BỘ GIÁO DỤC VÀ ĐÀO TẠO VIỆN KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

VIỆN CÔNG NGHỆ THÔNG TIN

DƯƠNG THĂNG LONG

PHƯƠNG PHÁP XÂY DỰNG HỆ MỜ DẠNG LUẬT VỚI NGỮ NGHĨA DỰA TRÊN ĐẠI SỐ GIA TỬ VÀ ỨNG DỤNG TRONG BÀI TOÁN PHÂN LỚP

LUẬN ÁN TIẾN SĨ TOÁN HỌC

HÀ NỘI - 2010

BỘ GIÁO DỤC VÀ ĐÀO TẠO VIỆN KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

VIỆN CÔNG NGHỆ THÔNG TIN

DƯƠNG THĂNG LONG

PHƯƠNG PHÁP XÂY DỰNG HỆ MỜ DẠNG LUẬT VỚI NGỮ NGHĨA DỰA TRÊN ĐẠI SỐ GIA TỬ VÀ ỨNG DỤNG TRONG BÀI TOÁN PHÂN LỚP

Chuyên ngành: BẢO ĐẢM TOÁN HỌC CHO MÁY TÍNH VÀ HỆ THỐNG TÍNH TOÁN

Mã số: 62.46.35.01

LUẬN ÁN TIẾN SĨ TOÁN HỌC

NGƯỜI HƯỚNG DẪN KHOA HỌC:
1. PGS. TSKH. NGUYỄN CÁT HỒ
2. TS. TRẦN THÁI SƠN

HÀ NỘI - 2010

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các kết quả được viết chung với các tác giả khác đều được sự đồng ý của đồng tác giả trước khi đưa vào luận án. Các kết quả trong luận án là trung thực và chưa từng được công bố trong bất kỳ công trình nào khác.

Tác giả

Dương Thăng Long

LỜI CẢM ƠN

Luận án được hoàn thành dưới sự hướng dẫn tận tình và nghiêm khắc của PGS. TSKH. Nguyễn Cát Hồ và TS. Trần Thái Sơn. Lời đầu tiên, tác giả xin bày tỏ lòng kính trọng và biết ơn sâu sắc tới hai Thầy.

Xin chân thành gửi lời cảm ơn tới TS. Vũ Như Lân, PGS. TS. Đặng Thành Phu, PGS. TSKH. Bùi Công Cường, PGS. TS. Phan Trung Huy, PGS. TS. Vũ Chấn Hưng về những đóng góp quý báu trong quá trình nghiên cứu cũng như trong thời gian hoàn thành luận án.

Tác giả xin chân thành gửi lời cảm ơn đến Ban lãnh đạo Viện Công nghệ thông tin, Phòng Đào tạo sau đại học, Phòng Các hệ chuyên gia và tính toán mềm đã tạo điều kiện thuận lợi trong quá trình học tập, nghiên cứu và hoàn thành luận án.

Xin cảm ơn Ban giám hiệu Viện Đại học Mở Hà Nội, Ban chủ nhiệm khoa Công nghệ Tin học và các Phòng chức năng trong Viện đã quan tâm giúp đỡ, tạo điều kiện để tác giả có thể thực hiện kế hoạch nghiên cứu đảm bảo tiến độ.

Cảm ơn các anh chị phòng Các hệ chuyên gia và tính toán mềm - Viện Công nghệ thông tin, các đồng nghiệp thuộc Khoa Công nghệ Tin học - Viện Đại học Mở Hà Nội đã động viên và trao đổi kinh nghiệm trong qúa trình hoàn thành luận án.

Cuối cùng, tác giả xin chân thành cảm ơn các thành viên trong Gia đình, những người luôn dành cho tác giả những tình cảm nồng ấm và sẻ chia những lúc khó khăn trong cuộc sống, luôn động viên giúp đỡ tác giả trong quá trình nghiên cứu. Luận án cũng là món quà tinh thần mà tác giả trân trọng gửi tặng đến các thành viên trong Gia đình.

