn mẫu đặc trưng*" cho quá trình huấn luyện cây như sau:

Thuật toán tìm tập huấn luyện đặc trưng từ dữ liệu mẫu.

Vào : *Tập mẫu huấn luyện D được chọn từ dữ liệu mẫu;*

Ra : *Tập mẫu huấn luyện đặc trưng* D^**;*

Mô tả thuật toán:

Begin

For each $i = 1$ to m do

Begin

If $(A_i \in \{Khoá, Ghi\ nhớ\})$ then $D = D - A_i$;

End;

$i = 1$;

While $(i < m)$ do

Begin

$j = i + 1$;

While $(j \leq m)$ do

Begin

If $(A_i \rightarrow A_j$ and (A_i *không phải thuộc tính khóa trong D*))

then $D = D - A_j$

Else

If $(A_j \rightarrow A_i$ and (A_j *không phải thuộc tính khóa trong D*))

then $D = D - A_i$;

$j = j+1$;

End;

$i = i + 1$;

End;

End;

Tính đúng đắn của thuật toán được suy ra từ các mệnh đề ở trên. Với tập dữ liệu nghiệp vụ được chọn có m thuộc tính, do chúng ta phải duyệt qua 2 lần lồng nhau các thuộc tính hiện có nên thuật toán có độ phức tạp $O(m^2)$.

2.3. Phân lớp dữ liệu bằng cây quyết định dựa trên ngưỡng miền trị thuộc tính

2.3.1. Cơ sở của việc xác định ngưỡng cho quá trình học phân lớp

Như chúng ta đã xét, các thuật toán học quy nạp cây quyết định đều dựa vào việc chọn thuộc tính có lượng thông tin tốt nhất để phân tách cây và sự phân chia tại mỗi nút phụ thuộc vào kiểu của thuộc tính là liên tục hay rời rạc. Tất cả các thuật toán đều cố định cách phân chia cho mọi thuộc tính rời rạc của tập huấn luyện theo *nhị phân* hoặc *k-phân*.

- Đối với cách chia *k-phân*, một điều dễ thấy là nếu thuộc tính *A* có lực lượng lớn sẽ làm cho số nút của cây tại một cấp tăng lên nhanh. Điều này làm tăng chiều rộng của cây nên cây sẽ dàn trải theo chiều ngang. Hơn nữa, cách chia này có khả năng dẫn đến lỗi khi dữ liệu không thể đoán nhận được lớp. Mặc dù vậy chia *k-phân* theo thuộc tính rời rạc có ưu điểm là độ phức tạp thấp, bởi vì sau khi phân thì thuộc tính đó không cần phải sử dụng lại nữa.

- Cách chia *nhị phân theo giá trị tại điểm phân chia* [48] [87] không làm tăng chiều rộng của cây, bởi cho dù *k* có lớn bao nhiêu cũng chỉ chia theo 2 nút, một nút là giá trị được chọn và một nút là tập còn lại. Tuy nhiên, điều này lại làm tăng nhanh chiều sâu của cây. Cách chia *nhị phân theo tập hợp tại điểm phân chia* [47] [52] luôn tách thuộc tính rời rạc làm 2 tập con nên và chi phí tính toán rất lớn và khó khăn trong việc duyệt cây kết quả cho quá trình dự đoán.

Từ những nhận định trên, ta nhận thấy cần phải xây dựng một thuật toán học với cách chia hỗn hợp *nhị phân, k-phân* theo từng thuộc tính nhằm có được cây với chiều rộng và chiều sâu hợp lý cho quá tình huấn luyện.

2.3.2. Thuật toán MixC4.5 dựa trên ngưỡng miền trị thuộc tính

Với tập mẫu huấn luyện *D* với *m* thuộc tính A_1, A_2, ... A_m có lực lượng tương ứng của mỗi thuộc tính là $|A_1|$, $|A_2|$, … $|A_m|$. Ta gọi *k* là ngưỡng giới hạn sự phân chia tại mỗi thuộc tính theo *nhị phân*, tức là nếu lực lượng của thuộc tính nhỏ hơn một giá trị được lựa chọn *k* cho trước thì sẽ phân theo *k-phân*, ngược lại phân theo *nhị phân*.

Do tính chất dàn trải của cây chỉ xảy ra trên thuộc tính rời rạc nên khi A_i không phải là thuộc tính riêng biệt (theo Định nghĩa 2.1) thì $|A_i| < (m - 1) \times |Y|$.

Với thuộc tính A_i có $|A_i| < (m - 1) \times |Y|$ thì các nhánh của cây không lớn hơn chiều cao nên cây kết quả không dàn trải theo chiều ngang. Khi $|A_i| \geq (m - 1) \times |Y|$, ta có sự phân chia theo chiều ngang của nút tương ứng lớn hơn chiều rộng của cây. Tuy vậy, vì tính chất thuần nhất của sự phân lớp nên ta cần phân chia các nhánh của cây theo các giá trị riêng biệt của thuộc tính dự đoán *Y*. Do vậy, giá trị $(m - 1) \times |Y|$ sẽ được chọn để xác định ngưỡng phân chia. Khi $|A_i| < (m - 1) \times |Y|$ thì ta sẽ phân chia theo *k-phân* như C4.5. Ngược lại thì $|A_i| \geq (m - 1) \times |Y|$ nên có thể xảy ra tình trạng quá khớp tại nút này, vì vậy ta phân chia thuộc tính này theo *nhị phân* của SPRINT.

Với tập mẫu dữ liệu nghiệp vụ huấn luyện *D* có *m* thuộc tính, ta có thuật toán MixC4.5 xây dựng cây quyết định *S* như sau:

Thuật toán MixC4.5

Vào: *mẫu huấn luyện D có n bộ, m thuộc tính dự đoán, thuộc tính quyết định Y.*

Ra: *Cây quyết định S.*

Mô tả thuật toán:

ChonMauDacTrung(D); //theo Mục 2.2.2

Tính ngưỡng k cho các thuộc tính;

Khởi tạo tập các nút lá S = D;

For each (*nút lá L thuộc S*) do

If (*L thuần nhất*) or (*L là rỗng*) then

Gán nhãn cho nút tương ứng với giá trị thuần nhất của L;

Else

Begin

X = Thuộc tính tương ứng có GainRatio lớn nhất;

Gán nhãn cho nút tương ứng với tên thuộc tính X;

If (*L là thuộc tính liên tục*) then *//Phân chia theo C4.5*

Begin

Chọn ngưỡng T tương ứng có Gain lớn nhất trên X;

$S_1 = \{x_i | x_i \in Dom(L), x_i \leq T\}$; $S_2 = \{x_i | x_i \in Dom(L), x_i > T\}$;

Tạo 2 nút con cho nút hiện tại tương ứng với hai tập S_1 *và* S_2;

Đánh dấu nút L đã xét;

End

Else *//L là thuộc tính rời rạc*

//phân chia k-phân theo C4.5 khi |L| không vượt qua ngưỡng k

If $|L| < k$ then

Begin

$P = \{x_i | x_i \in K, x_i$ *đơn nhất*$\}$;

For each ($x_i \in P$) do

Begin

$S_i = \{x_j | x_j \in Dom(L), x_j = x_i\}$;

Tạo nút con thứ i cho nút hiện tại tương ứng với S_i;

End;

End

Else

//phân chia nhị phân theo SPRINT tại giá trị tương ứng khi

//có |L| vượt ngưỡng k

Begin

Lập ma trận đếm cho các giá trị trong L;

T = Giá trị trong L có Gain lớn nhất;

$S_1 = \{x_i | x_i \in Dom(L), x_i = T\}$; $S_2 = \{x_i | x_i \in Dom(L), x_i \neq T\}$;

Tạo 2 nút con cho nút hiện tại ứng với hai tập S_1 *và* S_2;

End;

Đánh dấu nút L đã xét;

End;

End;

Với *m* là số thuộc tính, *n* là số thể hiện của tập huấn luyện. Tuần tự, chúng ta mất $O(m \times n)$ vì phải duyệt qua toàn bộ mẫu để xác định ngưỡng *k* cho *m* thuộc tính, là ngưỡng xác định sẽ chia tách theo nhị phân hay *k-phân* và tuần tự. Sau đó chúng ta mất chi phí $O(m^2 \times n)$ để lựa chọn các thuộc tính đặc trưng cho tập mẫu huấn luyện nhằm tránh tình trạng quá khớp trên cây.

Trong quá trình huấn luyện, với thuộc tính liên tục MixC4.5 hoàn toàn trùng khớp với C4.5 và SPRINT. Đối với thuộc tính rời rạc, MixC4.5 được thiết kế dựa trên sự tổng hợp của C4.5 và SPRINT, khi lực lượng của thuộc tính đang xét chưa vượt ngưỡng *k*, do chúng ta sử dụng *k-phân* theo C4.5 nên độ phức tạp lúc này là $O(m \times n \times log\ n)$. Ngược lại, khi vượt quá ngưỡng *k*, chúng ta phân chia nhị phân theo giá trị theo SPRINT nên độ phức tạp lúc này là $O(m \times n^2 \times log\ n)$. Vậy độ phức tạp của thuật toán MixC4.5 là $O(m \times n^2 \times log\ n)$.

Tính đúng và tính dừng của thuật toán được rút ra từ các thuật toán C4.5 và SPRINT do MixC4.5 được kết hợp từ hai thuật toán này.

2.3.3. Cài đặt thử nghiệm và đánh giá thuật toán MixC4.5

Chương trình thực nghiệm được cài đặt bằng ngôn ngữ Java Eclipse Mars Release 4.50 trên máy tính có cấu hình: Processor: Intel® Core™i5-2450 CPU @ 2.50GHz (4CPUs), ~ 2.50 GHz, RAM4GB, System type 64bit.

a. Với tập mẫu huấn luyện gồm 1500 bảng ghi và 500 bộ giá trị kiểm tra được lấy từ 2155 bộ dữ liệu từ các bảng Customers, Details, OrderDetails, Products của cơ sở dữ liệu Northwind, các thông số cụ thể như Bảng 2.2. Kết quả thực nghiệm đo được ở các Bảng 2.3 và Bảng 2.4.

Bảng 2.2. Thông số thuộc tính tập huấn luyện chọn từ cơ sở dữ liệu Northwind

STT	Tên trường	Lực lượng	Kiểu thuộc tính	Miền trị
1	Customers.CompanyName	91	Rời rạc	
2	Customers.ContactName	91	Rời rạc	
3	Customers.ContactTitle	12	Rời rạc	
4	Customers.City	69	Rời rạc	
5	Customers.Region	19	Rời rạc	
6	Customers.Phone	91	Rời rạc	
7	Products.ProductName	77	Rời rạc	
8	Products.SupplierID	29	Rời rạc	
9	Products.CategoryID	8	Rời rạc	

10	Products.QuantityPerUnit	70	Rời rạc	
11	Products.UnitPrice	62	Liên tục	[2.50, 265.0]
12	Products.UnitsInStock	51	Liên tục	[0, 125]
13	Products.UnitsOnOrder	10	Liên tục	[0, 100]
14	Products.ReorderLevel	7	Liên tục	[0, 30]
15	Products.Discontinued	2	Logic	
16	Orders.CustomerID	89	Rời rạc	
17	Orders.EmployeeID	9	Liên tục	[1, 9]
18	Orders.OrderDate	480	Rời rạc	
19	Orders.RequiredDate	454	Rời rạc	
20	Orders.ShippedDate	388	Rời rạc	
21	Orders.Freight	79	Liên tục	[0.02, 1007.6]
22	Orders.ShipCity	70	Rời rạc	
23	Orders.ShipRegion	20	Rời rạc	
24	OrderDetails.ProductID	77	Liên tục	[1, 77]
25	OrderDetails.UnitPrice	116	Liên tục	[2.0, 263.5]
26	OrderDetails.Quantity	55	Liên tục	[1, 130]
27	OrderDetails.Discount	11	Liên tục	[0.0, 0.25]

Bảng 2.3. Bảng so sánh kết quả huấn luyện của thuật toán MixC4.5 với 1000 mẫu trên cơ sở dữ liệu Northwind

Thuật toán	**Thời gian (s)**	**Tổng số nút**	**Độ chính xác (%)**
C4.5	11.2	389	69.2
SLIQ	220.2	89	76.4
SPRINT	89.2	122	79.8
MixC4.5	**73.3**	**130**	**78.2**

Bảng 2.4. Bảng so sánh kết quả huấn luyện của thuật toán MixC4.5 với 1500 mẫu trên cơ sở dữ liệu Northwind

Thuật toán	**Thời gian (s)**	**Tổng số nút**	**Độ chính xác (%)**
C4.5	20.4	552	76.4
SLIQ	523.3	162	82.4
SPRINT	184.0	171	83.2
MixC4.5	**186.6**	**172**	**86.6**

b. Với tập mẫu huấn luyện gồm 5000 bảng ghi và các bộ dữ liệu kiểm tra gồm 500 và 1000 mẫu được lấy từ 8000 bộ dữ liệu của cơ sở dữ liệu Mushroom, các thông số cụ thể như Bảng 2.5. Kết quả thực nghiệm của quá trình huấn luyện và kiểm tra được trình bày ở Bảng 2.6.

Bảng 2.5. Thông số thuộc tính tập huấn luyện từ cơ sở dữ liệu Mushroom

STT	**Tên trường**	**Lực lượng**	**Kiểu thuộc tính**	**Miền trị**
1	Bruises	2	Rời rạc	
2	Capshape	6	Rời rạc	
3	CapSurface	4	Rời rạc	
4	CapColor	10	Rời rạc	
5	Bruises	2	Rời rạc	
6	Odor	9	Rời rạc	
7	GillAttachment	2	Rời rạc	
8	GillSpacing	2	Rời rạc	
9	GillSize	2	Rời rạc	
10	GillColor	12	Rời rạc	
11	StalkShape	2	Rời rạc	
12	StalkRoot	5	Rời rạc	
13	StalkSurfaceAboveRing	4	Rời rạc	
14	StalkSurfaceBelowRing	4	Rời rạc	

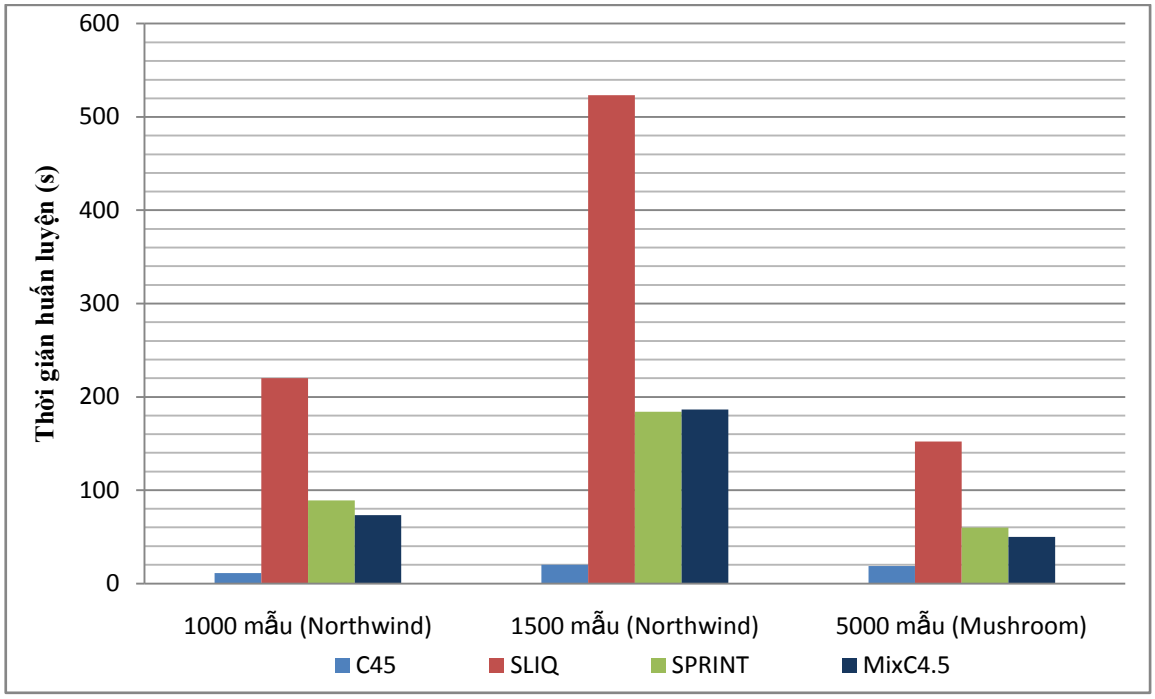
15	StalkColorAboveRing	9	Logic	
16	StalkColorBelowRing	9	Rời rạc	
17	VeilType	9	Rời rạc	
18	VeilColor	4	Rời rạc	
19	RingNumber	3	Rời rạc	
20	RingType	5	Rời rạc	
21	SporePrintColor	9	Rời rạc	
22	Population	62	Mờ	Rõ: [1, 50]
23	Habitat	126	Mờ	Rõ: [20, 90]
24	Classes	2	Logic	

Bảng 2.6. Bảng so sánh kết quả của thuật toán MixC4.5 với 5000 mẫu huấn luyện trên cơ sở dữ liệu có chứa thuộc tính mờ Mushroom

Thuật toán	**Thời gian huấn luyện (s)**	**Độ chính xác (%) 500 mẫu kiểm tra**	**Độ chính xác (%) 1000 mẫu kiểm tra**
C4.5	18.9	54.8	51.2
SLIQ	152.3	51.8	52.2
SPRINT	60.1	54.2	54.6
MixC4.5	**50.2**	**54.8**	**54.6**

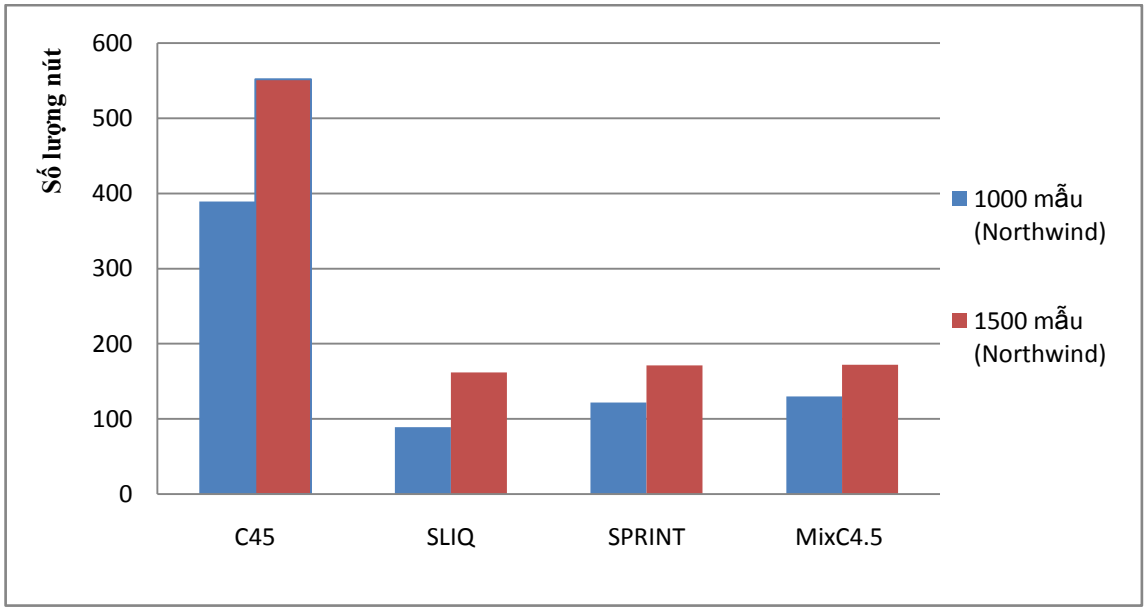
c. Từ kết quả thu được, chúng ta có các nhận xét, đánh giá như sau:

♦ **Thời gian huấn luyện:** Thuật toán C4.5 luôn thực hiện *k-phân* tại các thuộc tính rời rạc và loại bỏ các thuộc tính của tập huấn luyện ở mỗi bước phân chia, nên C4.5 luôn đạt tốc độ thực hiện nhanh nhất. Thời gian xử lý của SLIQ là lớn nhất do phải thực hiện các phép tính *Gini* trên mỗi giá trị rời rạc của thuộc tính rời rạc. Do cách phân chia của MixC4.5 trộn lẫn giữa C4.5 và SPRINT và C4.5 nhanh hơn SPRINT nên thời gian huấn luyện của MixC4.5 khá tương đồng tốt với SPRINT, thể hiện ở Hình 2.3.

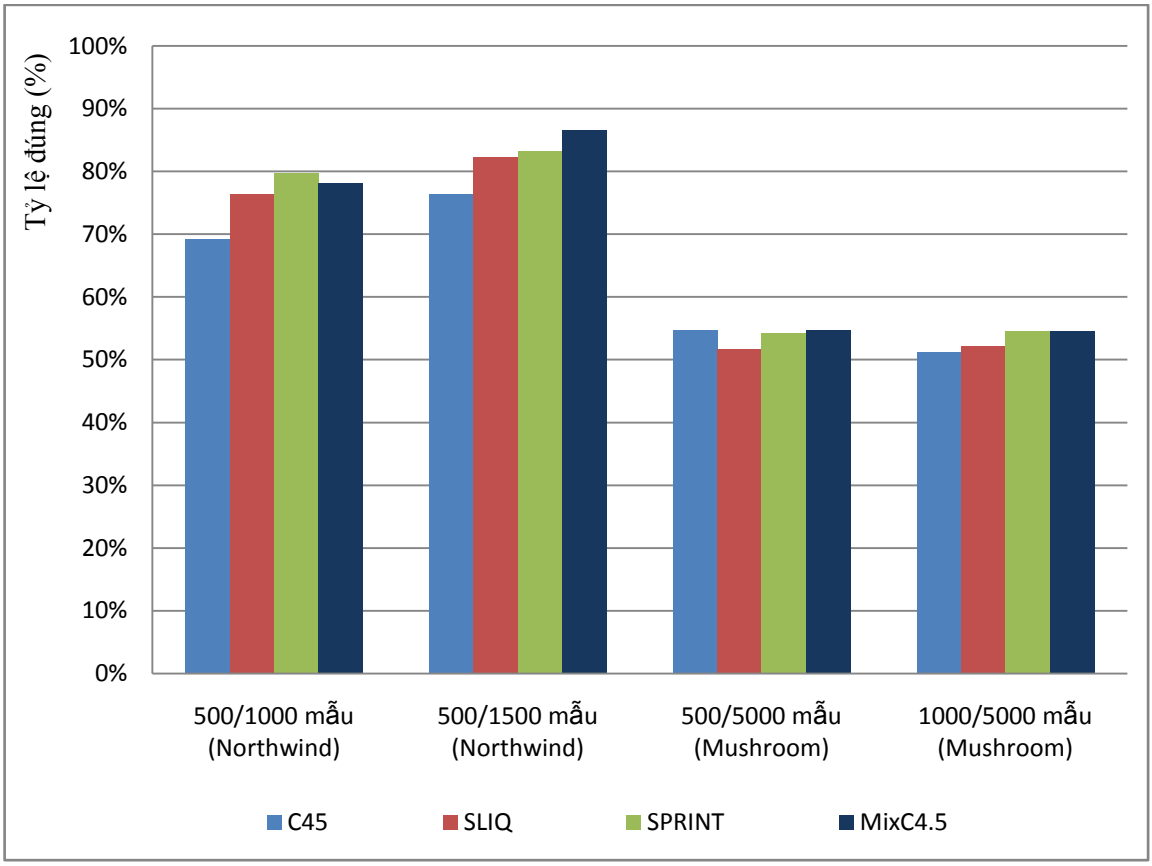


Hình 2.3. So sánh thời gian huấn luyện của MixC4.5 với các thuật toán khác.

♦ **Kích thước cây kết quả:** SLIQ do thực hiện cách chia nhị phân theo tập hợp nên số nút của nó luôn nhỏ nhất và C4.5 luôn phân chia *k-phân* nên số nút luôn lớn nhất. Thuật toán MixC4.5 tương đồng kém với SPRINT do số lượng nút của thuật toán SPRINT ít hơn C4.5, Hình 2.4.



Hình 2.4. So sánh số nút trên cây kết quả của MixC4.5 với các thuật toán khác.



Hình 2.5. So sánh tỷ lệ đúng trên kết quả của MixC4.5 với các thuật toán khác.

♦ **Hiệu quả dự đoán:** Thuật toán MixC4.5 cải tiến từ sự kết hợp các thuật toán C4.5 và SPRINT nên cho cây kết quả có khả năng dự đoán khả quan hơn các thuật toán kinh điển. Trong tất cả các trường hợp dự đoán trên cả dữ liệu rõ và dữ liệu có chứa giá trị mờ, MixC4.5 luôn cho kết quả dự đoán phù hợp hơn C4.5, SLIQ và SPRINT do chúng ta đã hạn chế được tình huống "*quá khớp*" trên cây kết quả.

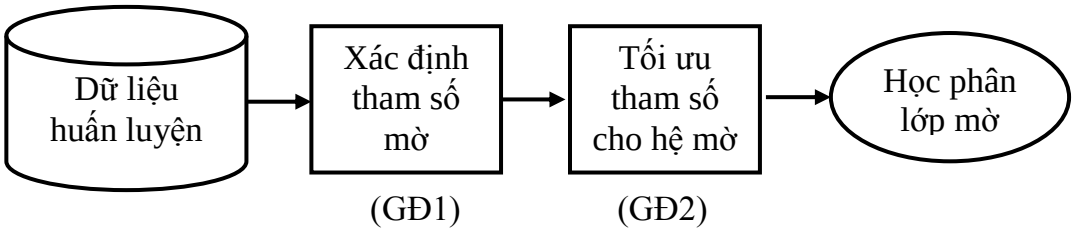
Tuy nhiên, đối sánh giữa các tập huấn luyện không có thuộc tính mờ (Northwind) và các tập huấn luyện có chứa thuộc tính mờ (Mushroom) thì khả năng dự đoán của MixC4.5 còn có sự chênh lệch lớn, khả năng dự đoán khi dữ liệu có chứa giá trị mờ chưa cao. Trong tất cả các trường hợp, các thuật toán kinh điển và thuật toán đề xuất MixC4.5 đều cho kết quả dự đoán đúng có tỷ lệ nhỏ hơn 60%, Hình 2.5. Điều này hoàn toàn hợp lý vì trong quá trình học các thuật toán đang xét không thể xử lý nên chọn giải pháp bỏ qua các giá trị mờ, vì thế kết quả dự đoán có sai số lớn.

2.4. Phân lớp dữ liệu bằng cây quyết định mờ dựa trên đối sánh điểm mờ

2.4.1. Xây dựng mô hình học phân lớp dữ liệu bằng cây quyết định mờ

Như chúng ta đã biết, bài toán phân lớp mờ đã và đang được nhiều tác giả nghiên cứu và ứng dụng, các phương pháp được biết đến như lập luận xấp xỉ mờ [5], hệ nơ-ron mờ [9], [12], [23], luật kết hợp mờ [8], [22], cây quyết định mờ [16], [17], [45],… Các phương pháp này sử dụng các phép toán truyền thống trên tập mờ để lập luận cho kết quả đầu ra. Mô hình thể hiện cho quá trình phân lớp mờ này bao gồm 2 giai đoạn, thể hiện như ở Hình 2.6.

- *Giai đoạn 1:* xác định hệ mờ bao gồm việc lựa chọn các biến vào, các tham số mờ, phân hoạch các khoảng mờ của các biến vào, lập luận đầu ra cho hệ mờ.



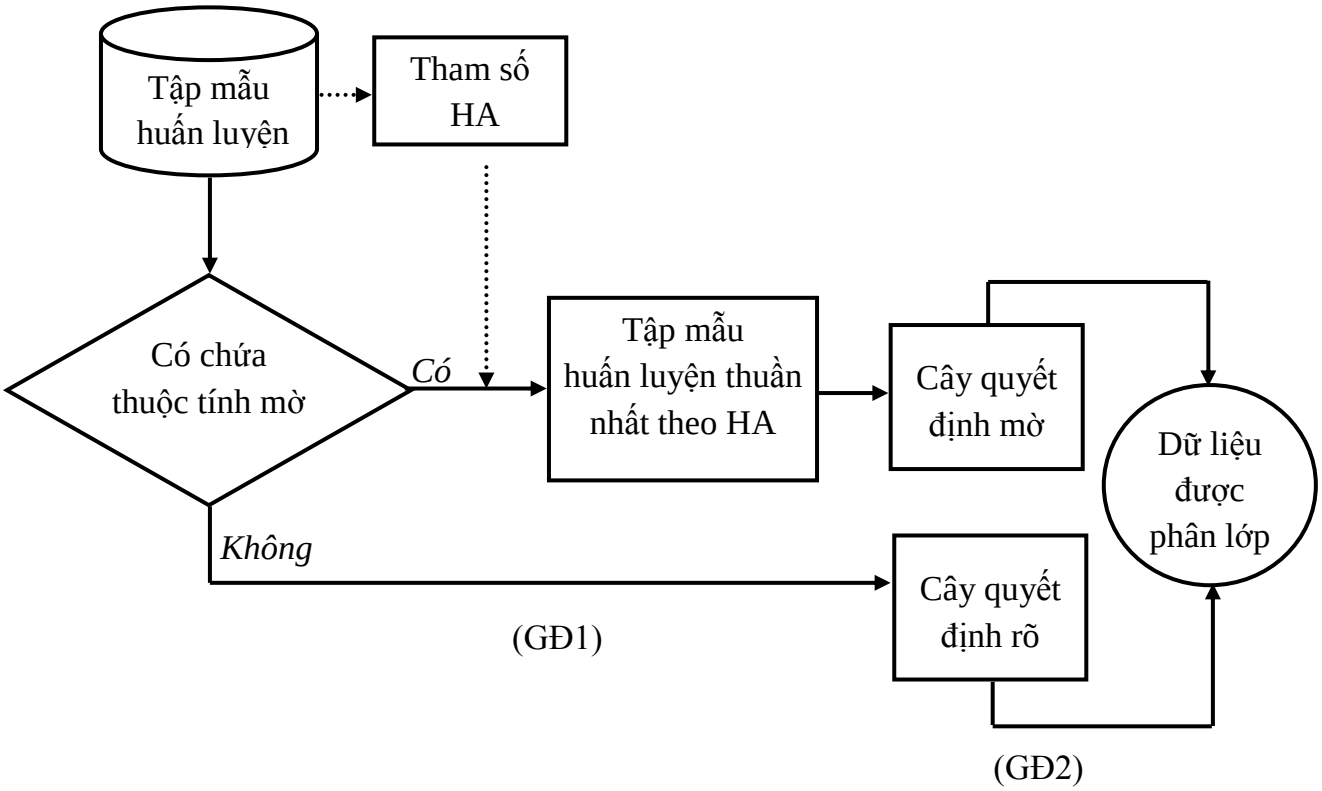
Hình 2.6. Mô hình cho quá trình học phân lớp mờ

- *Giai đoạn 2:* tối ưu các tham số của hệ mờ nhằm nâng cao hiệu quả của việc học phân lớp.

Do nhu cầu phản ánh thế giới thực nên các kho dữ liệu nghiệp vụ rất đa dạng. Vì vậy, tập mẫu huấn luyện được lấy từ các kho dữ liệu sẽ có các thuộc tính chứa cả giá trị rõ và giá trị mờ - thường được biểu diễn bằng các giá trị ngôn ngữ và về bản chất, đây không phải là thuộc tính rời rạc. Các phương pháp học cây quyết định truyền thống mặc dầu có độ phức tạp thấp nhưng lại xem là các thuộc tính mờ như là thuộc tính rời rạc và tiến hành *k-phân* theo giá trị tại điểm này [11], [32], [40], [69], [70],… Do vậy, cây kết quả nhận được sẽ dàn trải theo chiều ngang nên sẽ dẫn đến tình trạng "quá khớp", vì vậy mô hình nhận được sau quá trình học không thật sự hiệu quả.

Phân lớp dữ liệu mờ nói chung và phân lớp dữ liệu mờ bằng cây quyết định nói riêng sử dụng lý thuyết tập mờ luôn gặp phải những hạn chế xuất phát từ bản thân nội tại của lý thuyết tập mờ đó là cấu trúc thứ tự cảm sinh trên các khái niệm mờ biểu thị bằng các giá trị ngôn ngữ không được thể hiện trên các

tập mờ. Thêm vào đó, việc áp dung các phương pháp học truyền thống để học cây quyết định mờ từ tập huấn luyện mờ chưa thể hiện rõ tính mờ trên cây kết quả. Do vậy, khai thác từ những đặc tính về tính có cấu trúc thứ tự của các phần tử là các giá trị ngôn ngữ và các phép toán mờ của đại số gia tử, đã gợi ý chúng ta cần xây dựng một mô hình linh hoạt và phù hợp hơn cho quá trình học phân lớp dữ liệu bằng cây quyết định mờ. Mô hình cho quá trình học được luận án đề xuất như Hình 2.7.



Hình 2.7. Mô hình đề nghị cho việc học phân lớp dữ liệu bằng cây quyết định

- *Giai đoạn 1:* giai đoạn này trước hết sẽ thiết lập các tham số HA và tối ưu cho các tham số này bằng cách lựa chọn từ chính dữ liệu mẫu đang có, phân hoạch các khoảng mờ cho miền trị của các thuộc tính, chuyển tập mẫu huấn luyện nghiệp vụ về tập mẫu chứa các giá trị ngôn ngữ sử dụng đại số gia tử (hoặc các đoạn con của [0, 1]). Giai đoạn này cũng bao hàm việc xử lý tập mẫu huấn luyện nhằm loại bỏ các thuộc tính không hữu ích hay xử lý các giá trị ngoại lai nếu có.

- *Giai đoạn 2:* Áp dụng các phương pháp học để xây dựng cây quyết định rõ hay mờ (cây quyết định có nhãn là các giá trị ngôn ngữ hay các khoảng con

của [0, 1]). Việc xây dựng cây quyết định rõ đã được đề cập ở [14], [24], [47], [48], [61],... Trường hợp xây dựng cây quyết định mờ với nhãn ngôn ngữ (hoặc đoạn con của [0, 1]), ta có thể sử dụng các giải thuật rõ ở đã biết và phân hoạch đa phân theo giá trị ngôn ngữ tại điểm phân chia mờ này hoặc phân lớp nhị phân dựa trên thứ tự của các giá trị ngôn ngữ trong đại số gia tử.

2.4.2. Vấn đề với tập mẫu huấn luyện không thuần nhất

Trong thế giới thực, dữ liệu nghiệp vụ rất đa dạng vì chúng được lưu trữ để phục vụ nhiều công việc khác nhau, nhiều thuộc tính đã được thuần nhất miền giá trị trước khi lưu trữ, nhưng cũng tồn tại nhiều thuộc tính mà miền trị của nó chứa cả dữ liệu rõ và dữ liệu mờ [5], [84], [85]. Ví dụ Bảng 2.7, dữ liệu được lưu trữ ở chứa rõ và mờ.

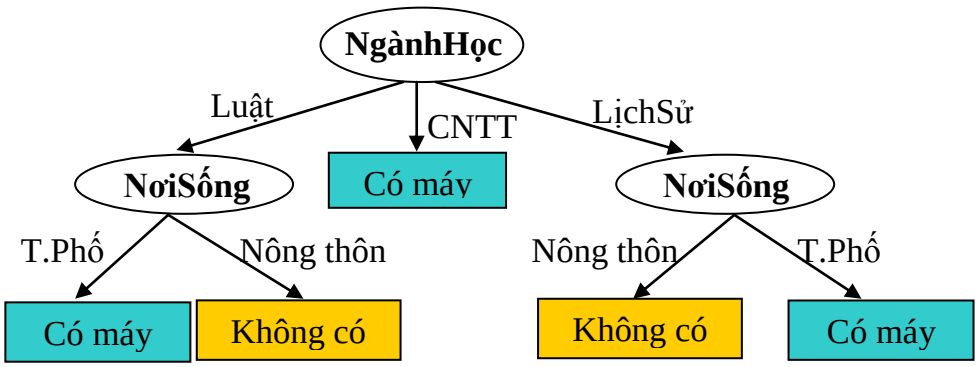
Bảng 2.7. Bảng dữ liệu DIEUTRA có thuộc tính *Lương* chứa dữ liệu rõ mà mờ

PhiếuĐT	HọVàTên	SốCMND	NơiSống	NgànhHọc	KinhTếGĐ	Lương	PhụCấp	MáyTính
M01045	Nguyễn Văn An	193567450	T.Phố	Luật	Chưa tốt	480	45	Không
M01087	Lê Văn Bình	191568422	NôngThôn	Luật	Chưa tốt	Thấp	40	Không
M02043	Nguyễn Thị Hoa	196986568	T.Phố	CNTT	Chưa tốt	530	52	Có
M02081	Trần Bình	191003117	T.Phố	LịchSử	Trung bình	200	34	Có
M02046	Trần Thị Hương	196001278	T.Phố	LịchSử	Khá	Cao	50	Có
M03087	Nguyễn Thị Lài	198235457	NôngThôn	LịchSử	Khá	Cao	100	Không
M03025	Vũ Tuấn Hoa	198875584	NôngThôn	CNTT	Khá	Rấtcao	200	Có
M03017	Lê Bá Linh	191098234	T.Phố	Luật	Trung bình	350	35	Không
M04036	Bạch Ân	196224003	T.Phố	Luật	Khá	1000	100	Có
M04037	Lý Thị Hoa	196678578	T.Phố	LịchSử	Trung bình	500	50	Có
M04042	Vũ Quang Bình	197543457	NôngThôn	Luật	Trung bình	Quácao	100	Có
M04083	Nguyễn Hoa	192267457	NôngThôn	CNTT	Trung bình	Ít thấp	40	Có
M05041	Lê Xuân Hoa	198234309	T.Phố	CNTT	Chưa tốt	550	55	Có
M05080	Trần Quế Chung	196679345	NôngThôn	LịchSử	Trung bình	500	50	Không

Định nghĩa 2.3. Thuộc tính mờ $A_i \in D$ được gọi là thuộc tính không thuần nhất khi miền giá trị của A_i chứa cả giá trị rõ (kinh điển) và giá trị ngôn ngữ. Ký hiệu D_{A_i} là tập các giá trị kinh điển của A_i và LD_{A_i} là tập các giá trị ngôn ngữ của A_i.

Lúc này, thuộc tính không thuần nhất A_i có miền trị là $Dom(A_i) = D_{A_i} \cup LD_{A_i}$.

Khi các thuộc tính chưa thuần nhất này xuất hiện trong tập mẫu huấn luyện, các thuật toán xây dựng cây truyền thống không thể tiến hành. Do đó, chúng ta cần phải tiền xử lý dữ liệu để có được tập mẫu huấn luyện thuần nhất. Vấn đề đặt ra là ta phải xử lý như thế nào để có được kết quả là khả quan. Với dữ liệu ở Bảng 2.7, nếu chúng ta lựa chọn giải pháp bỏ các mẫu "*lỗi*" tức là các mẫu chứa giá trị ngôn ngữ, ta sẽ thu được cây kết quả "*lệch*" như ở Hình 2.8.



Hình 2.8. Cây quyết định kết quả "sai lệch" khi tập mẫu huấn luyện bị loại bỏ giá trị ngôn ngữ

Một cách tự nhiên là chúng ta sử dụng tính chất có thứ tự của các giá trị ngôn ngữ trong đại số gia tử để thuần nhất dữ liệu, tức là chuyển toàn bộ các dữ liệu kinh điển về các giá trị ngôn ngữ hoặc ngược lại.

Định nghĩa 2.4. Cho $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, ν là hàm định lượng ngữ nghĩa của $Dom(A_i)$. Hàm $IC : Dom(A_i) \to [0, 1]$ được xác định như sau:

1. Nếu $LD_{A_i} = \emptyset$ và $D_{A_i} \neq \emptyset$ thì $\forall \omega \in Dom(A_i)$ ta có $IC(\omega) = 1 - \frac{\psi_{\max} - \omega}{\psi_{\max} - \psi_{\min}}$

với $Dom(A_i) = [\psi_{min}, \psi_{max}]$ là miền trị kinh điển của A_i.

2. Nếu $D_{A_i} \neq \emptyset$, $LD_{A_i} \neq \emptyset$ thì $\forall \omega \in Dom(A_i)$ ta có $IC(\omega) = \{\omega \times \nu(\psi_{maxLV})\}/\psi_{max}$, với $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$ là miền trị ngôn ngữ của A_i.

Vậy, nếu chúng ta chọn các tham số W và độ đo tính mờ cho các gia tử sao cho $\nu(\psi_{maxLV}) \approx 1.0$ thì $(\{\omega \times \nu(\psi_{maxLV})\}/\psi_{max}) \approx 1 - \frac{\psi_{\max} - \omega}{\psi_{\max} - \psi_{\min}}$.

Mệnh đề 2.6. Với bất kỳ một thuộc tính không thuần nhất A_i, ta luôn có thể thuần nhất tất cả các giá trị kinh điển D_{A_i} và giá trị ngôn ngữ LD_{A_i} của A_i về giá

trị số thuộc đoạn [0, 1], để từ đó có thể ánh xạ về giá trị ngôn ngữ hay giá trị kinh điển tương ứng.

Tính đúng của mệnh đề được suy ra từ Định nghĩa 1.16 và Định nghĩa 2.4.

■

Ví dụ 2.2. Trong tập mẫu ở Bảng 2.7, ta xây dựng 1 ĐSGT để thuần nhất cho thuộc tính không thuần nhất $A_{Lương}$ = {450, Thấp, 520, 340, Cao, Cao, Rất cao, 350, 1000, 500, Rất cao, Ít thấp, 550, 500} như sau:

$\underline{X}_{Lương}$ = ($X_{Lương}$, $G_{Lương}$, $H_{Lương}$, ≤), với $G_{Lương}$ = {*cao, thấp*}, $H^{+}_{Lương}$= {hơn, *rất*}, $H^{-}_{Lương}$ = {*khả năng, ít*} và quan hệ ngữ nghĩa: *rất* > *hơn, ít* >*khả năng*. $W_{Lương}$ = 0.6, *fm*(*thấp*) = 0.4, *fm*(*cao*) = 0.6, *fm*(*rất*) = 0.35, *fm*(*hơn*) = 0.25, *fm*(*khả năng*) = 0.20, *fm*(*ít*) = 0.20.

Ta có *fm*(*rất thấp*) = 0.35 × 0.4 = 0.14, *fm*(*hơn thấp*) = 0.25 × 0.4 = 0.10, *fm*(*ít thấp*) = 0.2 × 0.4 = 0.08, *fm*(*khả năng thấp*) = 0.2 × 0.4 = 0.08.

Vì *rất thấp* < *hơn thấp* < *thấp* < *khả năng thấp* < *ít thấp* nên : *I*(*rất thấp*) = [0, 0.14], *I*(*hơn thấp*) = [0.14, 0.24], *I*(*khả năng thấp*) = [0.24, 0.32], *I*(*ít thấp*) = [0.32, 0.4].

Ta có *fm*(*rất cao*) = 0.35 × 0.6 = 0.21, *fm*(*hơn cao*) = 025 × 0.6 = 0.15, *fm*(*ít cao*) = 0.2 × 0.6 = 0.12, *fm*(*khả năng cao*) = 0.2 × 0.6 = 0.12.

Vì *ít cao* < *khả năng cao* < *cao* < *hơn cao* < *rất cao* nên : *I*(*ít cao*) = [0.4, 0.52], *I*(*khả năng cao*) = [0.52, 0.64], *I*(*hơn cao*) = [0.64, 0.79], *I*(*rất cao*) = [0.79, 1].

Chọn ψ_1= 10000 ∈ $X_{Lương}$ khi đó ∀ω ∈ *Num*(*Lương*), *IC*(ω) = {0.45, 0.24, 0.52, 0.34, 0.64, 0.64, 0.79, 0.35, 1, 0.50, 0.79, 0.4, 0.55, 0.50}. Do đó Φ_2(0.45) = *ít cao* vì 0.45 ∈ *I*(*ít cao*) và tương tự cho các giá trị còn lại.

Vậy, thuộc tính *Lương* được thuần nhất theo ngôn ngữ là:

$A_{Lương}$ = {*Ít cao, Thấp, Khả năng cao, Ít thấp, Cao, Cao, Rất cao, Ít thấp, Rất cao, Khả năng cao, Rất cao, Ít thấp, Khả năng cao, Khả năng cao*} và tương ứng khi thuần nhất theo giá trị số là: $A_{Lương}$ = {450, 240, 520, 340, 640, 640, 790, 350, 1000, 500, 790, 400, 550, 500}.

Việc thuần nhất các giá trị cho thuộc tính mờ là cần thiết vì các thuật toán

huấn luyện điều cần sự thuần nhất của tập mẫu. Định nghĩa 2.4 và Mệnh đề 2.6 đã cho chúng ta một cách thuần nhất dữ liệu cho thuộc tính mờ. Tuy vậy, cách thức này đòi hỏi phải tìm được các giá trị ψ_{min}, ψ_{max} của thuộc tính mờ tương ứng A_i nên các giá trị nằm ngoài miền này vẫn chưa được xý lý.

2.4.3. Một cách định lượng giá trị ngôn ngữ ngoại lai trong tập mẫu huấn luyện

Như đã xét ở Mục 2.4.2, với bất kỳ một thuộc tính không thuần nhất A_i, ta có thể chuyển thuần nhất về giá trị số thuộc đoạn [0, 1] để từ đó chuyển về giá trị ngôn ngữ hay giá trị kinh điển tùy thuộc yêu cầu của việc sử dụng tập mẫu. Trong quá trình xây dựng các ánh xạ chuyển, việc nhận ra các giá trị biên [ψ_{min}, ψ_{max}] đối với miền trị kinh điển D_{A_i} của A_i hay [ψ_{minLV}, ψ_{maxLV}] đối với miền trị ngôn ngữ của LD_{A_i} là thực sự cần thiết.

Hầu hết các trường hợp xảy ra, các giá trị biên đã đề cập thường được lưu trữ sẵn trong tập mẫu và chúng dễ dàng xác định thông qua việc duyệt tập huấn luyện và chọn. Lúc này, các giá trị ngôn ngữ trong LD_{A_i} khi được làm rõ sẽ có giá trị nằm trong [ψ_{min}, ψ_{max}] của D_{A_i} và ngược lại. Tuy vậy, đôi khi cũng xuất hiện các giá trị nằm ngoài các khoảng xác định này, cụ thể là các giá trị ngôn ngữ của LD_{A_i} nhưng giá trị rõ không tương ứng trong miền giá trị [ψ_{min}, ψ_{max}] của D_{A_i} và ngược lại.

Định nghĩa 2.5. Cho thuộc tính không thuần nhất $A_i \in D$ có $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, $D_{A_i} = [\psi_{min}, \psi_{max}]$, $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$. Nếu $x \in LD_{A_i}$ mà $\nu(x) < IC(\psi_{min})$ hoặc $\nu(x) > IC(\psi_{max})$ thì x được gọi là giá trị ngôn ngữ ngoại lai.

Với thuộc tính không thuần nhất *Lương* trong tập mẫu huấn luyện ở Bảng 2.7, khi giá trị ngôn ngữ "*quá cao*" hay "*quá thấp*" mà ν(*quá cao*) $\notin$ [*IC*(200), *IC*(1000)] hoặc ν(*quá thấp*) $\notin$ [*IC*(200), *IC*(1000)] tức là miền giá trị [200, 1000] không phản ánh đúng các giá trị ngôn ngữ này cần thể hiện nên chúng là các giá trị ngôn ngữ ngoại lai.