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Các ký hiệu:

$\mathcal{A}X$	Đại số gia tử tuyến tính
$\underline{\mathcal{A}X}$	Đại số gia tử tuyến tính đầy đủ
$\boldsymbol{\mathcal{A}x^2}$	Đại số 2 gia tử
$\mu(h)$, $fm(x)$	Độ đo tính mờ gia tử h và của hạng từ x
υ	Giá trị định lượng theo điểm của giá trị ngôn ngữ
$\mu_A(v)$	Hàm định lượng của giá trị ngôn ngữ A (đo độ thuộc của v)
$sm(x,y)$	Hàm xác định mức độ gần nhau của hai hạng từ x và y
$\mathcal{I}$	Khoảng tính mờ của giá trị ngôn ngữ
$\boldsymbol{X}_k$	Tập các hạng từ có độ dài đúng k
$\boldsymbol{X}_{(k)}$	Tập các hạng từ có độ dài không quá k
$\boldsymbol{I}_k$	Hệ khoảng tính mờ mức k của các giá trị ngôn ngữ
$\boldsymbol{I}_{(k)}$	Hệ khoảng tính mờ từ mức 1 đến mức k của các giá trị ngôn ngữ
$\mathcal{I}^g$	Khoảng tương tự bậc g của giá trị ngôn ngữ
$\mathfrak{S}_{(k)}$	Hệ khoảng tương tự ở mức k của các giá trị ngôn ngữ

Các chữ viết tắt:

ĐSGT	Đại số gia tử
ĐS2GT	Đại số 2 gia tử
SGA	Simulated Annealing - Genetic Algorithm
IFRG1	Initial Fuzzy Rules Generation 1
IFRG2	Initial Fuzzy Rules Generation 2
HAFRG	Hedge Algebras based Fuzzy Rules Generation
FPO-SGA	Fuzzy Parameters Optimization - SGA
RBO-SGA	Rule base Optimization - SGA

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH

MỞ ĐẦU

Trong cuộc sống loài người, ngôn ngữ được hình thành một cách tự nhiên để giải quyết nhu cầu trao đổi thông tin với nhau. Hơn thế, nó 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. Thật đáng tiếc, thế giới thực thì vô hạn trong khi ngôn ngữ của chúng ta lại hữu hạn, tất yếu sẽ xuất hiện những cụm từ không chính xác hoặc mơ hồ. Tuy nhiên, khả năng của con người thật tài tình, bằng những tư duy, lập luận dựa trên nền hữu hạn của ngôn ngữ đã xây dựng, khám phá vô vàn các tri thức khoa học, khai thác và cải tạo được thế giới hiện thực, nhằm thúc đẩy xã hội loài người ngày một phát triển mạnh mẽ, tốt đẹp và hoàn thiện hơn. Đó là điều không thể phủ nhận sức mạnh của ngôn ngữ, trái lại nó rất hữu ích cho nhân loại.

Ngày nay với sự phát triển vượt bậc của khoa học công nghệ, nhiều thiết bị máy móc được tạo ra nhằm giúp con người giải phóng sức lao động, không chỉ lao động chân tay mà còn cả lao động trí óc. Dĩ nhiên, các thiết bị máy móc đó phải càng "thông minh", có khả năng tư duy, lập luận và sự sáng tạo kiểu như bộ não người. Để thực hiện điều này, rất nhiều nhà khoa học đã và đang nghiên cứu cả về lý thuyết lẫn ứng dụng, đưa ra các phương pháp, các quy trình nhằm kế thừa, mô phỏng khả năng của con người vào các thiết bị máy móc. Trước hết, các nhà khoa học đã phải hình thức hóa toán học các vấn đề ngôn ngữ và xử lý ngôn ngữ mà con người vẫn làm. Người đi tiên phong trong lĩnh vực này là Lotfi A. Zadeh. Trong [80], ông đã đề xuất khái niệm mờ từ những khái niệm mơ hồ, không rõ ràng, không chắc chắn và hình thức hóa toán học nó bằng tập mờ (*fuzzy set*), xác định bởi các hàm thuộc (*membership function*). Trên cơ sở đó, lý thuyết tập mờ được hình thành làm nền tảng cho các phương pháp mô phỏng tư duy lập luận của con người, cho phép biểu diễn và thao tác tính toán trong các mô hình ứng dụng.

Dựa trên lý thuyết tập mờ của L.A. Zadeh, các nhà khoa học đã tiếp cận và phát triển theo nhiều hướng khác nhau, cả về lý thuyết lẫn ứng dụng thực tiễn.

Chúng ta có thể tìm thấy các kết quả này qua các công trình của D. Dubois, H. Prade, C.S. George Lee, H.J. Zimmermann, T.J. Ross, R. Fuller, J.J. Buckley, R. Kruse, D. Nauck, N.K. Kasabov, W. Pedrycz,... [15], [22], [25], [48], [52], [55], [69], [72], [82]. Trong đó, phải kể đến các phương pháp lập luận xấp xỉ mà khái niệm biến ngôn ngữ (*linguistic variable*, trong [81]) và lôgíc mờ (*fuzzy logic*, trong [2], [81]) đóng vai trò then chốt, nhằm mô phỏng quá trình lập luận của con người. 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. Do đó hầu như không thể có một mô hình toán học hoàn hảo để mô phỏng cơ chế suy luận này.