Các phương pháp tiền xử lý dữ liệu truyền thống như sử dụng giá trị hằng toàn cục hay sử dụng giá trị trung bình của thuộc tính, phương pháp hồi quy,... [20], [58], [71] không thể sử dụng để xác định các giá trị ngoại lai cho thuộc tính

không thuần nhất này. Một giải pháp có thể sử dụng là hoặc bỏ qua các trường hợp ngoại lai hoặc xem chúng cùng lớp tương đương với các giá trị ngôn ngữ khác trong miền trị $[\psi_{min}, \psi_{max}]$, chẳng hạn ở trong tập mẫu trên ta có thể đồng nhất ngữ nghĩa "*quá cao*" với "*rất cao*", nhưng việc làm này sẽ làm mất thông tin vì không thể hiện đúng bản chất sự việc. Sử dụng ý kiến chuyên gia cho việc xác định giá trị rõ của các giá trị ngoại lai này không phải lúc nào cũng cho kết quả như mong muốn vì còn phụ thuộc vào trình độ của chuyên gia. Ở đây, luận án sẽ vận dụng phương pháp thuần nhất giá trị dựa đại số gia tử ở trên để đưa ra phương pháp xấp xỉ nhằm xác định giá trị rõ cho các giá trị ngôn ngữ ngoại lai này.

Theo phương pháp đã đề xuất ở Mục 2.4.2 cho việc thuần nhất các giá trị của thuộc tính không thuần nhất, ta thấy tính mờ của các giá trị trong đại số gia tử là một đoạn con của [0, 1] cho nên họ các đoạn con như vậy của các giá trị có cùng độ dài sẽ tạo thành phân hoạch của [0, 1]. Phân hoạch ứng với các giá trị có độ dài từ lớn hơn sẽ mịn hơn và khi độ dài lớn vô hạn thì độ dài của các đoạn trong phân hoạch giảm dần về 0. Hơn nữa, trong đại số gia tử thì các giá trị ngôn ngữ là một tập sắp thứ tự tuyến tính và theo Định nghĩa 2.5 thì trong trật tự sắp xếp tăng dần, các giá trị ngoại lai sẽ nằm ngoài đoạn $[\nu(\psi_{minLV}), \nu(\psi_{maxLV})]$ hiện có.

Vì vậy, một điều hoàn toàn tự nhiên là ta sẽ chia các đoạn con [0, $\nu(\psi_{minLV})$] và [$\nu(\psi_{maxLV})$, 1] tương ứng thành các phân hoạch nhỏ hơn nhằm xác định lại độ dài của các đoạn [0, $\nu(\psi_{GiáTrịNgoạiLai})$] và [$\nu(\psi_{GiáTrịNgoạiLai})$, $\nu(\psi_{minLV})$] hay [$\nu(\psi_{maxLV})$, $\nu(\psi_{GiáTrịNgoạiLai})$] và [$\nu(\psi_{GiáTrịNgoạiLai})$, 1] để từ đó có xác định giá trị rõ cho các giá trị ngôn ngữ ngoại lai này. Ta có thuật toán định lượng cho các giá trị ngôn ngữ ngoại lai như sau:

Thuật toán định lượng cho các giá trị ngôn ngữ ngoại lai.

Vào: *Thuộc tính không thuần nhất chứa giá trị ngôn ngữ ngoại lai* A_i

Ra: *Thuộc tính với miền trị được thuần nhất* A_i

Mô tả thuật toán:

Tách riêng các giá trị ngoại lai này ra khỏi A_i*, được* A'_i *;*

Thực hiện việc thuần nhất các giá trị cho A'_i *theo cách đã đề cập ở Mục* 2.4.2;

So sánh các $\psi_{GiáTrịNgoạiLai}$ *với* ψ_{Max} *và* ψ_{Min} *của* A'_i.

Thực hiện lại phân hoạch trên đoạn [0, 1] ;

If $\psi_{GiáTrịNgoạiLai} < \psi_{MinLV}$ then

Begin

Phân hoạch $[0, \nu(\psi_{MinLV})]$ *thành*

$[0, \nu(\psi_{GiáTrịNgoạiLai})]$ *và* $[\nu(\psi_{GiáTrịNgoạiLai}), \nu(\psi_{MinLV})]$;

$fm(h_{GiáTrịNgoạiLai}) \sim fm(h_{MinLV}) \times I(\psi_{MinLV})$;

$fm(h_{MinLV}) = fm(h_{MinLV}) - fm(h_{GiáTrịNgoạiLai})$;

End;

If $\psi_{GiáTrịNgoạiLai} > \psi_{MaxLV}$ then

Begin

Phân hoạch $[\nu(\psi_{MaxLV}), 1]$ *thành*

$[\nu(\psi_{MaxLV}), \nu(\psi_{GiáTrịNgoạiLai})]$ *và* $[\nu(\psi_{GiáTrịNgoạiLai}), 1]$;

$fm(h_{GiáTrịNgoạiLai}) \sim fm(h_{MaxLV}) \times I(\psi_{MaxLV})$;

$fm(h_{MaxLV}) = fm(h_{MaxLV}) - fm(h_{GiáTrịNgoạiLai})$;

End;

Dựa vào $IC(\omega)$ *của* A'_i, *tính lại* $IC(\omega)$ *cho* A_i *;*

Thuần nhất giá trị cho A_i .

Chứng minh: Thuật toán có độ phức tạp $O(n)$ vì chúng ta chỉ duyệt tuần tự thông qua các giá trị của tập dữ liệu trong tất cả các nhiệm vụ, và thực hiện ánh xạ chuyển đổi dữ liệu cho các giá trị này.

Do tất cả các phân hoạch trên không vượt ra khỏi đoạn đang xét là $|fm(h_{minLV})|$ hay $|fm(h_{maxLV})|$ nên không làm phá vỡ các phân hoạch đang có của đoạn [0, 1]. Do $I(\psi_{MinLV}) > 0$ và $I(\psi_{MaxLV}) < 1$, nên cách phân hoạch trên là phù hợp với phương pháp thuần nhất đã nêu ở Mục 2.4.2.

Vì vậy, thuật toán này xác định được giá trị rõ nằm ngoài miền giá trị đang có cho giá trị ngôn ngữ ngoại lai trong tập mẫu theo Định nghĩa 2.5.

■

Ví dụ 2.3. Cho tập mẫu huấn luyện như Bảng 2.7, hãy thuần nhất cho thuộc tính *Lương*, với $A_{Lương}$= {480, *Thấp*, 530, 200, *Cao*, *Cao*, *Rất cao*, 350,1000, 500,

Quá cao, Ít thấp, 550, 500},

Ta có: $Dom(Lương) = D_{Lương} \cup LD_{Lương}$

$D_{Lương}$ = {200, 350, 480, 500, 530, 550, 1000}; ψ_{min} = 200; ψ_{max} = 1000

$LD_{Lương}$ = {*Quá thấp, Rất thấp, Thấp, Ít thấp, Cao, Rất cao, Quá cao*}.

Trong $LD_{Lương}$, các giá trị ngôn ngữ: *Rất thấp, Thấp, Ít thấp, Cao, Rất cao* là các giá trị thông thường có miền trị nằm trong miền [ψ_{min}, ψ_{max}] đã biết còn các giá trị ngôn ngữ *Quá thấp, Quá cao* không nằm trong miền trị này, chúng là các giá trị ngoại lai của *Lương*. Lúc này, chúng ta phải định lượng cho toàn bộ các giá trị ngôn ngữ của $LD_{Lương}$ nhằm đạt tính thuần nhất của tập mẫu huấn luyện nhưng không có ý kiến của chuyên gia để xác định lại miền trị [ψ_{min}, ψ_{max}].

Đầu tiên, ta thuần nhất các giá trị cho *Lương* nhưng không xét các giá trị ngoại lai trong mô hình. Lúc này:

$D_{Lương}$ = {200, 350, 480, 500, 530, 550, 1000}; $LD_{Lương}$ = {*Thấp, Ít thấp, Cao, Rất cao*}. Xây dựng 1 ĐSGT để tính cho thuộc tính không thuần nhất *Lương* như sau: $\underline{X}_{Lương}$ = ($X_{Lương}$, $G_{Lương}$, $H_{Lương}$, ≤), với $G_{Lương}$ = {*cao, thấp*}, $H^{+}_{Lương}$ = {hơn, *rất*}, $H^{-}_{Lương}$ = {*khả năng, ít*}. Quan hệ ngữ nghĩa: *rất* > *hơn* và *ít* >*khả năng*. $W_{Lương}$ = 0.4, *fm*(*thấp*) = 0.4, *fm*(*cao*) = 0.6, μ(*rất*) = 0.35, μ(*hơn*) = 0.25, μ(*khả năng*) = 0.20, μ(*ít*) = 0.20.

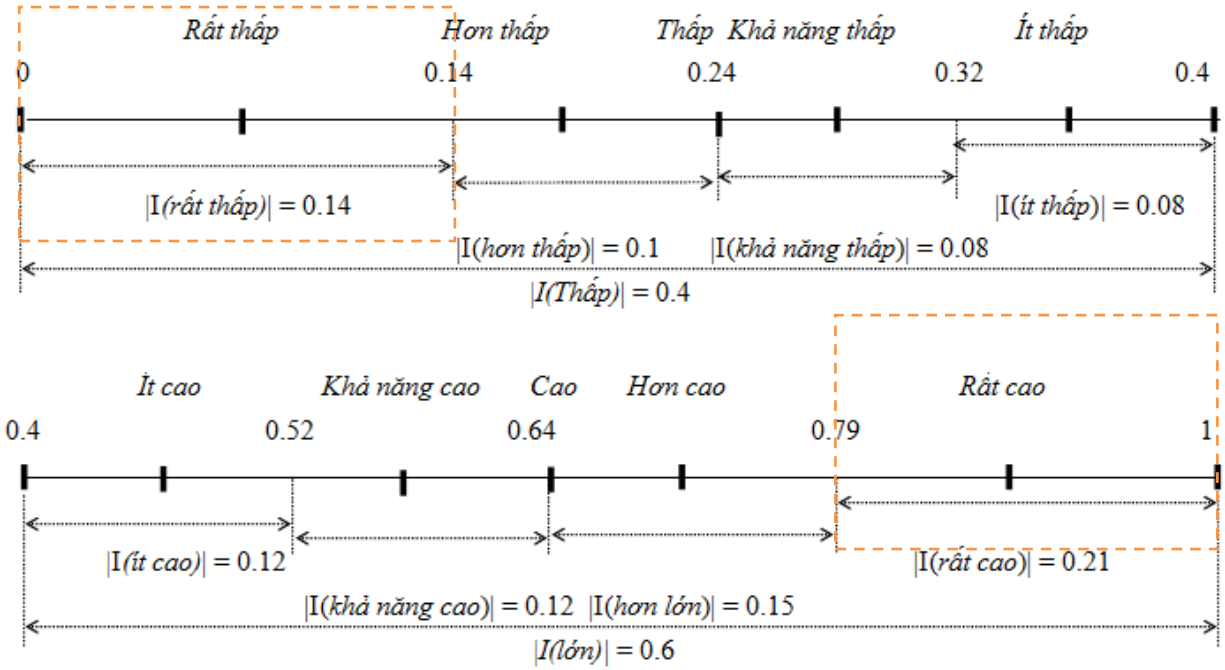
Ta có *fm*(*rất thấp*) = 0.35 × 0.4 = 0.14, *fm*(*hơn thấp*) = 0.25 × 0.4 = 0.10, *fm*(*ít thấp*) = 0.2 × 0.4 = 0.08, *fm*(*khả năng thấp*) = 0.2 × 0.4 = 0.08.

Vì *rất thấp* < *hơn thấp* < *thấp* < *khả năng thấp* < *ít thấp* nên: *I*(*rất thấp*) = [0, 0.14], *I*(*hơn thấp*) = [0.14, 0.24], *I*(*khả năng thấp*) = [0.24, 0.32], *I*(*ít thấp*) = [0.32, 0.4].

Ta có *fm*(*rất cao*) = 0.35 × 0.6 = 0.21, *fm*(*hơn cao*) = 0.25 × 0.6 = 0.15, *fm*(*ít cao*) = 0.2 × 0.6 = 0.12, *fm*(*khả năng cao*) = 0.2 × 0.6 = 0.12.

Vì *ít cao* < *khả năng cao* < *cao* < *hơn cao* < *rất cao* nên: *I*(*ít cao*) = [0.4, 0.52], *I*(*khả năng cao*) = [0.52, 0.64], *I*(*hơn cao*) = [0.64, 0.79], *I*(*rất cao*) = [0.79, 1].

Ta có hình ảnh của sự phân bố tính mờ của thuộc tính *Lương* được mô tả tóm tắt như ở Hình 2.9.



Hình 2.9. Tính mờ của thuộc tính *Lương* khi chưa xét các giá trị ngoại lai

Chọn $\psi_1 = 100 \in X_{Lương}$ khi đó $\forall \omega \in Num(Lương)$, $IC(\omega)$ = {0.35, 0.24, 0.41, 0, 0.64, 0.64, 1, 0.19, 1, 0.38, 0.4, 0.44, 0.38}.

Tiếp theo, ta phải tính cho các giá trị ngoại lai đã xác định ở trên. Ta sẽ chọn các phân hoạch thích hợp của các khoảng mờ để chèn các giá trị ngoại lai vào các khoảng này.

Do giá trị ngoại lai *quá cao* > *rất cao* nên ta sẽ phân hoạch đoạn [0.79, 1] tương ứng của |*I*(*lớn*)|. Như vậy ta có: *fm*(quá *cao*) ~ *fm*(*rất cao*) × *I*(*rất cao*) = 0.21 × 0.79 = 0.17 nên *I*(*rất cao*) = [0.79, 0.96], *I*(*quá cao*) = [0.96, 1]. *Quá thấp* < *rất thấp* nên ta sẽ phân hoạch đoạn [0, 0.14] tương ứng của |*I*(*thấp*)|.

Nên *fm*(*quá thấp*) ~ *fm*(*rất thấp*) × *I*(*rất thấp*) = 0.14 × 0.14 = 0.02 nên *I*(*rất thấp*) = [0.02, 0.14], *I*(*quá cao*) = [0, 0.02]. Thuộc tính *Lương* nhận được như sau: $A_{Lương}$= {480, *Thấp*, 503, 200, *Cao*, *Cao*, *Rất cao*, 350, 1000, 500, *Quá cao*, *Ít thấp*, 550, 500, *Quá thấp*}, $IC(\omega)$ = {0.35, 0.24, 0.41, 0.02, 0.64, 0.64, 0.79, 0.19, 0.79, 0.38, 1, 0.4, 0.44, 0.38, 0}. Do đó, $\Phi_2(0.35)$ = *ít thấp* vì $0.35 \in$ *I*(*ít thấp*).

Tương tự cho các giá trị còn lại, ta có thuộc tính *Lương* theo ngôn ngữ sẽ là: {*Ít thấp, Thấp, Ít cao, Rất thấp, Cao, Cao, Rất cao, Hơn thấp, Rất cao, Ít thấp, Quá cao, Ít thấp, Ít cao, Ít thấp, Quá thấp*} tương ứng với thuần nhất theo giá trị

là: {480, 240, 530, 200, 640, 640, 1000, 350, 1000, 500, 1300, 500, 400, 550, 500, 120}.

2.4.4. Thuật toán học cây quyết định mờ FMixC4.5 dựa trên đối sánh điểm mờ

a. Tư tưởng thuật toán: như đã phân tích và đánh giá ở Mục 2.3.3 của luận án, mặc dầu thuật toán MixC4.5 đã giải quyết được một phần hạn chế của các thuật toán truyền thống như C4.5, SLIQ, SPRINT nhưng với tập huấn luyện có chứa thuộc tính mờ thì hiệu quả của MixC4.5 vẫn chưa cao. Theo mô hình đã đề xuất ở Mục 2.4.1 của luận án, chúng ta cần kiểm tra sự hiện diện của thuộc tính mờ trong tập huấn luyện để có phương pháp huấn luyện phù hợp, điều này tránh được các tính toán không cần thiết trong trường hợp tập huấn luyện không chứa thuộc tính mờ.

Với các giải pháp đã đề xuất ở Mục 2.4.2 và Mục 2.4.3 của luận án, chúng ta sẽ lần lượt kiểm duyệt các thuộc tính mờ và xây dựng các đại số gia tử tương ứng nhằm thuần nhất dữ liệu cho nó nhằm phục vụ cho giai đoạn huấn luyện.

b. Thuật toán FMixC4.5: với tập mẫu dữ liệu nghiệp vụ huấn luyện *D* có *m* thuộc tính, chứa thông tin mờ chưa thuần nhất, ta có thuật toán FMixC4.5 được cải tiến từ MixC4.5 cho việc học cây quyết định *S* như sau:

Thuật toán FMixC4.5

Vào: *mẫu D có n bộ, m thuộc tính dự đoán và thuộc tính quyết định Y.*

Ra: *Cây quyết định S.*

Mô tả thuật toán:

ChonMauDacTrung(D);

If *(tập huấn luyện không có thuộc tính mờ)* then Call *thuật toán MixC4.5;*

Else

Begin

For each *(thuộc tính mờ X của D)* do

Begin

Xây dựng đại số gia tử $\underline{X}_k$ *tương ứng với thuộc tính mờ X;*

Kiểm tra và tách các giá trị ngoại lai;

Chuyển giá trị số, giá trị ngôn ngữ của X về giá trị thuộc đoạn $\subseteq$ [0, 1];

Xử lý các giá trị ngoại lai;

End*;*

Call *thuật toán MixC4.5 ;*

End;

Với *m* là số thuộc tính, *n* là số thể hiện của tập huấn luyện, độ phức tạp của MixC4.5 là $O(m \times n^2 \times log\ n)$*;* Với FMixC4.5, trước tiên ta mất $O(m \times n)$ để kiểm tra sự xuất hiện của các thuộc tính mờ và tuần tự sau đó là $O(n)$ cho mỗi tiến trình xử lý trên từng thuộc tính mờ. Vậy độ phức tạp của FMixC4.5 cũng là $O(m \times n^2 \times log\ n)$.

Tính đúng của giải thuật FMixC4.5 được chứng minh thông qua thuật toán MixC4.5 và các Mục 2.4.2 và Mục 2.4.3 của luận án.

■

2.4.5. Cài đặt thử nghiệm và đánh giá thuật toán FMixC4.5

Chương trình thực nghiệm được cài đặt bằng ngôn ngữ Java Eclipse Mars Release 4.50 trên máy tính có cấu hình: Processor: Intel® Core™i5-2450 CPU @ 2.50GHz (4CPUs), ~ 2.50 GHz, RAM4GB, System type 64bit.

Với cơ sở dữ liệu Mushroom có hơn 8000 mẫu tin gồm 24 thuộc tính, trong đó có 2 thuộc tính mờ không thuần nhất đã mô tả ở ở Bảng 2.6. Tương tự khi đối sánh với MixC4.5 và các thuật toán truyền thống, luận án tách riêng biệt 5000 mẫu tin cho tập huấn luyện và dùng 3000 mẫu còn lại để chọn ngẫu nhiên 2000 mẫu dùng cho việc kiểm tra. Kết quả thực nghiệm của FMixC4.5 so với MixC4.5 và thuật toán truyền thống C4.5 được thể hiện ở Bảng 2.8.

Bảng 2.8. Bảng so sánh kết quả kiểm tra độ chính xác của thuật toán FMixC4.5 trên cơ sở dữ liệu có chứa thuộc tính mờ Mushroom

Thuật toán	**Thời gian huấn luyện (s)**	**Số lượng mẫu và độ chính xác dự đoán (%)**				
		100	**500**	**1000**	**1500**	**2000**
C4.5	18.9	57.0	51.2	54.8	66.2	70.0
MixC4.5	50.2	58.8	54.6	54.8	66.2	70.0
FMixC4.5	58.2	71.0	72.2	72.6	77.9	77.2

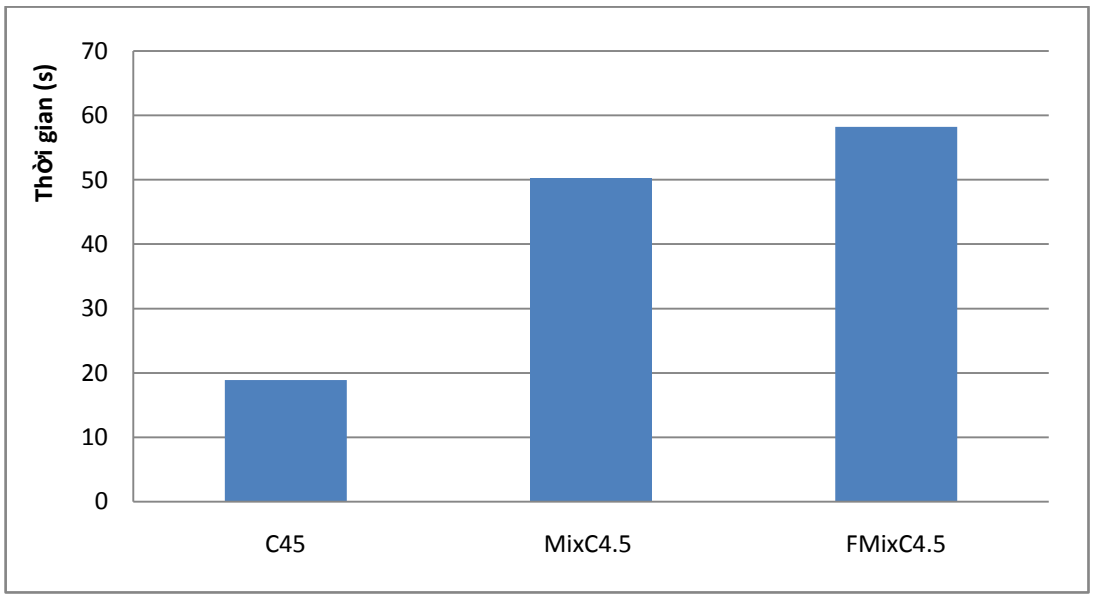
Bảng 2.9. Bảng so sánh thời gian kiểm tra của thuật toán FMixC4.5 trên cơ sở dữ liệu có chứa thuộc tính mờ Mushroom

Thuật toán	**Số lượng mẫu và thời gian thực hiện dự đoán (s)**				
	100	**500**	**1000**	**1500**	**2000**
C4.5	0.2	0.7	1.6	2.1	2.9
MixC4.5	0.2	0.8	1.7	2.2	3.0
FMixC4.5	**0.4**	**1.0**	**1.9**	**2.8**	**3.8**

Từ kết quả thu được tại các Bảng 2.8 và Bảng 2.9, chúng ta có các nhận xét, đánh giá như sau:

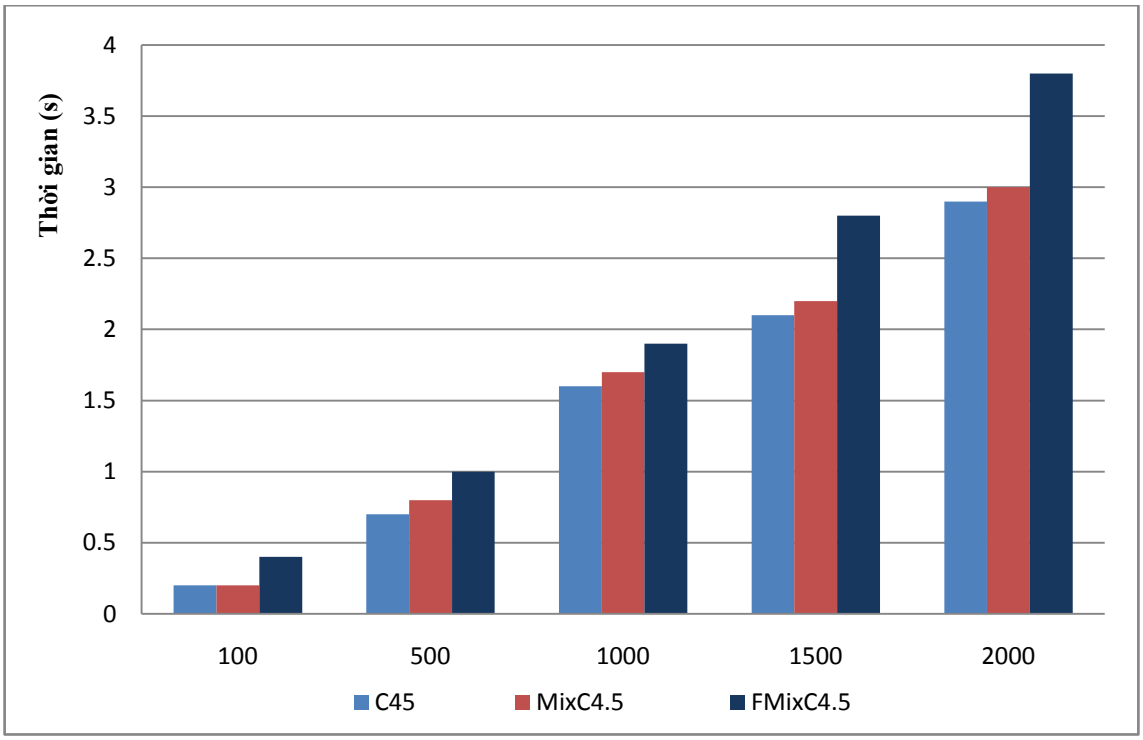
♦ **Chi phí Thời gian:** mặc dầu có cùng độ thức tạp nhưng thuật toán MixC4.5 luôn có thời gian thực hiện tốt hơn FMixC4.5 trong cả giai đoạn huấn luyện, Hình 2.10 và quá trình dự đoán, Hình 2.11.

Do thuật toán MixC4.5 và các thuật toán truyền thống khác như C4.5 đã bỏ qua các giá trị mờ trong tập mẫu, không phải mất thời gian xử lý trong cả hai quá trình huấn luyện và dự đoán nên thời gian thực hiện sẽ nhanh hơn thuật toán FMixC4.5 được đề xuất.



Hình 2.10. So sánh thời gian huấn luyện với 5000 mẫu Mushroom của FMixC4.5 với các thuật toán khác

Đối với thuật toán FMixC4.5, vì phải trải qua quá trình xây dựng các đại số gia tử cho các trường mờ và thời gian để thuần nhất các giá trị mờ, xử lý giá trị ngoại lai trong cả hai giai đoạn huấn luyện và dự đoán nên thời gian thực hiện của FMixC4.5 luôn nhiều hơn MixC4.5 và các thuật toán truyền thống khác.

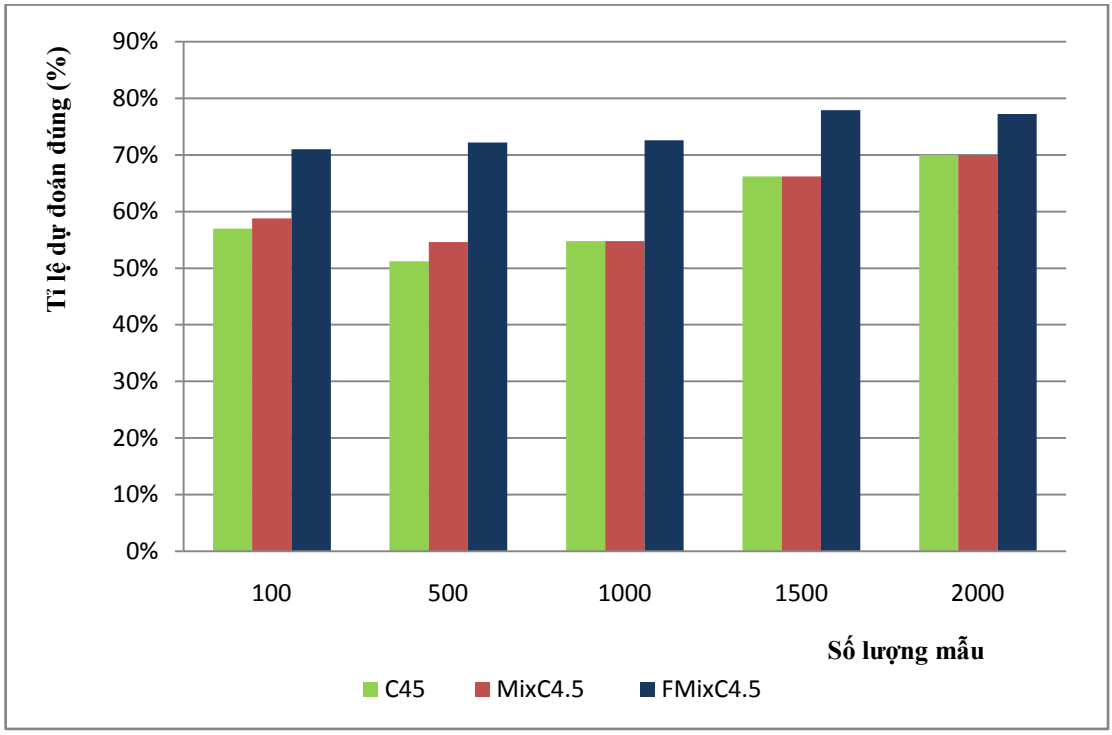


Hình 2.11. So sánh thời gian kiểm tra với 2000 mẫu Mushroom của FMixC4.5 với các thuật toán khác

♦ **Kết quả dự đoán:** vì MixC4.5 và các thuật toán truyền thống luôn bỏ qua các giá trị mờ trong tập mẫu, chỉ quan tâm các giá trị rõ nên làm mất dữ liệu tại các trường mờ. Do đó, kết quả dự đoán không cao vì không thể dự đoán hiệu quả cho các trường hợp xuất hiện giá trị mờ. Việc xây dựng một đại số gia tử tại các trường mờ, và dùng nó để thuần nhất tập mẫu cho chúng ta tập huấn luyện thuần nhất chứa cả dữ liệu rõ và mờ, nên kết quả của cây được huấn luyện bằng FMixC4.5 sẽ tốt hơn, vì thế kết quả dự đoán có tỷ lệ đúng cao hơn khi sử dụng C4.5 và MixC4.5, Hình 2.12.

Trong bài toán phân lớp, hiệu quả trong quá trình dự đoán là yếu tố quyết định. Việc thuật toán FMixC4.5 có thời gian thực hiện cao hơn các thuật toán truyền thống nhưng kết quả dự đoán tốt hơn là yếu tố tích cực. Hơn nữa, việc

huấn luyện thông thường có tần suất sử dụng ít hơn quá trình dự đoán vì mô hình cây sau khi huấn luyện sẽ sử dụng nhiều lần.



Hình 2.12. So sánh tỷ lệ đúng trên cây kết quả của FMixC4.5 với các thuật toán khác của bộ dữ liệu Mushroom

2.5. Kết luận Chương 2

Với mục tiêu khắc phục các hạn chế của các thuật toán học cây quyết định truyền thống như C4.5, SPLIQ, SPRINT trên các tập huấn luyện có chứa thuộc tính mờ, chương này của luận án tập trung:

1. Phân tích mối tương quan giữa các thuật toán học cây quyết định nền tảng và phân tích sự ảnh hưởng của tập mẫu huấn luyện đối với hiệu quả cây kết quả thu được, trình bày một phương pháp nhằm trích chọn được tập mẫu huấn luyện đặc trưng phục vụ cho quá trình huấn luyện và đề xuất thuật toán MixC4.5 phục vụ quá trình học.

2. Nhận thấy việc xây dựng mô hình cây quyết định hoặc cây quyết định mờ phụ thuộc vào mục đích và tập mẫu huấn luyện, luận án đã đưa ra mô hình học nhằm đáp ứng yêu cầu cho yêu cầu này. Đồng thời luận án cũng phân tích,

đưa ra các khái niệm về tập mẫu không thuần nhất, giá trị ngoại lai và xây dựng thuật toán để có thể thuần nhất cho các thuộc tính có chứa các giá trị này.

3. Trên cơ sở phân tích, chỉ ra cách thuần nhất cho các thuộc tính không thuần nhất của tập mẫu và khái niệm cùng cách thức xử lý giá trị ngoại lai, chương này của luận án cũng đã xây dựng thuật toán FMixC4.5 nhằm phục vụ cho quá trình học xây dựng cây quyết định trên tập huấn luyện này. Các kết quả thực nghiệm được đối sánh đã cho thấy khả năng dự đoán của MixC4.5, FMixC4.5 tốt hơn các thuật toán truyền thống khác.

Chương 3.
PHƯƠNG PHÁP HUẤN LUYỆN CÂY QUYẾT ĐỊNH MỜ CHO BÀI TOÁN PHÂN LỚP DỮ LIỆU DỰA TRÊN ĐỐI SÁNH KHOẢNG MỜ

3.1. Giới thiệu

Với mục tiêu xây dựng được cây quyết định S đạt hiệu quả cao cho quá trình phân lớp, tức $f_h(S) \rightarrow max$ trên tập huấn luyện D, Chương 2 của luận án đã tập trung giải quyết những hạn chế của các phương pháp học truyền thống bằng cách đưa ra thuật toán học MixC4.5 và FMixC4.5. Mặc dầu MixC4.5 tỏ ra hiệu quả hơn so với các cách học truyền thống như C4.5, SLIQ, SPRINT, nhưng khi tập huấn luyện D có thuộc tính mờ thì tính hiệu quả dự đoán của nó vẫn chưa đạt kỳ vọng mong muốn. Đối sánh dữ liệu ở các Bảng 2.4 và Bảng 2.6 cho thấy với mẫu dự đoán chứa giá trị mờ thì khả năng dự đoán đúng chỉ đạt xấp xĩ trên 50% và thấp hơn nhiều khi tập mẫu không chứa giá trị mờ.

Theo cách tiếp cận của ĐSGT, dựa vào tính có thứ tự của các giá trị ngôn ngữ trong tập huấn luyện, thuật toán FMixC4.5 đã tiến hành thuần nhất các giá trị của thuộc tính mờ trong tập huấn luyện nên cây kết quả thu được cho kết quả dự đoán khả quan hơn các phương pháp học truyền thống khác, Hình 2.12. Tuy vậy, do quá trình thuần nhất giá trị ngôn ngữ LD_{A_i}và giá trị số của D_{A_i} của thuộc tính mờ A_i về các giá trị trong đoạn [0, 1] làm xuất hiện các sai số, vì có thể có nhiều giá trị kinh điển gần nhau được quy về một điểm trong đoạn [0, 1] nên kết quả dự đoán của FMixC4.5, Hình 2.12, Bảng 2.8 vẫn chưa thật sự đáp ứng kỳ vọng. Tỷ lệ chính xác trung bình của quá trình dự đoán không có thuộc tính mờ chiếm khoảng 80%, Bảng 2.4, nhưng khi dữ liệu có chứa thuộc tính mờ thì kết quả trung bình dự đoán hiện chúng ta có được với FMixC4.5 chỉ xấp xĩ dưới 75%, Bảng 2.8.

Thêm vào đó, với mục tiêu đã được đặt ra tại (1.13) thì hàm mục tiêu $f_h(S) \rightarrow max$ còn bao hàm sự linh hoạt trong quá trình dự đoán, tức có khả năng dự

đoán cho nhiều tình huống khác nhau. Như đã chỉ ra ở Hình 1.8, do sự phân tách tại các thuộc tính mờ trong mô hình cây kết quả theo các điểm phân chia, nên khó khăn trong trường hợp cần dự đoán cho các giá trị khoảng có miền trị đan xen giữa hai nhánh của cây được huấn luyện. Khi nghiên cứu các giá trị mờ, chúng ta thấy chúng được biểu diễn thành từng khoảng mờ và là các đoạn con của [0, 1]. Như vậy, khi gặp các khó khăn về sự phân lớp dựa trên việc chia tách theo điểm mờ của bài toán cần giải quyết, một cách tự nhiên đó là chúng ta nghĩ đến việc phân chia theo từng khoảng mờ của chúng.

Theo cách tiếp cận dựa trên khoảng mờ, chương này của luận án tập trung nghiên cứu mối tương quan của các khoảng mờ, đề xuất phương pháp đối sánh dựa trên khoảng mờ và xây dựng thuật toán học phân lớp dựa trên khoảng mờ HAC4.5 nhằm thu được cây kết quả có khả năng dự đoán cao. Bên cạnh đó, nhận thấy trong quá trình huấn luyện mô hình cây quyết định S từ tập mẫu huấn luyện D, cần phải đồng thời đạt được các mục tiêu đó là tính hiệu quả của quá trình phân lớp và tính đơn giản và dễ hiểu đối với người dùng, tức phải đồng thời đạt được mục tiêu của các hàm $f_h(S) \rightarrow max$ và $f_n(S) \rightarrow min$. Luận án đã đề xuất khái niệm về khoảng mờ lớn nhất và cải tiến thuật toán HAC4.5 thành thuật toán HAC4.5* nhằm đáp ứng cho các mục tiêu đã đặt ra.

Các thuật toán HAC4.5 và HAC4.5* đề xuất trong luận án được cài đặt thử nghiệm, đánh giá dựa trên các bộ dữ liệu mẫu Mushroom, Adult nhằm đối sánh với các thuật toán truyền thống và các công trình liên quan khác. Đồng thời các kết quả này cũng đã được công bố ở các công trình [CT4], [CT5] và [CT7] của luận án.

3.2. Phương pháp đối sánh giá trị khoảng trên thuộc tính mờ

3.2.1. Xây dựng cách thức đối sánh giá trị khoảng dựa trên đại số gia tử

Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$ và $x, y \in X$, ta có mối tương quan của các khoảng mờ x, y như đã đề cập ở Mục 1.2.4 của luận án. Với một giá trị khoảng $[a, b]$ bất kỳ, để xét mối tương quan một giá trị $x \in X$ với $[a, b]$, trước hết ta phải chuyển $[a, b]$ về đoạn con của [0,1].

Định nghĩa 3.1. Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$. Với $x \in X$, $I(x) \subseteq [0, 1]$, $|I(x)| = fm(x)$ và một giá trị khoảng $[a, b]$. Hàm f được gọi là hàm biến đổi về đoạn [0,

1] khi tương ứng, hàm f sẽ chuyển đổi giá trị khoảng $[a, b]$ về một đoạn con của [0, 1] và ta đặt $[I_a, I_b] = [f(a), f(b)] \subseteq [0, 1]$.

Vì tính mờ của x và $[I_a, I_b]$ là một đoạn con của [0,1], do đó để xét mối tương quan $x \in X$ và $[a, b]$, theo Định nghĩa 1.19, chúng ta chỉ cần xét dựa vào phần giao của hai đoạn con của [0,1] này là x và $[I_a, I_b]$.

Như đã trình bày ở Mục 1.3.2 của luận án, các thuật toán phân lớp dữ liệu bằng cây quyết định phải tính và lựa chọn thuộc tính có lợi ích thông tin lớn nhất để phân tách cây. Khi tính lợi ích thông tin cho thuộc tính, cần so sánh các giá trị trong thuộc tính đang xét để phân lớp theo quan hệ thứ tự, nhằm tránh sự phân chia thuần túy theo chiều ngang. Do vậy, trước khi nghiên cứu sự phân lớp theo các khoảng giá trị, chúng ta cần phải xây dựng một mối quan hệ thứ tự cho các khoảng này.

Định nghĩa 3.2. Cho 2 khoảng rõ khác nhau $[a_1, b_1]$ và $[a_2, b_2]$ tương ứng với các khoảng mờ $[I_{a_1}, I_{b_1}]$, $[I_{a_2}, I_{b_2}] \subseteq [0, 1]$. Ta nói khoảng $[a_1, b_1]$ đứng trước $[a_2, b_2]$ hay $[a_2, b_2]$ đứng sau $[a_1, b_1]$, *viết* $[a_1, b_1] < [a_2, b_2]$ hay $[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}]$ nếu:

1. $b_2 > b_1$ tức $I_{b_2} > I_{b_1}$,
2. *Nếu* $I_{b_2} = I_{b_1}$ tức $b_2 = b_1$ thì $I_{a_2} > I_{a_1}$ tức $a_2 > a_1$.

và lúc này ta nói dãy $[a_1, b_1]$, $[a_2, b_2]$ là dãy 2 khoảng có quan hệ thứ tự trước sau.

Định lý 3.1. Cho k khoảng khác nhau từng đôi một $[a_1, b_1]$, $[a_2, b_2]$, ... ,$[a_k, b_k]$, ta luôn sắp để được một dãy có k khoảng với quan hệ thứ tự trước sau.

Chứng minh: Thật vậy, với k khoảng khác nhau từng đôi một $[a_1, b_1]$, $[a_2, b_2]$, ... ,$[a_k, b_k]$, ta luôn tìm được khoảng trước nhất của dãy là $[a_i, b_i]$, với $a_i = min(a_1, a_2,..., a_n)$; nếu có nhiều khoảng $[a_j, b_j]$, $i = 1..k$ mà $a_j = a_i$ thì chọn ta khoảng $[a_i, b_i]$ là khoảng có b_i bé nhất trong các giá trị b_j. Việc chọn b_i luôn tìm được duy nhất vì các khoảng đã cho khác nhau từng đôi một nên nếu $a_i = a_j$ thì $b_i \neq b_j$ (Định nghĩa 1.18).

Sau khi tìm được khoảng trước nhất của dãy là $[a_i, b_i]$, ta tiếp tục tìm khoảng thứ 2 và cứ thế tiếp tục. Do các phần tử của dãy là hữu hạn nên sau k lần tìm và sắp, ta có được dãy gồm k khoảng mà các phần tử của dãy đã được sắp

theo quan hệ thứ tự trước sau.

■

Cho bài toán học phân lớp dữ liệu bằng cây quyết định mờ $S : D \rightarrow Y$ với $D = A_1 \times ... \times A_m$ ở (1.4). Khi thuộc tính A_i có miền trị $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, Chương 2 của luận án đã chỉ ra phương pháp để thuần nhất các giá trị của D_{A_i} và LD_{A_i} về đoạn [0, 1] và Mục 3.2.1 của luận án cũng đã chỉ ra cách thức để có thể đối sánh trên các khoảng này. Tuy vậy, theo Định nghĩa 2.5, bài toán chỉ được giải quyết tức Mệnh đề 2.6 chỉ đúng khi chúng ta biết giá trị ψ_{min}, ψ_{max} *của* miền trị kinh điển D_{A_i} tương ứng với A_i. Tuy nhiên trong thực tế, nhiều lúc ta không biết cụ thể giá trị ψ_{min}, ψ_{max} của thuộc tính đang xét mà chỉ biết đoạn con $[\psi_1, \psi_2]$ của chúng. Vậy, cần xây dựng một phương pháp để có thể xác định giá trị khoảng cho toàn bộ các giá trị ngôn ngữ của thuộc tính trước khi thực hiện đối sánh, nhằm huấn luyện cây đạt hiệu quả.

3.2.2. Phương pháp định lượng khoảng mờ khi chưa biết miền trị *MIN, MAX* của các thuộc tính mờ

Định nghĩa 3.3. Cho thuộc tính không thuần nhất A_i, có $Dom(A_i) = D_{A_i} \cup LD_{A_i}$, $D_{A_i} = [\psi_1, \psi_2]$ và $LD_{A_i} = [\psi_{minLV}, \psi_{maxLV}]$. A_i được gọi là thuộc tính mờ không thuần nhất chưa xác định *Min - Max* khi $\psi_{minLV} < \psi_{LV1}, \psi_{LV2} < \psi_{maxLV}$ mà $v(\psi_{LV1}) = IC(\psi_1)$ và $v(\psi_{LV2}) = IC(\psi_2)$.

Với A_i là thuộc tính mờ không thuần nhất chưa xác định *Min - Max*, lúc này ta phải tìm các giá trị $IC(\omega_i)$ còn lại, tức các $IC(\omega_i)$ thỏa $IC(\psi_i) < IC(\psi_1)$ hoặc $IC(\psi_i) > IC(\psi_2)$.

Theo Định nghĩa 2.5, do $IC(\omega_i) = 1 - \frac{\psi_{max} - \omega_i}{\psi_{max} - \psi_{min}}$ nên tất cả các ω_i nằm giữa $[\psi_1, \psi_2]$ sẽ đúng với quy tắc này, tức là $IC(\omega_i) = 1 - \frac{\psi_2 - \omega_i}{\psi_2 - \psi_1}$ với $\omega_i \in [\psi_1, \psi_2]$. Do vậy, ta có thể xây dựng một ĐSGT để định lượng giá trị cho các giá trị tương ứng trong khoảng này.

Theo phương pháp xây dựng ĐSGT, ta thấy tính mờ của các giá trị trong ĐSGT là một đoạn con của [0,1] cho nên họ các đoạn con như vậy của các giá trị có cùng độ dài sẽ tạo thành phân hoạch của [0,1]. Phân hoạch ứng với các giá trị có độ dài từ lớn hơn sẽ mịn hơn và khi độ dài lớn vô hạn thì độ dài của các đoạn

trong phân hoạch giảm dần về 0. Hơn nữa, các giá trị ngôn ngữ là một tập sắp thứ tự tuyến tính nên ta sẽ chia các đoạn con tương ứng thành các phân hoạch nhỏ hơn nhằm xác định lại độ dài của các đoạn [0, $\nu(\psi_i)$] hay [$\nu(\psi_i)$, 1] để từ đó có xác định giá trị rõ cho các giá trị ngôn ngữ này. Đây chính là điểm để tính các $IC(\omega)$ không nằm trong đoạn [ψ_1, ψ_2] bằng cách phân chia liên tiếp các đoạn con này để xác định các $IC(\omega_i)$ tương ứng.

Thêm vào đó, do độ lớn của các ω_i sẽ tỷ lệ với bán kính $f(H(x)) \subseteq [0,1]$ tức là:

1. $\omega_1 > \omega_2$ khi $IC(\omega_1) > IC(\omega_2)$

2. $\frac{\omega_1}{IC(\omega_1)} = \frac{\omega_2}{IC(\omega_2)}$ khi tất cả các $IC(\omega_1)$, $IC(\omega_2)$ đều nằm về cùng một phía với W.

Do vậy, ta có thể xác định giá trị định lượng cho các giá trị ngôn ngữ theo vết dầu loan về hai phía của đoạn [ψ_1, ψ_2] tại bước thứ i như sau:

- Với ω_i mà giá trị ngôn ngữ tương ứng trong đoạn [ψ_{LV2}, ψ_{maxLV}], ta tính tuần tự tăng theo đoạn [ψ_{LV2}, ψ_{maxLV}], với:

$$\omega_i = \nu(\psi_2) \times \frac{IC(\omega_2)}{IC(\omega_i)} \quad (3.1)$$

đồng thời dịch chuyển vị trí ψ_{LV2} và ψ_2 đến vị trí i vừa tìm được.

- Với ω_i mà giá trị ngôn ngữ tương ứng trong đoạn [ψ_{minLV}, ψ_{LV1}], ta tính tuần tự giảm theo đoạn [ψ_{LV1}, ψ_{minLV}], với:

$$\omega_i = \nu(\psi_1) \times \frac{IC(\omega_1)}{IC(\omega_i)} \quad (3.2)$$

đồng thời dịch chuyển vị trí ψ_{LV1}và ψ_1 lùi về vị trí i vừa tìm được.

Vậy, thuật toán xác định khoảng mờ cho thuộc tính không thuần nhất, chưa xác định *Min-Max* như sau:

Thuật toán định lượng khoản mờ cho thuộc tính mờ không thuần nhất, chưa xác định miền trị Min-Max.