Quá trình lập luận của con người nói chung và lập luận xấp xỉ nói riêng là quá trình tìm kiếm những kết luận không chắc chắn từ các giả thiết không chắc chắn theo cách gần đúng. Các phương pháp lập luận xấp xỉ thường được xây dựng dựa trên các phát biểu dưới dạng luật "If ... then ...", trong đó phần giả thiết (hay gọi là vế trái của luật) gồm nhiều điều kiện kết hợp với nhau bằng từ "*and*" (phép và). Các luật mờ này được chia làm hai dạng, trên mỗi dạng có các phương pháp lập luận được xây dựng tương ứng:

- Dạng luật Mamdani [55]: phần kết luận của mỗi luật là một khái niệm mờ và biểu diễn bởi một hàm thuộc giải tích. Trong dạng này, có hai phương pháp lập luận được xây dựng: Phương pháp thứ nhất, theo truyền thống, xem mỗi luật là một quan hệ mờ và kết nhập chúng thành một quan hệ mờ chung $\mathcal{R}$, đóng vai trò là một toán tử. Lập luận tức là tìm kiếm đầu ra B' cho mỗi đầu vào A', $B' = \mathcal{R}(A')$. Với rất nhiều cách chọn các phép *t-norm*, *t-conorm* và *kéo theo* để tính toán, mỗi cách chọn như vậy sẽ cho kết quả B' khác nhau. Nhìn chung không thể nói cách chọn các phép toán như thế nào là tốt nhất mà phụ thuộc vào từng bài toán cụ thể và trực quan cảm nhận của người giải bài toán đó. Điều này rất phù hợp với lập luận xấp xỉ và tạo tính mềm dẻo trong ứng dụng của phương pháp. Trong phương pháp lập luận thứ hai, mỗi luật mờ được xem như một điểm trong không gian ngôn ngữ, xây dựng các ánh

xạ định lượng ngữ nghĩa cho các giá trị ngôn ngữ để chuyển các điểm đó về không gian thực tạo thành một "*siêu lưới*". Thực hiện nội suy trên siêu lưới này để tìm kết quả đầu ra đối với một đầu vào cho trước.

- Dạng luật Tagaki-Sugeno [79]: phần kết luận của luật mờ là một giá trị rõ, xác định bởi một hàm giải tích hay thậm chí là một giá trị hằng. Dạng này bước đầu được các tác giả đề xuất trong các ứng dụng điều khiển, hiện nay nhiều nhà nghiên cứu đã ứng dụng trong các bài toán khai phá dữ liệu [10], [30]-[33], [42]-[47], [60]. Các phương pháp lập luận cũng được xây dựng trong dạng này: Thứ nhất, luật có mức "*đốt cháy*" dữ liệu đầu vào cao nhất sẽ được chọn và kết quả lập luận là phần kết luận của luật đó. Đây gọi là phương pháp lập luận *single-winner-rule*. Thứ hai, các luật đóng vai trò "*bầu cử*" (*vote*) cho mẫu dữ liệu đối với lớp của vế phải luật dựa trên mức đốt cháy của luật đối với dữ liệu đó, lớp nào có tổng mức đốt cháy cao nhất sẽ được dùng để phân lớp cho dữ liệu đầu vào tương ứng. Phương pháp lập luận này gọi là *weighted-vote*. Hệ luật mờ dạng Tagaki-Sugeno cùng với hai phương pháp lập luận *single-winner-rule* và *weighted-vote* khá trực quan, không phải khử mờ kết quả lập luận, rất phù hợp trong việc xây dựng các mô hình ứng dụng của một số bài toán trong khai phá dữ liệu như nhiều tác giả đã nghiên cứu [10], [12], [17], [20], [27], [30]-[33], [42]-[47], [60].