Vào: *Thuộc tính không thuần nhất, chưa xác định Min-Max* A_i

Ra: *Thuộc tính với miền trị được thuần nhất theo khoảng mờ* A_i

Mô tả thuật toán:

Xây dựng ĐSGT trong miền [ψ_1, ψ_2];

Tính các IC(ω_i) *tương ứng cho các giá trị trong đoạn* [ψ_1, ψ_2];

For each ($v(\psi_{LV_i}) \notin [IC(\psi_1), IC(\psi_2)]$) do

Begin

If $v(\psi_{LV_i}) < IC(\psi_1)$ then

Begin

Phân hoạch [0, $v(\psi_1)$] *thành* [0, $v(\psi_i)$] và [$v(\psi_i)$, $v(\psi_1)$];

Tính $fm(h_i) \sim fm(h_1) \times I(\psi_1)$;

$fm(h_1) = fm(h_1) - fm(h_i)$;

$\omega_i = v(\psi_1) \times \frac{IC(\omega_1)}{IC(\omega_i)}$;

Tính $IC(\omega_i)$;

Gán vị trí ψ_i *thành vị trí* ψ_1;

End;

If $v(\psi_{LV_i}) > IC(\psi_2)$ then

Begin

Phân hoạch [$v(\psi_2)$, 1] *thành* [$v(\psi_2)$, $v(\psi_i)$] và [$v(\psi_i)$, 1];

Tính $fm(h_i) \sim fm(h_2) \times I(\psi_2)$;

$fm(h_2) = fm(h_2) - fm(h_i)$;

$\omega_i = v(\psi_2) \times \frac{IC(\omega_2)}{IC(\omega_i)}$;

Tính $IC(\omega_i)$;

Gán vị trí ψ_i *thành vị trí* ψ_2;

End;

End;

Tính đúng của thuật toán: dễ dàng tính được thuật toán thực hiện với độ phức tạp là $O(n)$. Do tất cả các phân hoạch trên không vượt ra khỏi đoạn đang xét là $|fm(h_1)|$ hay $|fm(h_2)|$ nên không làm phá vỡ các phân hoạch đang có của đoạn [0, 1]. Do $I(\psi_1) > 0$ và $I(\psi_2) < 1$ nên cách phân hoạch trên là phù hợp với phương pháp thuần nhất đã nêu ở Chương 2 của luận án.

■

Ví dụ 3.1. Cho tập mẫu huấn luyện như ở Bảng 3.1, ta cần định lượng cho các giá trị ngôn ngữ ở thuộc tính *Lương* với ngữ nghĩa [*Thấp, Cao*] tương ứng với miền trị xác định trong tập mẫu là [300, 800].

Bảng 3.1. Tập mẫu huấn luyện chứa thuộc tính *Lương* không thuần nhất, chưa xác định *Min - Max*

NơiSống	NgànhHọc	KinhTếGĐ	Lương	MáyTính
T.Phố	Luật	Chưa tốt	480	Không
NôngThôn	Luật	Chưa tốt	Thấp	Không
T.Phố	CNTT	Chưa tốt	530	Có
T.Phố	LịchSử	Trung bình	Rất thấp	Có
T.Phố	LịchSử	Khá	Cao	Có
NôngThôn	LịchSử	Khá	800	Không
NôngThôn	CNTT	Khá	Rất cao	Có
T.Phố	Luật	Trung bình	300	Không
T.Phố	Luật	Khá	800	Có
T.Phố	LịchSử	Trung bình	500	Có
NôngThôn	Luật	Trung bình	Rất cao	Có
NôngThôn	CNTT	Trung bình	Ít thấp	Có
T.Phố	CNTT	Chưa tốt	550	Có
NôngThôn	LịchSử	Trung bình	500	Không

Tập mẫu có thuộc tính *Lương* là chưa thuần nhất nên ta phải thuần nhất các giá trị cho *Lương*. Ta *có Dom*(*Lương*) = $D_{Lương} \cup LD_{Lương}$. $D_{Lương}$ = {300, 480, 500, 530, 550, 800}*;* $\psi_1 = 300$; $\psi_2 = 800$. $LD_{Lương}$ = {*Rất thấp, Thấp, Ít thấp, Cao, Rất cao*}. Các giá trị ngôn ngữ trong miền trị kinh điển [300, 800] là [*Thấp, Cao*].

1. Tính các giá trị *IC*(ω) trong *Lương* tương ứng trong đoạn [ψ_1, ψ_2] = [300, 800]. Lúc này: $D_{Lương}$={300, 480, 500, 530, 550, 800}*;* $LD_{Lương}$ = {*Thấp, Ít thấp, Cao*}. Xây dựng ĐSGT để tính cho thuộc tính *Lương* như sau:

$\underline{X}_{Lương}$ = ($X_{Lương}$, $G_{Lương}$, $H_{Lương}$, ≤), với $G_{Lương}$ = {*cao, thấp*}, $H^+_{Lương}$ =

{*hơn*, *rất*}, $H^-_{\text{Lương}}$ = {*khả năng*, *ít*}. Quan hệ ngữ nghĩa: *rất* > *hơn* và *ít* >*khả năng*. $W_{Lương}$ = 0.4, *fm*(*thấp*) = 0.4, *fm*(*cao*) = 0.6, μ(*rất*) = 0.35, μ(*hơn*) = 0.25, μ(*khả năng*) = 0.20, μ(*ít*) = 0.20.

Lúc này ta có: *fm*(*rất thấp*) = 0.35 × 0.4 = 0.14, *fm*(*hơn thấp*) = 0.25 × 0.4 = 0.10, *fm*(*ít thấp*) = 0.2 × 0.4=0.08, *fm*(*khả năng thấp*) = 0.2 × 0.4 = 0.08. Vì *rất thấp* < *hơn thấp* < *thấp* < *khả năng thấp* < *ít thấp* nên: *I*(*rất thấp*) = [0, 0.14], *I*(*hơn thấp*) = [0.14, 0.24], *I*(*khả năng thấp*) = [0.24, 0.32], *I*(*ít thấp*) = [0.32, 0.4].

Tương tự *fm*(*rất cao*) = 0.35 × 0.6 = 0.21, *fm*(*hơn cao*) = 025 × 0.6 = 0.15, *fm*(*ít cao*) = 0.2 × 0.6 = 0.12, *fm*(*khả năng cao*) = 0.2 × 0.6 = 0.12. Vì *ít cao* < *khả năng cao* < *cao* < *hơn cao* < *rất cao* nên: *I*(*ít cao*) = [0.4, 0.52], *I*(*khả năng cao*) = [0.52, 0.64], *I*(*hơn cao*) = [0.64, 0.79], *I*(*rất cao*) = [0.79, 1].

DOM(Lương)= {480, Thấp, 530, Rất thấp, Cao, 800, Rất cao, 300, 800, 500, Rất cao, Ít thấp, 550, 500}, Chọn ψ_1 = 800 $\in$ $X_{Lương}$ khi đó $\forall\omega \in$ *Num*(*Lương*), *IC*(ω) = {0.36, 0.24, 0.46, _, 0.64, 1, _, 0, 1, 0.40, _, 0.32, 0.50, 0.40}.

2. Tính cho các giá trị ngoài khoảng bằng cách tìm các phân hoạch thích hợp của các khoảng mờ để chèn các giá trị ngoại lai vào các khoảng này.

Do giá trị *Rất cao* > *Hơn cao* nên ta sẽ phân hoạch đoạn [0.79, 1] tương ứng của |*I(lớn)*|.

Như vậy ta có: *fm*(Rất *cao*) ~ *fm*(*Hơn cao*) × *I*(*Hơn cao*) = 0.21 × 0.79 = 0.17. Nên *I*(*Hơn cao*) = [0.79, 0.96], *I*(*Rất cao*) = [0.96, 1]. Do đó $\psi_{Rất\ \text{cao}}$ = 970.

Vì *Rất thấp* < *Hơn thấp* nên ta sẽ phân hoạch đoạn [0, 0.14] tương ứng của |*I(thấp)*|. *fm*(Rất *thấp*) ~ *fm*(*Hơn thấp*) × *I*(*Hơn thấp*) = 0.14 × 0.14 = 0.02. Nên *I*(*Hơn thấp*) = [0.02, 0.14], *I*(*Rất thấp*) = [0, 0.02]. Do đó $\psi_{Rất\ \text{thấp}}$ = 40.

3. Tính lại *IC*(ω) với [ψ_1, ψ_2] = [4, 97]. Lúc này ta có: *IC*(ω) = {0.47, 0.24, 0.53, 0, 0.60, 0.82, 1, 0.28, 0.82, 0.49, 1, 0.40, 0.55, 0.49}.

Vậy thuộc tính *Lương* sau khi được định lượng có giá trị khoảng: {[0.47, 0.47], [0.40, 0.40], [0.53, 0.53], [0.00, 0.02], [0.60, 0.60], [0.82, 0.82],[0.96, 1.00], [0.28, 0.28],[0.82, 0.82], [0.49, 0.49], [0.96, 1.00], [0.32, 0.4], [0.55, 0.55], [0.49, 0.49]}.

3.3. Phân lớp dữ liệu bằng cây quyết định mờ dựa trên cách thức đối sánh khoảng mờ

3.3.1. Thuật toán phân lớp dữ liệu bằng cây quyết định mờ HAC4.5 dựa trên đối sánh khoảng mờ

a. Ý tưởng thuật toán: dựa vào thuật toán C4.5 của Quinlan phát triển [62], bằng cách tính tỷ lệ lợi ích thông tin làm cơ sở để tìm điểm phân chia. Sau khi chọn được thuộc tính để phân lớp tập dữ liệu, nếu thuộc tính là kiểu rời rạc thì phân lớp theo giá trị phân biệt của chúng, nếu thuộc tính là liên tục thì ta phải tìm ngưỡng của phép tách để chia thành 2 tập con theo ngưỡng đó.

Do thuộc tính mờ của tập huấn luyện đã được đã được phân hoạch theo khoảng mờ và là một đoạn con của [0, 1], theo Mục 3.2.2. Thêm vào đó, trên các khoảng mờ của thuộc tính mờ, chúng ta có thể sắp thứ tự tuyến tính theo quan hệ trước sau, Mục 3.2.1, nên ta có thể so sánh để phân ngưỡng cho tập giá trị này tại một khoảng bất kỳ $I(x) = [I_a, I_b] \subseteq [0, 1]$ tương tự như các giá trị số liên tục trong C4.5. Vậy, việc tìm ngưỡng cho phép tách trên thuộc tính mờ cũng dựa theo tỷ lệ lợi ích thông tin của đoạn mờ đang xét và các ngưỡng của các khoảng mờ trong thuộc tính tại nút đó, tương tự như cách tính tỷ lệ lợi ích thông tin của thuộc tính số trong C4.5.

b. Tính lợi ích thông tin cho các khoảng mờ tại thuộc tính mờ: với thuộc tính mờ A_i đã được định lượng theo khoảng mờ, không mất tính tổng quát, ta giả sử có k khoảng khác nhau và chúng ta sắp xếp theo quan hệ thứ tự trước sau.

$$[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}] < \ ... < [I_{a_k}, I_{b_k}] \tag{3.3}$$

Ta có k ngưỡng được tính $Th_i^{HA} = [I_{a_i}, I_{b_i}]$, (với $1 \le i < k$). Tại mỗi ngưỡng Th_i^{HA} của đoạn mờ $[I_{a_i}, I_{b_i}]$ được chọn, tập dữ liệu D còn lại của nút này được chia làm 2 tập:

$$D_1 = \{\forall [I_{a_j}, I_{b_j}] : [I_{a_j}, I_{b_j}] < Th_i^{HA}\} \tag{3.4}$$

$$D_2 = \{\forall [I_{a_j}, I_{b_j}] : [I_{a_j}, I_{b_j}] > Th_i^{HA}\} \tag{3.5}$$

Lúc này ta có:

$$Gain^{HA}(Th_i^{HA}, D) = Entropy(D) - \frac{|D_1|}{|D|} \times Entropy(D_1) - \frac{|D_2|}{|D|} \times Entropy(D_2) \quad (3.6)$$

$$SplitInfo^{HA}(Th_i^{HA}, D) = -\frac{|D_1|}{|D|} \times log_2 \frac{|D_1|}{|D|} - \frac{|D_2|}{|D|} \times log_2 \frac{|D_2|}{|D|} \quad (3.7)$$

$$GainRatio^{HA}(Th_i^{HA}, D) = \frac{Gain^{HA}(Th_i^{HA}, D)}{SplitInfo^{HA}(Th_i^{HA}, D)} \quad (3.8)$$

Trên cơ sở tính toán tỷ lệ lợi ích thông tin cho các ngưỡng, ngưỡng nào có tỷ lệ lợi ích thông tin lớn nhất sẽ được chọn để phân tách tập *D*.

c. Thuật toán HAC4.5: với tập mẫu dữ liệu nghiệp vụ huấn luyện *D* có *m* thuộc tính, chứa thông tin mờ chưa thuần nhất, ta có thuật toán học phân lớp dữ liệu bằng cây quyết định theo khoảng mờ HAC4.5 như sau:

Thuật toán HAC4.5

Vào: *mẫu D có n bộ, m thuộc tính dự đoán và thuộc tính quyết định Y.*
Ra: *Cây quyết định theo khoảng mờ S.*

Mô tả thuật toán:

For each (*thuộc tính mờ X của D*) do
Begin
Xây dựng ĐSGT $\underline{X}_k$ tương ứng với thuộc tính mờ X;
Chuyển các giá trị số và giá trị ngôn ngữ của X về các giá trị đoạn $\subseteq$ [0, 1];
End;
Khởi tạo tập các nút lá S; S = D;
For each *(nút lá L thuộc S)* do
If (*L thuần nhất) Or (L là rỗng*) then
Gán nhãn cho nút tương ứng với giá trị thuần nhất của L;
Else
Begin
X = Thuộc tính tương ứng có GainRatio hay $GainRatio^{HA}$ lớn nhất;
Gán nhãn cho nút tương ứng với tên thuộc tính X;
If (*L là thuộc tính mờ*) then

Begin

T = Ngưỡng có GainRatio^{HA} lớn nhất;

Bổ sung nhãn T vào S;

$S_1 = \{I_{x_i} \mid I_{x_i} \subseteq L, I_{x_i} < T\}$;

$S_2 = \{I_{x_i} \mid I_{x_i} \subseteq L, I_{x_i} > T\}$;

Tạo 2 nút con cho nút hiện tại tương ứng với hai tập S_1 và S_2;

Đánh dấu nút L đã xét;

End

Else If (*L là thuộc tính liên tục*) then

Begin

Chọn ngưỡng T tương ứng có Gain lớn nhất trên X;

$S_1 = \{x_i \mid x_i \in Dom(L), x_i \leq T\}$; $S_2 = \{x_i \mid x_i \in Dom(L), x_i > T\}$;

Tạo 2 nút con cho nút hiện tại tương ứng với hai tập S_1 và S_2;

Đánh dấu nút L đã xét;

End

Else //*L là thuộc tính rời rạc*

Begin

$P = \{x_i \mid x_i \in K, x_i$ *đơn nhất*$\}$;

For each (*mỗi* $x_i \in P$) do

Begin

$S_i = \{x_j \mid x_j \in Dom(L), x_j = x_i\}$;

Tạo nút con thứ i cho nút hiện tại tương ứng với S_i;

End;

Đánh dấu nút L đã xét;

End;

End;

Với *m* là số thuộc tính, *n* là số thể hiện của tập huấn luyện, độ phức tạp của C4.5 là $O(m \times n \times log\ n)$; Với HAC4.5, trước tiên ta mất $O(n^2)$ cho mỗi một thuộc tính mờ để tính các phân hoạch đoạn mờ. Sau đó, độ phức tạp của thuật toán tại mỗi bước lặp theo thuộc tính m_i là $O(n \times log\ n)$ nếu m_i không phải là

thuộc tính mờ và là $O(n^2 \times log\ n)$ nếu là thuộc tính mờ do phải mất thêm $O(n)$ để tìm ngưỡng cho các khoảng mờ đối với thuộc tính này. Vậy độ phức tạp của HAC4.5 là $O(m \times n^2 \times log\ n)$.

Tính đúng của giải thuật được rút ra từ tính đúng của C4.5 và cách thức đối sánh các giá trị khoảng ở Mục 3.2.1.

■

Do sử dụng tư tưởng của C4.5 nên tại nút phân chia này không xảy ra việc phân chia *k-phân,* tránh được tình trạng dàn trải theo chiều ngang dẫn đến tình trạng "q*uá khớp*" trên cây kết quả. Việc thêm cho phí *O(n)* là không quá lớn nên chấp nhận được trong quá trình huấn luyện, hơn nữa, quá trình huấn luyện chỉ thực hiện một lần và dùng để dự đoán cho nhiều lần.

3.3.2. Cài đặt thử nghiệm và đánh giá thuật toán HAC4.5

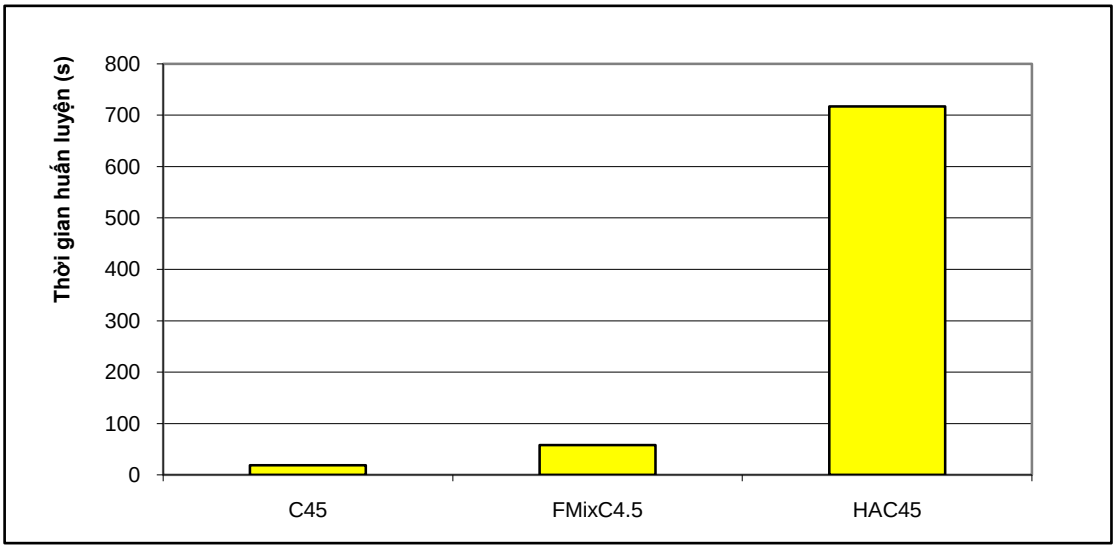
Chương trình thực nghiệm được cài đặt bằng ngôn ngữ Java (Eclipse Mars Release (4.5.0) trên máy tính có cấu hình: Processor: Intel® Core™i5-2450 CPU @ 2.50GHz (4CPUs), ~ 2.50 GHz, RAM 4GB, System type 64bit cho đồng thời cả 3 thuật toán: C4.5, đối sánh theo điểm mờ FMixC4.5 và đối sánh theo khoảng mờ HAC4.5 trên đồng thời 2 bộ dữ liệu là Mushroom và Adult.

a. Kết quả trên dữ liệu phân loại nấm Mushroom

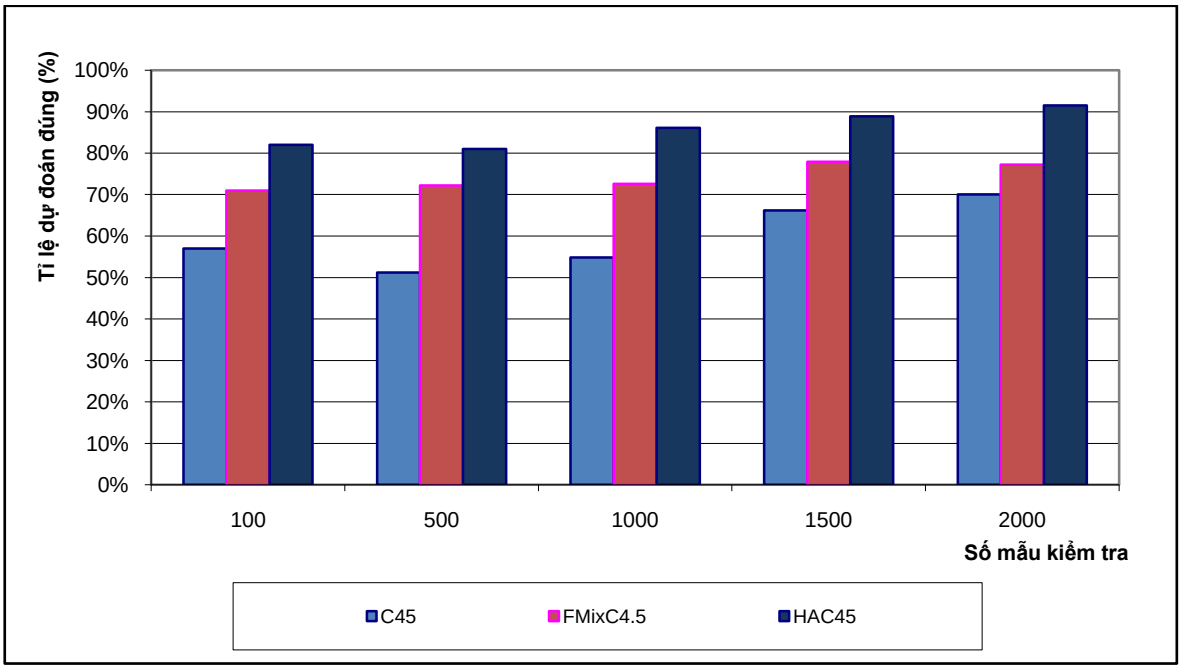
Bộ dữ liệu Mushroom có hơn 8000 mẫu tin gồm 24 thuộc tính như ở Bảng 2.5. Luận án tách riêng biệt 5000 mẫu tin cho tập huấn luyện và dùng 3000 mẫu còn lại để chọn ngẫu nhiên 2000 mẫu dùng cho việc kiểm tra.

Bảng 3.2. Bảng so sánh kết quả dùng 5000 mẫu huấn luyện của thuật toán C4.5, FMixC4.5 và HAC4.5 trên cơ sở dữ liệu có chứa thuộc tính mờ Mushroom

Thuật toán	**Thời gian huấn luyện (s)**	**Số lượng mẫu và độ chính xác dự đoán (%)**				
		100	**500**	**1000**	**1500**	**2000**
C4.5	18.9	57.0	51.2	54.8	66.2	70.0
FMixC4.5	58.2	71.0	72.2	72.6	77.9	77.2
HAC4.5	717.3	82.0	81.0	86.1	88.9	91.5



Hình 3.1. So sánh thời gian huấn luyện của HAC4.5 với 5000 mẫu của Mushroom



Hình 3.2. Tỷ lệ kiểm tra từ 100 đến 2000 trên mẫu dữ liệu Mushroom của HAC4.5

b. Kết quả trên dữ liệu dự đoán thu nhập Adult

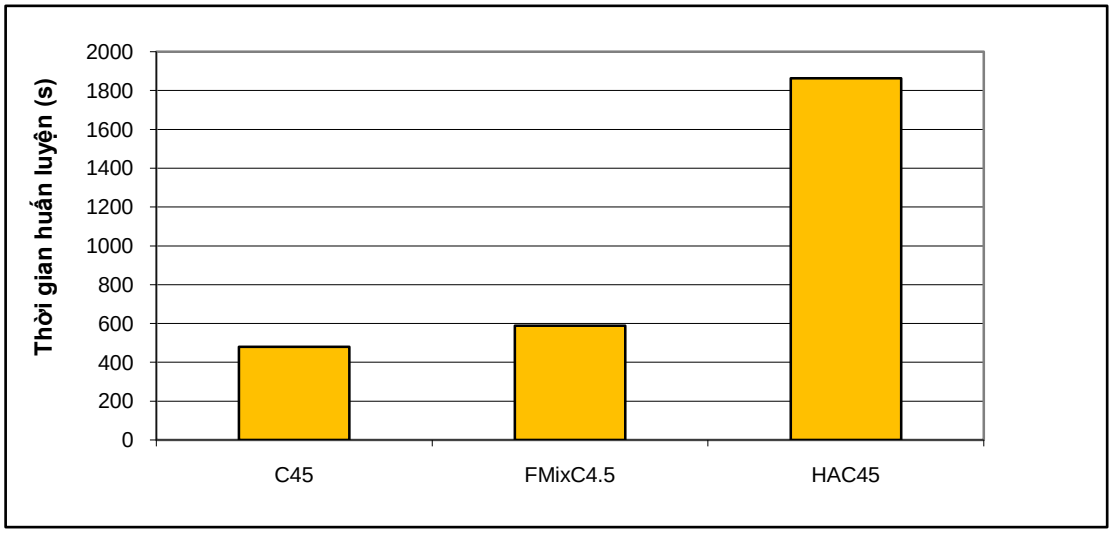
Bộ dữ liệu Adult có hơn 40000 mẫu tin gồm 15 thuộc tính bao gồm dữ liệu rời rạc, liên tục, Logic và mờ, trong đó có 2 thuộc tính *Age* (*Tuổi*) và *HoursPerWeek* (*Số giờ làm việc*) chứa cả dữ liệu rõ và mờ, chi tiết được mô tả ở Bảng 3.3. Luận án tách riêng biệt 20000 mẫu tin cho tập huấn luyện và dùng 20000 mẫu còn lại để chọn ngẫu nhiên 5000 mẫu dùng cho việc kiểm tra.

Bảng 3.3. Thông số thuộc tính tập huấn luyện từ cơ sở dữ liệu Adult

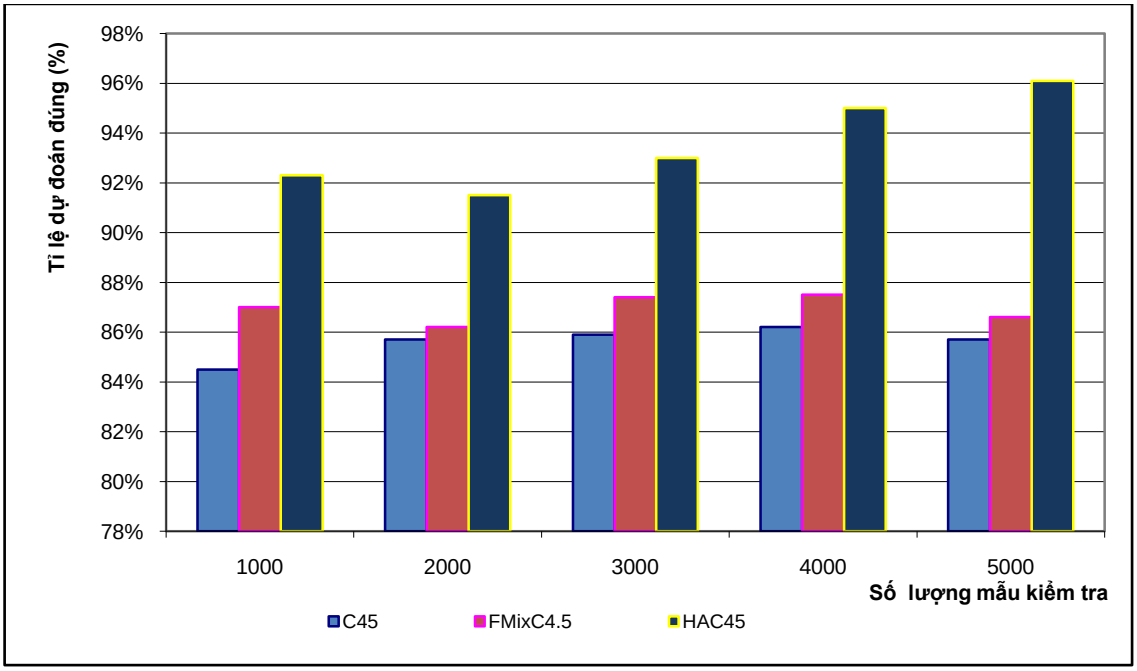
STT	Tên trường	Lực lượng	Kiểu thuộc tính	Miền trị
1	Age	200	Mờ	Rõ: [17, 100]
2	Workclass	9	Rời rạc	
3	FnlWgt	27026	Số	[12285, 1490400]
4	Education	16	Rời rạc	
5	EducationNum	16	Số	[1, 16]
6	MaritalStatus	7	Rời rạc	
7	Occupation	15	Rời rạc	
8	Relationship	6	Rời rạc	
9	Race	5	Rời rạc	
10	Sex	2	Logic	
11	CapitalGain	122	Số	[0, 99999]
12	CapitalLoss	99	Số	[0, 4356]
13	HoursPerWeek	160	Mờ	Rõ: [1, 60]
14	NativeCountry	42	Rời rạc	
15	Salary	2	Logic	

Bảng 3.4. Bảng so sánh kết quả với 20000 mẫu huấn luyện của thuật toán C4.5, FMixC4.5 và HAC4.5 trên cơ sở dữ liệu có chứa thuộc tính mờ Adult

Thuật toán	Thời gian huấn luyện (s)	Số lượng mẫu và độ chính xác dự đoán (%)				
		1000	2000	3000	4000	5000
C4.5	479.8	84.5	85.7	85.9	86.2	85.7
FMixC4.5	589.1	87.0	86.2	87.4	87.5	86.6
HAC4.5	1863.7	92.3	91.5	93.0	95.0	96.1



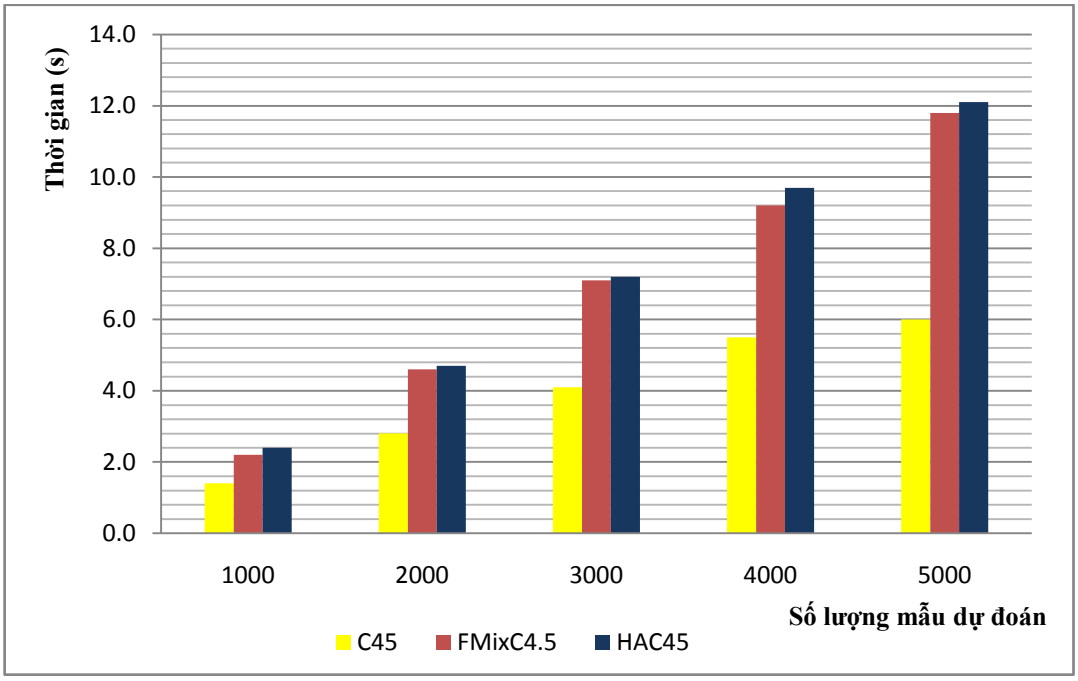
Hình 3.3. So sánh thời gian huấn luyện với 20000 mẫu của Adult



Hình 3.4. So sánh tỷ lệ kiểm tra từ 1000 đến 5000 trên mẫu dữ liệu của Adult.

Bảng 3.5. Đối sách thời gian kiểm tra từ 1000 đến 5000 mẫu trên dữ liệu Adult

Thuật toán	**Số lượng mẫu và thời gian dự đoán (s)**				
	1000	**2000**	**3000**	**4000**	**5000**
C4.5	1.4	2.8	4.1	5.5	6.0
FMixC4.5	2.2	4.6	7.1	9.2	11.8
HAC4.5	2.4	4.7	7.2	9.7	12.1



Hình 3.5. So sánh thời gian kiểm tra từ 1000 đến 5000 trên dữ liệu Adult

c. Đánh giá kết quả thực nghiệm

Việc đồng thời cài đặt cả 3 thuật toán C4.5, đối sánh theo điểm mờ FMixC4.5 và đối sánh theo khoảng mờ HAC4.5, kết quả đánh giá trên cùng các bộ dữ liệu là Mushroom và Adult đã cho chúng ta:

Chi phí thời gian: thuật toán C4.5 luôn cho thời gian nhanh nhất trong tất cả các bộ mẫu kể cả trong quá trình huấn luyện hay kiểm tra, vì nó bỏ qua các giá trị mờ trong tập mẫu nên không mất thời gian xử lý. Việc chúng ta thuần nhất tập mẫu dựa trên đối sánh theo điểm và sau đó dùng tập mẫu này để huấn luyện cây phải trải qua quá trình xây dựng các ĐSGT cho các trường mờ và chi phí để thuần nhất các giá trị ban đầu nên tốn nhiều thời gian hơn so với C4.5.

Vì phải trải qua quá trình xây dựng các ĐSGT cho các trường mờ và chi phí để chuyển đổi các giá trị về đoạn [0, 1] ban đầu, hơn nữa, tại mỗi bước lặp cần thêm thời gian để chọn đoạn phân chia nên thời gian huấn luyện của HAC4.5 khá chậm, tốn nhiều thời gian so với các thuật toán khác. Trong quá trình dự đoán, do cũng mất thời gian cho quá trình xử lý dữ liệu mờ của các giá trị dự đoán nên thời gian dự đoán của HAC4.5 cũng nhiều hơn so với C4.5, Hình 3.5.

Kết quả dự đoán: vì C4.5 bỏ qua các giá trị mờ trong tập mẫu, chỉ quan tâm các giá trị rõ nên làm mất dữ liệu tại các trường mờ, do đó kết quả dự đoán không

cao. Việc xây dựng một ĐSGT tại các trường mờ và dùng nó để thuần nhất tập mẫu dựa trên đối sánh theo điểm cho chúng ta tập huấn luyện thuần nhất chứa cả dữ liệu rõ và mờ nên kết quả của cây được huấn luyện sẽ tốt hơn, vì thế kết quả trong các quá trình dự đoán của FMixC4.5 tốt hơn C4.5. Tuy vậy, kết quả dự đoán của FmixC4.5 vẫn chưa thật sự tốt vì việc phân hoạch theo điểm mờ sẽ làm xuất hiện sai số của các giá trị rõ tại các điểm chọn để phân chia.

Kết quả dự đoán tại HAC4.5 cho kết quả tốt hơn vì trong quá trình huấn luyện cây, chúng ta đã xử lý được các giá trị mờ nhưng vẫn giữ nguyên các giá trị rõ nên không làm xuất hiện sai số trong quá trình phân hoạch, do đó kết quả của quá trình dự đoán của HAC4.5 tốt hơn các thuật toán khác. Mặc dầu HAC4.5 phải tốn nhiều thời gian cho quá trình huấn luyện nhưng cho cây kết quả có khả năng dự đoán cao, và vì quá trình huấn luyện chúng ta chỉ thực hiện một lần mà việc dự đoán dựa trên cây kết quả được thực hiện nhiều lần nên chi phí thời gian trong quá trình xây dựng HAC4.5 là chấp nhận được.

3.4. Xây dựng khái niệm khoảng mờ lớn nhất và phương pháp học nhằm tối ưu mô hình cây quyết định mờ

3.4.1. Phát biểu bài toán học phân lớp dữ liệu bằng cây quyết định mờ theo hướng đa mục tiêu

Trước hết chúng ta cần nhắc lại, trong bài toán phân lớp dữ liệu bằng cây quyết định mờ $S : D \rightarrow Y$ đã phát biểu ở (1.4), chúng ta có $Y = \{y_1, ..., y_n\}$ là tập các nhãn của các lớp, $D = A_1 \times ... \times A_m$ là tích Đề-các của các miền của m thuộc tính tương ứng. Với $f_h(S)$ là hàm đánh giá khả năng dự đoán của cây, $f_n(S)$ là hàm thể hiện số nút của cây kết quả nhằm đánh giá tính đơn giản của cây đối với người dùng. Mục tiêu của bài toán đã được nêu ở (1.13) nhằm đạt $f_h(S) \rightarrow max$ và $f_n(S) \rightarrow min$.

Các nghiên cứu ở Chương 2 và Mục 3.3 của luận án đã chỉ ra hai mục tiêu ở trên khó có thể đạt được đồng thời. Khi số nút của cây giảm đồng nghĩa với lượng tri thức về bài toán giảm thì nguy cơ phân lớp sai tăng lên, nhưng khi có quá nhiều nút cũng có thể gây ra sự quá khớp thông tin trong quá trình phân lớp. Bên cạnh đó, sự phân chia tại mỗi nút ảnh hưởng đến tính phổ quát hay cá thể tại nút đó. Nếu sự phân chia tại một nút là nhỏ sẽ làm tăng tính phổ quát và ngược lại nếu sự phân chia lớn sẽ làm tăng tính cá thể của nút đó. Tính phổ quát của nút trên cây sẽ làm tăng khả năng dự đoán nhưng nguy cơ gây sai số lớn, trong khi

tính cá thể giảm khả năng dự đoán nhưng lại tăng tính đúng đắn nhưng nó cũng là nguyên nhân của tình trạng quá khớp trên cây. Các đề xuất ở MixC4.5, FMixC4.5, HAC4.5 chính là các giải pháp nhằm giảm thiểu sai số và linh hoạt trong dự đoán. Tuy nhiên số nút trên cây kết quả tăng so với các phương pháp học truyền thống nên không đơn giản với người dùng và mất thời gian duyệt cây khi sử dụng, nên chúng chính là một thỏa hiệp nhằm đạt mục tiêu $f_h(S) \rightarrow max$ còn mục tiêu $f_n(S) \rightarrow min$ thì chưa được giải quyết.

3.4.2. Khái niệm về khoảng mờ lớn nhất và cách thức tính khoảng mờ lớn nhất cho các thuộc tính mờ

Trong ĐSGT, một hạng từ có thể mang ngữ nghĩa của một hạng từ khác tức hạng từ được dùng để sinh ra nó. Chẳng hạn "*very old"* mang ngữ nghĩa của "*old"*, "*very little old"* cũng mang ngữ nghĩa của *"old"*. Vậy, hai hạng từ "*very old"* và *"very little old"* đều có quan hệ kế thừa ngữ nghĩa (hay quan hệ kế thừa) của "*old*", ta gọi "*very old"* và *"very little old"* có quan hệ kế thừa ngữ nghĩa.

Định nghĩa 3.4. Cho một ĐSGT $\underline{X} = (X, G, H, \preceq)$, với $\forall x, y \in X$ được gọi là có quan hệ kế thừa ngữ nghĩa với nhau và được ký hiệu $\sim(x, y)$ nếu $\exists z \in X$, $x = h_{i_n} .. h_{i_1} z$, $y = h_{j_m} .. h_{j_1} z$.

Mệnh đề 3.1. $\forall x, y \in X$ xác định hai khoảng mờ mức k và mức l lần lượt là $I_k(x)$ và $I_l(y)$, chúng hoặc không có quan hệ kế thừa, hoặc có quan hệ kế thừa với nhau nếu $\exists z \in X$, $|z| = v$, $v \leq min(l, k)$, $I_L(z) \leq I_L(y)$, $I_R(z) \geq I_R(y)$, và $I_L(z) \leq I_L(x)$, $I_R(z) \geq I_R(x)$ hay $I_v(z) \supseteq I_k(x)$ và $I_v(z) \supseteq I_l(y)$, tức là x, y được sinh ra từ z.

Chứng minh: Mệnh đề này dễ dàng suy ra từ tính phân hoạch các khoảng mờ.

■

Khi x và y có quan hệ kế thừa ngữ nghĩa, ta nói rằng khoảng tính mờ $I_v(z)$ bao hàm hai khoảng tính mờ $I_k(x)$ và $I_l(y)$, hay z bao hàm ngữ nghĩa của x và y; ký hiệu: $z = \sim(x,y)$. Theo định nghĩa, hai hạng từ x, y có quan hệ ngữ nghĩa nếu chúng có dạng $x = h_{i_n} .. h_{i_1} z$, $y = h_{j_m} .. h_{j_1} z$. Nếu $h_{i_1} = h_{j_1} = h'_1$, $h_{i_2} = h_{j_2} = h'_2$,... thì ta có $z = h'_1 c$ hoặc $z = h'_1 h'_2 c$ đều bao hàm ngữ nghĩa của x và y. Tuy nhiên, thực tế sử dụng z thay thế cho cả x và y có thể làm mất ngữ nghĩa của chúng và rõ ràng thực tế, ta phải xác định z sao cho ngữ nghĩa càng gần với x, y càng tốt.

Định nghĩa 3.5. Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$, với $x, y, z \in X$, $z = \sim(x, y)$. Nếu $\exists z_1 \in X$, $z_1 = \sim(x, y)$ và $len(z) \geq len(z_1)$ thì ta nói z có ngữ nghĩa gần với x, y nhất, hay khoảng mờ z có độ dài lớn nhất và được ký hiệu $z = \sim_{max}(x, y)$.

Định nghĩa 3.6. Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$, với $\forall x, y \in X$ *và* $\sim(x, y)$. Mức độ gần nhau của x và y theo quan hệ kế thừa ngữ nghĩa ký hiệu là $sim(x, y)$ và được định nghĩa như sau:

$$sim(x, y) = \frac{m}{max\ (k,l)}(1 - |v(x) - v(y)|) \qquad (3.9)$$

trong đó $k = len(x)$, $l = len(y)$ và $m = len(z)$ với $z = \sim_{max}(x, y)$.

Mệnh đề 3.2. Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$, với $\forall x, y \in X$, ta có các tính chất về mức độ gần nhau của các hạng từ như sau:

1. Hàm $sim(x, y)$ có tính chất đối xứng, tức là $sim(x, y) = sim(y, x)$
2. x, y không có quan hệ kế thừa ngữ nghĩa $\Leftrightarrow sim(x,y) = 0$
3. $sim(x, y) = 1 \Leftrightarrow x = y$,
4. $\forall x, y, z \in X_k$, $x \leq y \leq z \Rightarrow sim(x, z) \leq sim(x, y)$ và $sim(x, z) \leq sim(y, z)$.

Chứng minh

1. Hiển nhiên theo định nghĩa

2. Theo định nghĩa, ta có $m = 0$ khi và chỉ khi x, y không có quan hệ kế thừa ngữ nghĩa. Vậy nếu x và y không có quan hệ kế thừa ngữ nghĩa $\Rightarrow m = 0 \Rightarrow sim(x,y) = 0$. Ngược lại, khi $sim(x, y) = 0 \Rightarrow m = 0$ hoặc $(1 - |\upsilon(x) - \upsilon(y)|) = 0$. Khi $m = 0$ tức x, y không có quan hệ kế thừa ngữ nghĩa, trường hợp $(1- |\upsilon(x) -\upsilon(y)|) = 0 \Rightarrow |\upsilon(x) - \upsilon(y)| = 1 \Rightarrow x = 0, y = 1$ hoặc $x = 1, y = 0 \Rightarrow x, y$ không có quan hệ kế thừa ngữ nghĩa.

3. Do $m \leq max(k, l)$ và $0 \leq \upsilon(x), \upsilon(y) \leq 1$ nên $sim(x, y) = 1 \Leftrightarrow m = max(k, l)$ và $|\upsilon(x) - \upsilon(y)| = 0 \Leftrightarrow x = y$.

4. Theo giả thiết $x \leq y \leq z \Rightarrow \upsilon(x) \leq \upsilon(y) \leq \upsilon(z) \Rightarrow 1- |\upsilon(x) - \upsilon(z)| \leq 1- |\upsilon(x) - \upsilon(y)|$ và $1 - |\upsilon(x) - \upsilon(z)| \leq 1 - |\upsilon(y) - \upsilon(z)|$. Mặt khác, cũng theo giả thiết $x, y, z \in X_k \Rightarrow len(x) = len(y) = len(z) = k$. Vậy đặt $w_1 = \sim_{max}(x, y)$, $w_2 = \sim_{max}(y, z)$, $w_3 = \sim_{\max}(x, z)$, theo qui tắc sinh các hạng từ của ĐSGT với giả thiết $x \leq y \leq z$ nên $len(w_1) \geq len(w_3)$ và $len(w_2) \geq len(w_3)$. Vậy theo Định nghĩa 6 ta có $sim(x, y) \leq sim(x, z)$ và $sim(y, z) \leq sim(x, z)$.

■

Định nghĩa 3.7. Định nghĩa về tính kề nhau của các khoảng mờ. Cho một ĐSGT $\underline{X} = (X, G, H, \leq)$, hai khoảng tính mờ $I(x)$ và $I(y)$ được gọi là kề nhau nếu chúng có một điểm mút chung, tức là $I_L(x) = I_R(y)$ hoặc $I_R(x) = I_L(y)$.

Giả sử $I_R(x) = I_L(y)$ ta có khoảng mờ $I(x)$ nằm kề trái của khoảng mờ $I(y)$ và theo Định nghĩa 3.2, ta có $I(x) < I(y)$.

Như vậy với hai khoảng mờ x và y, ta có thể sử dụng khoảng mờ z đại diện mà không làm mất nhiều ngữ nghĩa của x và y bằng cách dựa vào hàm $sim(x, y)$ và việc sử dụng z thay thế cho x và y được xem như phép kết nhập khoảng mờ x, y thành khoảng mờ z. Ta có thuật toán tính khoảng mờ lớn nhất của hai khoảng mờ cho trước như sau:

Thuật toán tính khoảng mờ lớn nhất của hai khoảng mờ cho trước

Vào: *ĐSGT* $\underline{X} = (X, G, H, \leq)$ *và* $x, y \in X$.

Ra: $z \in X$, $z = {\sim}_{max}(x, y)$.

Mô tả thuật toán:

$k = len(x)$;

$l = len(y)$;

$v = min(k, l)$;

While $v > 0$ do

Begin

If $\exists z \in X, |z| = v$ and $I_k(x) \subseteq I_v(z)$ and $I_l(y) \subseteq I_v(z)$ then return $I_v(z)$

Else $v = v - 1$;

End;

Return *NULL;*

3.4.3. Thuật toán phân lớp dữ liệu bằng cây quyết định mờ HAC4.5* theo cách tiếp cận khoảng mờ lớn nhất

a. Ý tưởng thuật toán

Ở đây, ta cũng dựa vào thuật toán C4.5 và sử dụng cách thức đối sánh khoảng mờ tại các thuộc tính mờ của tập huấn luyện ở Mục 3.2.1. Tuy nhiên, việc tính lợi ích thông tin cho thuộc tính mờ theo khoảng mờ có thể xảy ra

trường hợp đó là các khoảng mờ khác nhau nhưng lại có chung kết quả dự đoán, điều này làm cho mô hình cây thu được có nhiều nút và khi chúng ta sử dụng mức *k* có giá trị càng lớn, để tăng tính chính xác, thì vấn đề này càng có khả năng xảy ra.

Hơn nữa, tại ngưỡng được chọn $Th_i^{HA} = [I_{a_i}, I_{b_i}]$, việc chia tập huấn luyện *D* thành các tập: $D_1 = \{\forall [I_{a_j}, I_{b_j}], |[I_{a_j}, I_{b_j}]| < Th_i^{HA}\}$ và $D_2 = \{\forall [I_{a_j}, I_{b_j}], |[I_{a_j}, I_{b_j}]| > Th_i^{HA}\}$ không phải lúc nào cũng là lựa chọn tốt nhất vì có thể thể xảy ra trường hợp một đoạn con của D_1 hoặc D_2 mới có lợi ích thông tin lớn nhất.

Do thuộc tính mờ *A* của tập huấn luyện đã được đã được phân hoạch theo khoảng mờ là một đoạn con của [0, 1] và miền dữ liệu của nó là một tập được sắp thứ tự tuyến tính theo quan hệ trước sau nên các khoảng mờ của chúng có tính kề trái và kề phải. Như vậy với hai khoảng mờ *x* và *y* nếu chúng có chung lớp dự đoán, ta có thể sử dụng khoảng mờ $z = \sim_{max}(x, y)$ thay thế mà không làm thay đổi ngữ nghĩa của *x* và *y* trong quá trình học phân lớp. Việc sử dụng phép kết nhập *z* thay thế cho *x* và *y* được thực hiện cho tất cả các khoảng mờ của thuộc tính mờ *A*.

Như vậy, thuộc tính mờ *A* của tập huấn luyện sau khi đã phân hoạch theo khoảng mờ theo khoảng mờ lớn nhất có *k* khoảng khác nhau và đã được sắp theo thứ tự trước sau là: $[I_{a_1}, I_{b_1}] < [I_{a_2}, I_{b_2}] < ... < [I_{a_k}, I_{b_k}]$ và ta tiếp tục việc tìm ngưỡng cho phép tách dựa theo tỷ lệ lợi ích thông tin của các ngưỡng tại nút đó như ở Mục 3.3.2.

b. Thuật toán HAC4.5*

Thuật toán HAC4.5*

Vào: *Tập mẫu huấn luyện D.*

Ra: *Cây quyết định khoảng mờ S.*

Mô tả thuật toán:

For each *(thuộc tính mờ X của D)* do

Begin

Xây dựng ĐSGT $\underline{X}_k$ tương ứng với thuộc tính mờ X;

Chuyển các giá trị số và giá trị ngôn ngữ của X về các giá trị $\subseteq$ [0, 1];

End*;*

Khởi tạo tập các nút lá S; S = D;

For each *(nút lá L thuộc S)* do

If (*L thuần nhất) Or (L.Tập thuộc tính là rỗng*) then

Gán nhãn cho nút tương ứng với giá trị thuần nhất của L;

Else

Begin

//Tìm và thay thế bằng các khoảng mờ lớn nhất

If (*L là thuộc tính mờ*) then

Begin

For each *(khoảng mờ x của thuộc tính L)* do

For each *(khoảng mờ y của thuộc tính L mà* $y \neq x$*)* do

Tìm và thay thế x bởi $z = \sim_{max}(x, y)$;

End;

X = Thuộc tính tương ứng có GainRatio hay $GainRatio^{HA}$ *lớn nhất;*

Gán nhãn cho nút tương ứng với tên thuộc tính X;

If (*L là thuộc tính mờ*) then

Begin

T = Ngưỡng có $GainRatio^{HA}$ *lớn nhất;*

Gán nhãn là T của thuộc tính X vào cho cây S;

$S_1 = \{I_{x_i} | I_{x_i} \subseteq L, I_{x_i} < T\}$;

$S_2 = \{I_{x_i} | I_{x_i} \subseteq L, I_{x_i} > T\}$;

Tạo 2 nút con cho nút hiện tại tương ứng với hai tập S_1 *và* S_2;

Đánh dấu nút L đã xét;

End

Else

If (*L là thuộc tính liên tục*) then

Begin

T = Ngưỡng có GainRatio lớn nhất;

S_1= {x_i| $x_i \in Dom(L)$, x_i <= T};

S_2= {x_i| $x_i \in Dom(L)$, x_i > T};

Tạo 2 nút con tương ứng với hai tập S_1 và S_2;

Đánh dấu nút L đã xét;

End

Else //*L là thuộc tính rời rạc*

Begin

P = {x_i| $x_i \in K$, x_i đơn nhất};

For each (*mỗi $x_i \in P$)* do

Begin

S_i = {x_j| $x_j \in Dom(L)$, $x_j = x_i$};

Tạo nút con thứ i cho nút hiện tại tương ứng với S_i;

End;

Đánh dấu nút L đã xét;

End;

End;

c. Đánh giá thuật toán

Như vậy, bằng việc tìm các khoảng mờ lớn nhất cho mọi khoảng mờ trong tập huấn luyện và xác nhập chúng, thuật toán HAC4.5* đã làm giảm thiểu số lượng khoảng mờ tham gia trong quá trình học xây dựng cây nên làm giảm số nút trên cây kết quả. Tuy vậy, do sự kết nhập các khoảng mờ dựa trên độ đo về tính tương tự của các khoảng mờ đó đối với thuộc tính huấn luyện nên nó không sai lệch đến kết quả phân lớp cuối cùng.

Trong mỗi bước lặp của thuật toán, chúng ta đều phải tính các khoảng mờ lớn nhất và thực hiện sát nhập các khoảng mờ với chi phí là $O(n^2)$. Để ý rằng trong thuật toán trên, chúng ta không thể chuyển việc tìm và sát nhập các khoảng mờ lớn nhất ra khỏi vòng lặp. Do khi chúng ta tìm được một thuộc tính phù hợp để tạo cây thì tại các điểm phân chia này, tập mẫu huấn luyện tương ứng trên mỗi nhánh của cây thay đổi nên các khoảng mờ của thuộc tính mờ thay đổi.

Với *m* là số thuộc tính, *n* là số thể hiện của tập huấn luyện, độ phức tạp

của C4.5 là $O(m \times n \times log\ n)$; Với HAC4.5*, trước tiên ta mất $O(n^2)$ cho mỗi một thuộc tính mờ để tính các phân hoạch đoạn mờ. Sau đó, độ phức tạp của thuật toán tại mỗi bước lặp theo thuộc tính m_i là $O(n \times log\ n)$ nếu m_i không phải là thuộc tính mờ và là $O(n \times n \times log\ n)$ nếu là thuộc tính mờ do phải mất thêm $O(n)$ để tìm ngưỡng cho các khoảng mờ đối với thuộc tính này và $O(n^2)$ để tìm khoảng mờ lớn nhất cho mỗi khoảng mờ trong tập huấn luyện. Tuy vậy, việc tìm khoảng mờ lớn nhất và xác định ngưỡng cho các khoảng mờ là tuần tự nên độ phức tạp của HAC4.5* là $O(m \times n^3 \times log\ n)$.

Tính đúng và tính dừng của thuật toán được rút ra từ tính đúng của C4.5 và cách thức đối sánh giữa các giá trị khoảng mờ.

■

3.4.4. Cài đặt thử nghiệm và đánh giá thuật toán HAC4.5*

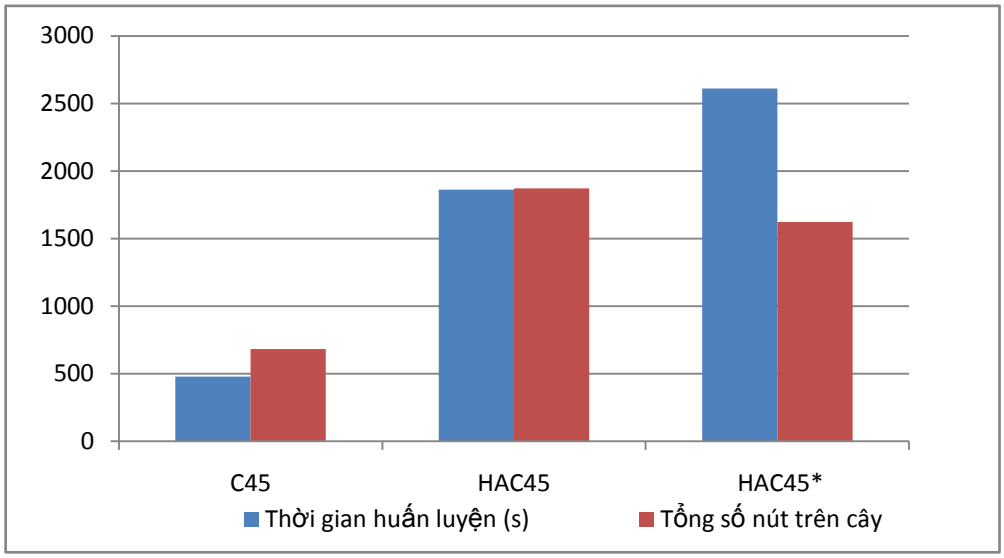
Chương trình thực nghiệm được cài đặt bằng ngôn ngữ Java (Eclipse Mars Release (4.5.0) trên máy tính có cấu hình: Processor: Intel® Core™i5-2450 CPU @ 2.50GHz (4CPUs), ~ 2.50 GHz, RAM4GB, System type 64bit cho đồng thời cả 3 thuật toán: C4.5, HAC4.5 và HAC4.5* trên cùng bộ dữ liệu đã dùng để thực nghiệm với các thuật toán khác là Adult.

a. Kết quả thực nghiệm của HAC4.5*

Với hơn 40000 mẫu tin gồm 14 thuộc tính bao gồm dữ liệu rời rạc, liên tục, logic và mờ của bộ dữ liệu Adult đã xét, trích chọn 20000 mẫu tin cho tập huấn luyện và dùng 20000 mẫu còn lại để chọn ngẫu nhiên 5000 mẫu dùng cho việc kiểm tra. Kết quả thực nghiệm ở Bảng 3.6 và Bảng 3.7.

Bảng 3.6. Đối sánh kết quả huấn luyện của HAC4.5* trên dữ liệu Adult

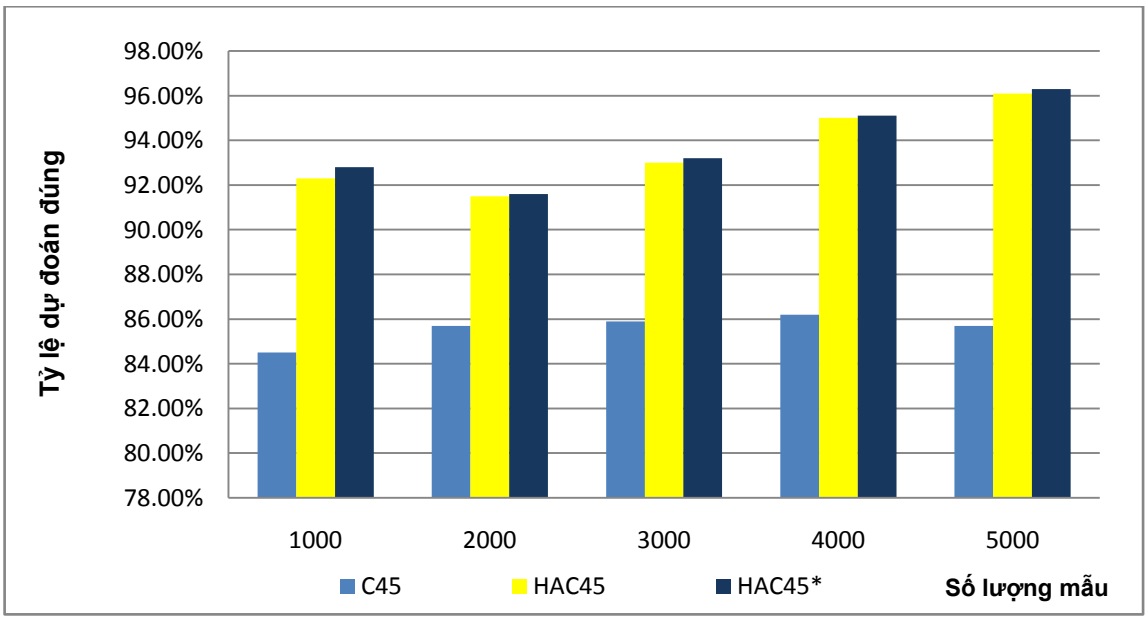
Thuật toán	Thời gian huấn luyện (s)	Tổng số nút trên cây kết quả
C4.5	479.8	682
HAC4.5	1863.7	1873
HAC4.5*	2610.8	1624



Hình 3.6. So sánh thời gian huấn luyện và số nút của cây kết quả của HAC4.5* trên tập mẫu Adult

Bảng 3.7. Tỷ lệ kiểm tra của HAC4.5* trên dữ liệu Adult

Số mẫu kiểm tra / Thuật toán	1000	2000	3000	4000	5000
C4.5	84.5%	85.7%	85.9%	86.2%	85.7%
HAC4.5	92.3%	91.5%	93.0%	95.0%	96.1%
HAC4.5*	92.8%	91.6%	93.2%	95.1%	96.3%



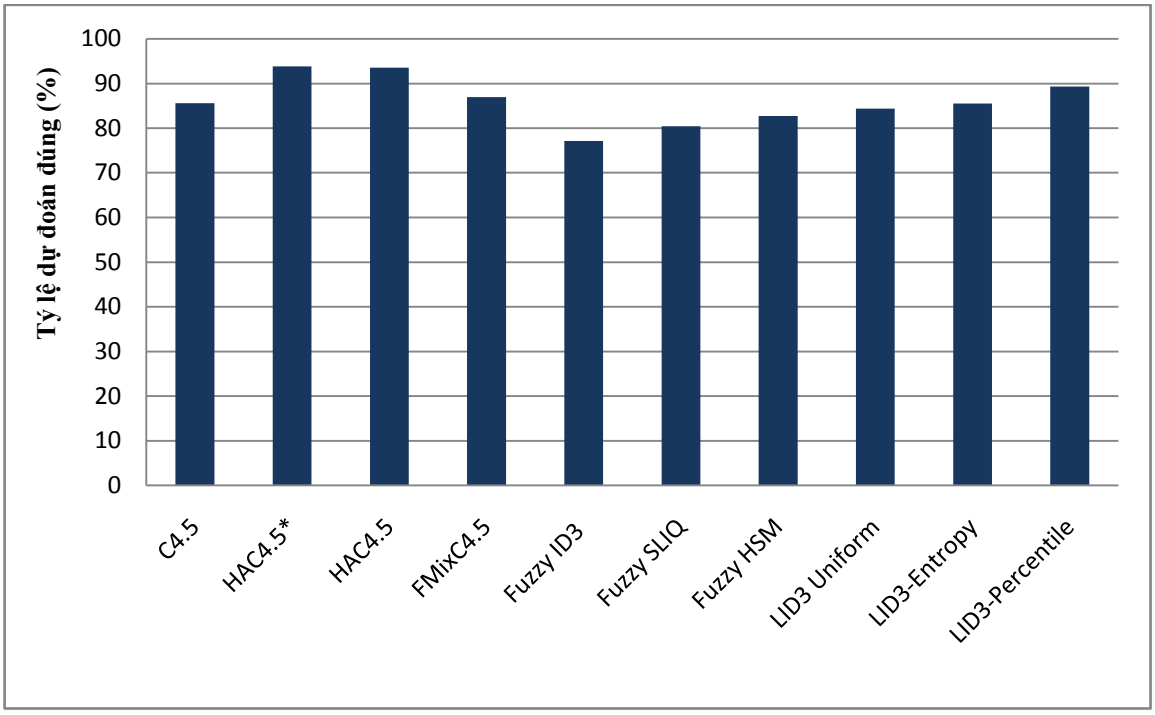
Hình 3.7. So sánh tỷ lệ kiểm tra của HAC4.5* trên mẫu dữ liệu Adult

b. Đối sánh kết quả thực nghiệm HAC4.5* với một số kết quả của các cách tiếp cận khác

Như đã nêu ở Mục 1.4.3 của luận án, quá trình huấn luyện cây quyết định mờ đã được nhiều tác giả nghiên cứu theo nhiều cách tiếp cận trên nhiều cách khác nhau. Các thuật toán nổi bậc của cách tiếp cận dựa trên lý thuyết tập mờ như: Fuzzy ID3, Fuzzy SLIQ, Fuzzy HSM [16], [19], [26], [32], [83] và các thuật toán LID3 Uniform, LID3 Entropy, LID3 Percentile của cách tiếp cận xây dựng cây quyết định ngôn ngữ [69], [84], [85] được so sánh với các thuật toán FMixC4.5, HAC4.5 và HAC4.5* đã đề xuất ở luận án được mô tả ở Bảng 3.8 và Hình 3.8.

Bảng 3.8. Kết quả dự đoán trung bình của các thuật toán FMixC4.5, HAC4.5 và HAC4.5* đối với các cách tiếp cận khác.

Thuật toán	**Tỷ lệ dự đoán chính xác (%)**
C4.5	85.60
HAC4.5*	93.80
HAC4.5	93.58
FMixC4.5	86.94
Fuzzy ID3	77.15
Fuzzy SLIQ	80.43
Fuzzy HSM	82.72
LID3 Uniform	84.37
LID3-Entropy	85.54
LID3-Percentile	89.31



Hình 3.8. So sánh tỷ lệ dự đoán của thuật toán FMixC4.5, HAC4.5 và HAC4.5* với các cách tiếp cận khác

c. Đánh giá kết quả thực nghiệm

Việc đồng thời cài đặt cả 3 thuật toán C4.5, HAC4.5 và HAC4.5* và so sánh, đánh giá các kết quả trên cùng các bộ dữ liệu đã cho phép chúng ta có các kết luận:

- **Chi phí huấn luyện:** thuật toán C4.5 luôn cho thời gian nhanh nhất trong tất cả các bộ mẫu kể cả trong quá trình huấn luyện hay kiểm tra, vì nó bỏ qua các giá trị mờ trong tập mẫu nên không mất thời gian xử lý.

HAC4.5 phải trải qua quá trình xây dựng các ĐSGT cho các trường mờ và chi phí để chuyển đổi các giá trị về đoạn [0, 1] ban đầu và tại mỗi bước cần thêm thời gian để chọn đoạn phân chia nên tốn nhiều thời gian hơn nhiều so với C4.5.

HAC4.5* vì mỗi bước lặp cần thêm thời gian để tìm các khoảng mờ lớn nhất cho miền trị mờ của thuộc tính mờ tương ứng nên HAC4.5* chậm nhất so với các thuật toán khác, Bảng 3.6, Hình 3.6.

- **Kết quả dự đoán:** C4.5 bỏ qua các giá trị mờ trong tập mẫu, chỉ quan tâm các giá trị rõ nên cây kết quả thu được khá giản đơn vì ít nút. Tuy nhiên, do việc bỏ qua các giá trị mờ nên làm mất dữ liệu tại các trường mờ, vì thế kết quả dự đoán không cao.

HAC4.5: với việc xây dựng một ĐSGT tại các trường mờ và dùng nó để thuần nhất tập mẫu nên chúng ta đã xử lý được các giá trị mờ mà vẫn giữ nguyên các giá trị rõ nên không làm xuất hiện thêm sai số trong quá trình phân hoạch,vì thế kết quả trong các quá trình dự đoán tốt hơn nhiều so với C4.5. Tuy vậy, so với C4.5 thì cây kết quả thu được không giản đơn vì có nhiều nút.

HAC4.5* cho kết quả tốt nhất vì trong quá trình huấn luyện cây, chúng ta đã tìm được các điểm phân hoạch tốt nhất tại các thuộc tính mờ nên ít cây kết quả thu được có sai số ít hơn, Bảng 3.7, Hình 3.7. Việc tìm các khoảng mờ lớn nhất và kết nhập các giá trị mờ trên thuộc tính mờ đã làm cho lực lượng của các thuộc tính mờ tương ứng giảm, vì thế số nút trên cây thu được cũng giảm, Hình 3.7, nên cây kết quả thu được là tốt nhất. Điều này đáp ứng các hàm mục tiêu ở Mục 3.4.1.

Hơn thế, đối sánh các thuật toán huấn luyện cây quyết định mờ FMixC4.5, HAC4.5 và HAC4.5* đã đề xuất trong luận án với các thuật toán trên các cách tiếp cận hiện có, tham chiếu ở Bảng 3.8 và Hình 3.8, luận án đã cho thấy việc sử dụng ĐSGT cho bài toán phân lớp dữ liệu mờ theo cách tiếp cận của luận án đạt hiệu quả dự đoán tốt hơn.

3.5. Kết luận chương 3

Trên cơ sở nhận thấy quá trình thuần nhất giá trị ngôn ngữ LD_{A_i}và giá trị số của D_{A_i} của thuộc tính mờ A_i về các giá trị trong đoạn [0, 1] làm xuất hiện các sai số và cây kết quả thu được theo FMixC4.5 chưa thật sự linh hoạt trong quá trình dự đoán. Chương này của luận án tập trung nghiên cứu quá trình học phân lớp dữ liệu bằng cây quyết định mờ nhằm đạt hai mục tiêu đề ra là $f_h(S) \rightarrow max$ và $f_n(S) \rightarrow min$. Cụ thể:

1. Nghiên cứu mối tương quan của các khoảng mờ, đề xuất phương pháp đối sánh dựa trên khoảng mờ và xây dựng thuật toán học phân lớp dựa trên khoảng mờ HAC4.5
2. Nghiên cứu và chỉ ra rằng miền trị *Min - Max* của thuộc tính mờ không phải luôn tồn tại sẵn trong tập huấn luyện. Dựa vào tính chất của ĐSGT, luận án xây dựng phương pháp nhằm có thể định lượng cho

các giá trị của thuộc tính không thuần nhất, chưa xác định *Min-Max* của tập huấn luyện.

3. Luận án đã đề xuất khái niệm khoảng mờ lớn nhất, thiết kế thuật toán HAC4.5* nhằm đồng thời đạt được các mục tiêu đó là tính hiệu quả của quá trình phân lớp và tính đơn giản và dễ hiểu đối với người dùng tức nhằm đồng thời đạt được các mục tiêu $f_h(S) \rightarrow max$ và $f_n(S) \rightarrow min$.

Thông qua việc phân tích, đánh giá các kết quả thực nghiệm trên các tập mẫu có chứa thông tin mờ của các cơ sở dữ liệu Mushroom và Adult cho đồng thời các thuật toán C4.5, HAC4.5, HAC4.5* đã cho thấy các kết quả của HAC4.5 và HAC4.5* đã có sự cải tiến đáng kể về các hàm mục tiêu $f_h(S)$ và $f_n(S)$.

KẾT LUẬN

Luận án tập trung nghiên cứu, phân tích và đánh giá các ưu nhược điểm của các kết quả đã được nghiên cứu cho việc học phân lớp bằng cây quyết định. Kết quả chính của luận án là nghiên cứu, đề xuất mô hình và các phương pháp cho việc học cây quyết định nhằm thu được cây kết quả đạt hiệu quả cao cho quá trình phân lớp và đơn giản, dễ hiểu đối với người dùng. Nội dung chính của luận án đã đạt được các kết quả cụ thể như sau:

1. Đề xuất mô hình linh hoạt cho quá trình học cây quyết định từ tập mẫu huấn luyện thực tế và phương pháp nhằm trích chọn được tập mẫu huấn luyện đặc trưng phục vụ cho quá trình huấn luyện. Phân tích, đưa ra các khái niệm về tập mẫu không thuần nhất, giá trị ngoại lai và xây dựng thuật toán để có thể thuần nhất cho các thuộc tính có chứa các giá trị này.

2. Đề xuất thuật toán xây dựng cây MixC4.5 trên cơ sở tổng hợp các ưu và nhược điểm của các thuật toán truyền thống CART, C4.5, SLIQ, SPRINT. Với việc chỉ ra các hạn chế của thuật toán FDT và FID3 cho việc học cây quyết định mờ, luận án đề xuất thuật toán FMixC4.5 phục vụ quá trình học cây quyết định trên tập mẫu không thuần nhất. Cả hai thuật toán MixC4.5 và FMixC4.5 đều được đánh giá thực nghiệm trên các cơ sở dữ liệu Northwind và Mushroom và kết quả có khả quan dự đoán tốt hơn các thuật toán truyền thống C4.5, SLIQ, SPRINT.

3. Đề xuất phương pháp đối sánh dựa trên khoảng mờ và xây dựng thuật toán học phân lớp dựa trên khoảng mờ HAC4.5. Xây dựng phương pháp nhằm có thể định lượng cho các giá trị của thuộc tính không thuần nhất, chưa xác định *Min - Max* của tập huấn luyện.

4. Luận án đưa ra khái niệm khoảng mờ lớn nhất và lấy đó làm cơ sở để thiết kế thuật toán học cây quyết định dựa trên khoảng mờ lớn nhất HAC4.5* nhằm đồng thời đạt được các mục tiêu đó là tính hiệu quả của quá trình phân lớp và tính đơn giản và dễ hiểu đối với người dùng. Các kết quả của HAC4.5,

HAC4.5* được phân tích, đánh giá thực nghiệm trên các cơ sở dữ liệu có chứa dữ liệu mờ Mushroom và Adult. Kết quả cho thấy khả năng dự đoán của các thuật toán đã đề xuất trong luận án là tốt hơn và số nút trên cây kết quả giảm nên cho hiệu quả phân lớp tốt hơn.

Các kết quả chính của luận án đã được công bố trong 7 công trình khoa học được đăng trong các hội nghị, tạp chí chuyên ngành trong và ngoài nước. Trong đó có 01 bài đăng ở tạp chí Khoa học và Công nghệ trường Đại học Khoa học Huế; 01 bài đăng ở tạp chí Khoa học Đại học Huế; 01 bài đăng ở kỷ yếu Hội thảo quốc gia Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), 02 bài đăng ở Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng CNTT&TT, Tạp chí Thông tin, Khoa học và Công nghệ, Bộ Thông tin và Truyền thông; 01 bài đăng ở tạp chí chuyên ngành Tin học và điều khiển; 01 bài đăng ở tạp chí quốc tế International Journal of Research in Engineering and Science (IJRES).

Mặc dầu vậy, trong việc lựa chọn tham số để xây dựng đại số gia tử nhằm định lượng giá trị ngôn ngữ trên tập mẫu huấn luyện, luận án đang sử dụng kiến thức của chuyên gia để xác định các tham số mà chưa có nghiên cứu nhằm đưa ra một phương pháp hoàn chỉnh cho việc lựa chọn này.

Hướng phát triển của luận án:

- Nghiên cứu nhằm đưa ra một phương pháp phù hợp để lựa chọn tham số cho ĐSGT của tập huấn luyện mà không phụ thuộc vào ý kiến chủ quan của chuyên gia.

- Mở rộng phương pháp học cây quyết định dựa trên khoảng mờ mà không hạn chế số gia tử khi xây dựng ĐSGT cho việc thuần nhất giá trị của thuộc tính mờ. Chắc chắn rằng phương pháp này mang tính tổng quát hơn cho việc ứng dụng về sau.

- Trên cơ sở của mô hình ứng dụng trong bài toán phân lớp, tiếp tục phát triển các mô hình để ứng dụng cho một số bài toán khác trong lĩnh vực khai phá dữ liệu như khai phá luật kết hợp, phân cụm dữ liệu,...

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

CT1. **Lê Văn Tường Lân,** Nguyễn Mậu Hân**,** Nguyễn Công Hào**,** *Một thuật toán học tạo cây quyết định cho bài toán phân lớp dữ liệu,* Tạp chí khoa học Đại học Huế, tập 81, số 3, trang 71-84, 2013.

CT2. **Lê Văn Tường Lân,** Nguyễn Mậu Hân**,** Nguyễn Công Hào. *Một cách tiếp cận chọn tập mẫu huấn luyện cây quyết định dựa trên đại số gia tử,* Kỷ yếu Hội nghị Quốc gia lần thứ VI *"*Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin" (FAIR), trang 251-258, 2013.

CT3. **Lê Văn Tường Lân,** Nguyễn Mậu Hân**,** Nguyễn Công Hào, *Một phương pháp xử lý giá trị ngoại lai trong tập mẫu huấn luyện cây quyết định sử dụng đại số gia tử,* Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng CNTT&TT, Tạp chí Thông tin, Khoa học và Công nghệ, Bộ TT&TT, tập V.2, số 14, trang 55-63, 2015.

CT4. **Lan L. V.**, Han N. M., Hao N. C., *A Novel Method to Build a Fuzzy Decision Tree Based On Hedge Algebras*, International Journal of Research in Engineering and Science (IJRES), Volume 4, Issue 4, pages 16-24, 2016.

CT5. **Le Van Tuong Lan**, Nguyen Mau Han, Nguyen Cong Hao, *Algorithm to build fuzzy decision tree for data classification problem based on fuzziness intervals matching*, Journal of Computer Science and Cybernetics, V.32, N.4, DOI 10.15625/1813-9663/30/4/8801, 2016.

CT6. **Lê Văn Tường Lân,** Nguyễn Mậu Hân**,** Nguyễn Công Hào**,** *Mô hình cây quyết định mờ cho bài toán phân lớp dữ liệu,* Tạp chí Khoa học và công nghệ, trường Đại học Khoa học – Đại học Huế, tập 81, số 3, trang 19-44, 2017.

CT7. **Lê Văn Tường Lân,** Nguyễn Mậu Hân**,** Nguyễn Công Hào**,** *Tối ưu quá trình học cây quyết định cho bài toán phân lớp theo cách tiếp cận khoảng mờ lớn nhất",* Chuyên san Các công trình nghiên cứu, phát triển và ứng dụng CNTT&TT, Tạp chí Thông tin, Khoa học và Công nghệ, Bộ TT&TT, Tập V-2, Số 18 (38), trang 42-50, 2017.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

[1]. Nguyễn Công Hào: *Cơ sở dữ liệu mờ với thao tác dữ liệu dựa trên đại số gia tử*, Luận án Tiến sĩ Toán học, Viện Công nghệ Thông tin, 2008.

[2]. Nguyễn Cát Hồ, *Cơ sở dữ liệu mờ với ngữ nghĩa đại số gia tử*, Bài giảng trường Thu - Hệ mờ và ứng dụng, Viện Toán học Việt Nam, 2008.

[3]. Lê Anh Phương, *Một tiếp cận xây dựng miền giá trị chân lý ngôn ngữ trong các hệ logic*, Luận án Tiến sĩ Toán học, Viện Công nghệ Thông tin và Truyền Thông – Đại học Bách Khoa Hà Nội, 2013.

[4]. Lê Xuân Việt, *Định lượng ngữ nghĩa các giá trị của biến ngôn ngữ dựa trên đại số gia tử và ứng dụng*, Luận án Tiến sĩ Toán học, Viện Công nghệ Thông tin, 2008.

[5]. Lê Xuân Vinh, *Về một cơ sở đại số và logíc cho lập luận xấp xỉ và ứng dụng*, Luận án Tiến sĩ Toán học, Viện Công nghệ Thông tin - Viện Khoa học và Công nghệ Việt Nam, 2006.

TIẾNG ANH

[6]. Abonyi J., Roubos J.A., Setnes M., *Learning fuzzy classification rules from labeled data*, Information Sciences, vol. 150, 2003.

[7]. Adler D., *Genetic Algorithms and Simulated Annealing: A Marriage Proposal*, Proc of the International Conf. On Neural Networks, vol. 2, pp. 1104-1109, 1994.

[8]. Alberto Fernández, María Calderón, Francisco Herrera, *Enhancing Fuzzy Rule Based Systems in Multi-Classication Using Pairwise Coupling with Preference Relations*, University of Navarra, Spain, 2009.

[9]. A. K. Bikas, E. M. Voumvoulakis, N. D. Hatziargyriou, *Neuro-Fuzzy Decision Trees for Dynamic Security Control of Power Systems*,

Department of Electrical and Computer Engineering, NTUA,Athens, Greece, 2008.

[10]. Anuradha, Gaurav Gupta, *Fuzzy Decision Tree Construction in Crisp Scenario through fuzzified Trapezoidal Membership Function*, Internetworking Indonesia Journal, Vol.7, No.2, pp. 21-28, 2015.

[11]. B. Chandra, *Fuzzy SLIQ Decision Tree Algorithm*, IEEE, 2008.

[12]. Bhatt R. B., *Neuro-fuzzy decision trees for content popularity model and multi-genre movie recommendation system over social network*, IEEE, 2009.

[13]. Biswajeet Pradhan, *A comparative study on the predictive ability of the decision tree, support vector machine and neuro-fuzzy models in landslide susceptibility mapping using GIS*, Computers & Geosciences, Volume 51, pp. 350-365, 2013.

[14]. Breiman L., Friedman J. H., Olshen R. A., *Classification and Regression Trees*, CRC Press, 1984.

[15]. Buckley J. J., Siler W., *Fuzzy Expert Systems and Fuzzy Reasoning*, John Wiley & Sons, Inc., USA, 2005.

[16]. Chida A., *Enhanced Encoding with Improved Fuzzy Decision Tree Testing Using CASP Templates*, Computational Intelligence Magazine, IEEE, 2012.

[17]. Chang, Robin L. P. Pavlidis, Theodosios, *Fuzzy Decision Tree Algorithms*, Man and Cybernetics, IEEE , 2007.

[18]. Charu C. Aggarwal , *Outlier Analysis*, IBM T. J. Watson Research Center Yorktown Heights, New York, 2016.

[19]. Daveedu Raju Adidela, Jaya Suma. G, Lavanya D. G., *Construction of Fuzzy Decision Tree using Expectation Maximization Algorithm*, International Journal of Computer Science and Management Research , Vol 1 Issue 3 October 2012.

[20]. D. Hawkins, *Identification of Outliers*, Chapman and Hall, 1980.

[21]. Dubois D., Prade H., *Fuzzy Sets in Approximate Reasoning and Information Systems*, Kluwer Academic Publishers, USA, 1999.

[22]. Fernandez A., Calderon M., Barrenechea E., *Enhancing Fuzzy Rule Based Systems in Multi-Classication Using Pairwise Coupling with Preference Relations*, EUROFUSE Workshop Preference Modelling and Decision Analysis, Public University of Navarra, Pamplona, Spain, 2009.

[23]. Fuller R., *Neural Fuzzy Systems*, Physica-Verlag, Germany, 1995.

[24]. Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, Rui Zhang, *Extreme Learning Machine for Regression and Multiclass Classification*, IEEE Transactions On Systems, Man, and Cybernetics, Vol. 42, No. 2, pp. 513-529, 2012.

[25]. Hamid Kiavarz Moghaddam, *Vehicle Accident Severity Rule Mining Using Fuzzy Granular Decision Tree*, University of Calgary, 2015.

[26]. Hesham A. Hefny, Ahmed S. Ghiduk, Ashraf Abdel Wahab, *Effective Method for Extracting Rules from Fuzzy Decision Trees based on Ambiguity and Classifiability*, Universal Journal of Computer Science and Engineering Technology, Cairo University, Egypt., pp. 55-63, 2010.

[27]. Ho N. C., Long N. V., *Fuzziness measure on complete hedges algebras and quantifying semantics of terms in linear hedge algebras*, Fuzzy Sets and Systems, vol.158, pp. 452-471, 2007.

[28]. Ho N. C., Nam H. V., *An algebraic approach to linguistic hedges in Zadeh's fuzzy logic*, Fuzzy Sets and Systems, vol. 129, pp. 229-254, 2002.

[29]. Ho N. C., Wechler W., *Hedge algebras: an algebraic approach to structures of sets of linguistic domains of linguistic truth variables*, Fuzzy Sets and Systems, 35(3), pp. 281-293, 1990.

[30]. Ho N. C., Wechler W., *Extended algebra and their application to fuzzy logic*, Fuzzy Sets and Systems, vol. 52, pp. 259–281, 1992.

[31]. Ho N. C., Lan V. N., Viet L. X., *Optimal hedge-algebras-based controller: Design and application*, Fuzzy Sets and Systems, vol. 159, pp. 968-989, 2008.

[32]. Hongze Qiu, Haitang Zhang, *Fuzzy SLIQ Decision Tree Based on Classification Sensitivity*, Modern Education and Computer Science (MECS), pp. 18-25, 2011.

[33]. Hou Yuan-long, Chen Ji-lin, Xing Zong-yi, Jia Li-min, Tong Zhong-zhi, *A Multi-objective Genetic-based Method for Design Fuzzy Classification Systems*, International Journal of Computer Science and Network Security, vol. 6, no. 8, pp. 110-117, 2006

[34]. Huang J., Ertekin S., Song Y., Zha H., Giles C. L., *Efficient Multiclass Boosting Classification with Active Learning*, Seventh SIAM International Conference, Minnesota University, America, 2007

[35]. Ishibuchi H., Nakashima T., *Effect of Rule Weights in Fuzzy Rule-Based Classification Systems*, IEEE Trans. on Fuzzy Systems, vol. 9, no. 4, 2001.

[36]. Ishibuchi H., Nojima Y., Kuwajima I., *Parallel distributed genetic fuzzy rule selection*, SpringerLink, vol. 13, no. 5, 2009.

[37]. James F. Smith, Vu N. H. T., *Genetic program based data mining of fuzzy decision trees and methods of improving convergence and reducing bloat*, Data Mining, Intrusion Detection, Information Assurance, 2007.

[38]. Jaime Carbonell, *An Empirical Comparison of Pruning Methods for Decision Tree Induction*, *Machine Learning*, Kluwer Academic Publishers, Boston, Manufactured in The Netherlands, Vol 4, pp. 227-243, 1989.

[39]. Jan Bohacik, C. Kambhampati, Darryl N. Davis, JFG Cleland, *Analysis of Fuzzy Decision Trees on Expert Fuzzified Heart Failure Data*, IEEE International Conference on Systems, Man and Cybernetics, pp. 350-355, 2013.

[40]. José Antonio Sanz, Alberto Fernández, Humberto Bustince, *A Linguistic Fuzzy Rule-Based Classification System Based On a New Interval-Valued Fuzzy Reasoning Method With Tuning and Rule Selection*, IEEE Transactions on Fuzzy systems, vol. 21, no. 3, pp. 399-411, 2013.

[41]. Jothikumar R., Siva Balan R. V., *C4.5 classification algorithm with back-track pruning for accurate prediction of heart disease*,

Computational Life Science and Smarter Technological Advancement, Biomedical Research, pp.107-111, 2016.

[42]. Kavita Sachdeva, Madasu Hanmandlu, Amioy Kumar, *Real Life Applications of Fuzzy Decision Tree*, International Journal of Computer Applications, 2012.

[43]. Kishor Kumar Reddy, Vijaya Babu, *A Survey on Issues of Decision Tree and Non-Decision Tree Algorithms*, International Journal of Artificial Intelligence and Applications for Smart Devices, Vol. 4, No. 1, pp. 9-32, 2016.

[44]. Larose D. T., *Data Mining: Methods and Models*, John Wiley & Sons, Inc. Pubs., Canada, 2006

[45]. Lee C. S. George, Lin C. T, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall International, Inc, 1995.

[46]. Moustakidis S., Mallinis G., Koutsias N., Theocharis J. B., Petridis V., *SVM-Based Fuzzy Decision Trees for Classification of High Spatial Resolution Remote Sensing Images*, Geoscience and Remote Sensing, IEEE, 2012.

[47]. Manish Mehta, Jorma Rissanen, Rakesh Agrawal, *SLIQ: A Fast Scalable Classifier for Data Mining*, IBM Almaden Reseach Center, 1996.

[48]. Manish Mehta, Jorma Rissanen, Rakesh Agrawal, *SPRINT: A Fast Scalable Classifier for Data Mining*, IBM Almaden Reseach Center, 1998.

[49]. Marcos E. Cintra, Maria C. Monard, Heloisa A. Camargo, *A Fuzzy Decision Tree Algorithm Based on C4.5*, Mathware & Soft Computing Magazine. Vol. 20, Num. 1, pp. 56-62, 2013.

[50]. Mariana V. Ribeiro, Luiz Manoel S. Cunha, Heloisa A. Camargo, Luiz Henrique A. Rodrigues, *Applying a Fuzzy Decision Tree Approach to Soil Classification*, Springer International Publishing Switzerland, pp. 87–96, 2014.

[51]. Mingsheng Ying, Bernadette Bouchon Meunier, *Approximate Reasoning with Linguistic Modifiers*, International journal of intelligent systems, vol. 13 pp. 403-418, 1998.

[52]. Narasimha Prasad, Mannava Munirathnam Naidu, *CC-SLIQ: Performance Enhancement with 2k Split Points in SLIQ Decision Tree Algorithm*, International Journal of Computer Science, 2014.

[53]. Olson D. L., Delen D., *Advances Data Mining Techniques*, Springer Pubs., Berlin, Germany, 2008.

[54]. Patil N. at al., *Comparison of C5. 0 & CART classification algorithms using pruning technique*. International Journal of Engineering Research and Technology, ESRSA Publications, 2012.

[55]. Pavel K., Jan P., Václav S., Ajith Abraham, *Fuzzy Classification by Evolutionary Algorithms*, pp. 313-318, IEEE, 2011.

[56]. Paweł Bujnowski, Eulalia Szmidt, Janusz Kacprzyk, *An Approach to Intuitionistic Fuzzy Decision Trees*, 9th Conference of the European Society for Fuzzy Logic and Technology, Published by Atlantis Press, pp. 1253-1260, 2015.

[57]. Peer Fatima, Parveen, Dr. Mohamed Sathik, *Fuzzy Decision Tree based Effective IMine Indexing*, International Journal of Computer Technology and Electronics Engineering (IJCTEE),Volume 1, Issue 2, 2011.

[58]. Perter Rousseeuw, Annick Leroy, *Robust Regression and Outlier Detection*, Wiley, 2003.

[59]. Prade H., Djouadi Y., Alouane B., *Fuzzy Clustering for Finding Fuzzy Partitions of Many-Valued Attribute Domains in a Concept Analysis Perspective*, International Fuzzy Systems Association World Congress and Conference of the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT), pp. 420-425, 2009.

[60]. Quinlan J. R., *Induction of decision trees*, Machine learning, 1986.

[61]. Quinlan J. R., *Simplifying decision trees*, International Journal of Man-Machine Studies, no. 27, pp. 221-234, 1987.

[62]. Quinlan, J. R. *C4.5: Programs for machine learning*, Morgan kaufmann, 1993.

[63]. Ricardo H. Tajiri, Eduardo Z. Marques, Bruno B. Z., Leonardo S. M., *A New Approach for Fuzzy Classification in Relational Databases,*

Database and Expert Systems Applications, Springer, pp. 511–518, 2011.

[64]. R.C. Barros et al., *Automatic Design of Decision-Tree Induction Algorithms*, Springer Briefs in Computer Science, pp. 7-45, 2015.

[65]. Rolly Intan, Oviliani Yenty Yuliana, Andreas Handojo, *Mining Fuzzy Multidimensional Association Rules Using Fuzzy Decision Tree Induction Approach*, International Journal of Computer and Network Security, 2009.

[66]. Ross T. J., *Fuzzy Logic with Engineering Applications*, John Wiley & Sons Ltd, UK, 2004.

[67]. Salvatore Ruggieri, *Efficient C4.5*, University Di Pisa, 2000.

[68]. Shou-Hsiung Cheng, *An Intelligent Stock-Selecting System Based on Decision Tree Combining Rough Sets Theory*, Springer-Verlag Berlin Heidelberg, pp. 501-508, 2013

[69]. Suzan Kantarci-Savas, Efendi Nasibov, *Fuzzy ID3 algorithm on Linguistic Dataset by using WABL defuzzification method*, The conference FUZZ-IEEE, Italy, 2017.

[70]. Vitaly Levashenko, Elena Zaitseva, *Fuzzy Decision Trees in Medical Decision Making Support System*, Proceedings of the Federated Conference on Computer Science and Information Systems pp. 213–219, IEEE, 2012.

[71]. V. Barnett, T. Lewis, *Outliers in Statistical Data*, Wiley, 1994.

[72]. Ying H., *General Tagaki-Sugeno fuzzy systems with simplifier linear rule consequent are universal controllers, models and filters*, Journal of Information Sciences, no. 108, pp. 91-107, 1998.

[73]. Wang T., Lee H., *Constructing a Fuzzy Decision Tree by Integrating Fuzzy Sets and Entropy*, ACOS'06 Proceedings of the 5th WSEAS international conference on Applied computer science, World Scientific and Engineering Academy and Society, USA, pp. 306-311, 2006.

[74]. Wei-Yin Loh , *Classification and regression trees*, John Wiley & Sons, Inc. Volume 1, 2011.

[75]. Wei-Yuan Cheng, Chia-Feng Juang, *A Fuzzy Model With Online Incremental SVM and Margin-Selective Gradient Descent Learning for Classification Problems*, IEEE Transactions on Fuzzy systems, vol. 22, no. 2, pp 324-337, 2014.

[76]. Yahmada K., Phuong N. H., Cuong B. C., *Fuzzy inference methods emploing T-norm with threshold and their implementation*. J. Advanced Computational Intelligence and Intel. Informatics 7, pp. 362 - 369, 2003.

[77]. Yakun Hu, Dapeng Wu, Antonio Nucci, *Fuzzy-Clustering-Based Decision Tree Approach for Large Population Speaker Identification*, IEEE, pp. 1-13, 2010.

[78]. Yi Yang, Wenguang Chen, *Taiga: Performance Optimization of the C4.5 Decision Tree Construction Algorithm*, IEEE - Tsinghua Science and Technology, Volume 21, Number 4, pp. 415-425, 2016.

[79]. Zadeh L. A., *Fuzzy sets*, Information and Control 8, pp.338-358, 1965.

[80]. Zadeh L. A., *A theory of approximate reasoning*, In J. E. Hayes, D. Michie, and L. I. Mikulich editors, Machine intelligence, Elsevier, Amsterda, pp.149-194, 1979.

[81]. Zadeh L. A., *Fuzzy sets and fuzzy information granulation theory*, Beijing Normal University Press, China, 2000.

[82]. Zahra Mirzamomen, Mohammadreza Kangavari, *Fuzzy Min-Max Neural Network Based Decision Trees*, University of Science and Technology, Tehran, Iran, 2015.

[83]. Zeinalkhani M., Eftekhari M., *Comparing Different Stopping Criteria For Fuzzy Decision Tree Induction Through IDFID3*, Iranian Journal Of Fuzzy Systems Vol. 11, No. 1, pp. 27-48, 2014.

[84]. Zengchang Q., Jonathan Lawry, *Linguistic Decision Tree Induction*, Department of Engineering Mathematics, University of Bristol, United Kingdom, 2007.

[85]. Zengchang Qin, Yongchuan Tang, *Linguistic Decision Trees for Classification*, Uncertainty Modeling for Data Mining, Springer, pp 77-119, 2014.

[86]. Zhang, J., Honavar, *Learning Decision Tree Classifiers from Attribute-Value Taxonomies and Partially Specified Data*, Proceedings of the International Conference on Machine Learning. Washington DC, 2003.

[87]. Zhihao Wang, Junfang Wang, Yonghua Huo, Yanjun Tuo, Yang Yang, *A Searching Method of Candidate Segmentation Point in SPRINT Classification*, Journal of Electrical and Computer Engineering, Hindawi Publishing Corporation, 2016.

[88]. Ziarko W., *Dependency Analysis and Attribute Reduction in the Probabilistic Approach to Rough Sets*, Feature Selection for Data and Pattern Recognition, Springer, pp. 93-111, 2015.