Nhìn chung, cho dù hệ các luật mờ được biểu diễn bằng cách nào cùng với các phương pháp lập luận được xây dựng tương ứng thì lý thuyết tập mờ vẫn được xem như nền tảng cho các phương pháp lập luận xấp xỉ. Nhưng bản thân lý thuyết tập mờ rất khó để mô phỏng hoàn chỉnh cấu trúc ngôn ngữ mà con người vẫn sử dụng để suy luận, cho dù cách tiếp cận này đã được ứng dụng thành công trên rất nhiều lĩnh vực của cuộc sống. Vì rằng 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. Chẳng hạn, về mặt ngữ nghĩa chúng ta luôn cảm nhận được "*yếu*" nhỏ hơn "*khỏe*", "*cao*" lớn hơn "*thấp*" nhưng hàm thuộc của chúng lại không sánh được với nhau. Mặt khác, trong [81] đã chỉ ra tập các khái niệm mờ không đóng đối với một số các phép toán trên các tập mờ. Vì vậy trong quá trình lập luận nhiều khi người 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 rất lớn và sai số cho quá trình. Hơn nữa, trong [9] chỉ ra rằng 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 (hay lôgíc mờ theo nghĩa Zadeh) còn thiếu một cơ sở đại số làm nền tảng.

Nhằm khắc khắc phục phần nào những nhược điểm trên, năm 1990, N.C. Ho & W. Wechler trong [37] đã khởi xướng phương pháp tiếp cận đại số đến cấu trúc tự nhiên của miền giá trị của các biến ngôn ngữ. 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 [38], ĐSGT mịn hóa [36], ĐSGT mở rộng đầy đủ [5], ĐSGT PN-không thuần nhất [9]. Trong đó, tiêu biểu là ĐSGT mịn hóa cùng với việc trang bị khái niệm độ đo tính mờ của các giá trị ngôn ngữ và phương pháp định lượng ngữ nghĩa [35]. Trên cơ sở đó, các phương pháp lập luận xấp xỉ dựa trên ĐSGT và ứng dụng trong một số lĩnh vực được các tác giả phát triển, có thể kể đến như phương pháp lập luận sử dụng mạng nơron trong điều khiển mờ [4], ứng dụng trong cơ sở dữ liệu mờ [3], lập luận bằng nội suy gia tử có tối ưu tham số và ứng dụng trong điều khiển mờ [8], [39]. Những kết quả này, dù chưa nhiều, nhưng rất khả quan và cho thấy ý nghĩa cũng như thế mạnh của ĐSGT trong ứng dụng.

Bên cạnh đó, sự bùng nổ của thời đại thông tin như hiện nay, lượng thông tin dữ liệu được tạo ra hàng ngày là rất lớn trong mọi lĩnh vực của cuộc sống. 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 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 nhu cầu đó, các nhà nghiên cứu đã đề xuất, nghiên cứu và phát triển các phương pháp mới trong khai phá dữ liệu (*data mining*). Các

bài toán được biết đến trong lĩnh vực này như phân lớp và nhận dạng mẫu (*classification*), hồi quy và dự báo (*regression*), phân cụm (*clustering*), khai phá luật kết hợp (*association rules*),... [15], [18], [27], [48], [63], [54], [69] với rất nhiều mô hình theo tiếp cận dựa trên tập mờ được đề xuất. Trong đó tiêu biểu là các mô hình dưới dạng hệ các luật mờ ứng dụng cho bài toán phân lớp được nghiên cứu khá mạnh mẽ, các kết quả rất phong phú [10], [12], [16], [17], [20], [23], [24], [26], [30]-[33], [40]-[47], [50], [53], [56], [58]-[60], [66], [74], [77]. Tuy nhiên các mô hình này đều tiếp cận dựa trên lý thuyết tập mờ và lôgíc mờ, đã gặp phải không ít những hạn chế mà xuất phát từ bản thân nội tại của lý thuyết tập mờ:

- Các phương pháp xây dựng hệ luật dựa trên tập mờ có sự tách biệt giữa các giá trị ngôn ngữ với tập mờ biểu diễn ngữ nghĩa của chúng đối với một bài toán, thậm chí một số phương pháp sử dụng thuật toán tìm kiếm tối ưu các tham số của các tập mờ đã làm méo ngữ nghĩa của các giá trị ngôn ngữ, cho dù đã đưa ra những ràng buộc trong khi tìm kiếm. Kết quả các tập mờ khó phản ánh ngữ nghĩa của các giá trị ngôn ngữ tương ứng, điều này được thể hiện trong [10], [50].

- Một số phương pháp khác trong [42]-[47], [60] lại thiết lập các tập mờ của các giá trị ngôn ngữ một cách cố định, theo chủ quan của con người. Trong khi, một giá trị ngôn ngữ sẽ mang ngữ nghĩa tương đối khác nhau trong các bài toán khác nhau. Chẳng hạn, nói về thời tiết thì từ "*rất lạnh*" mang ngữ nghĩa với nhiệt độ vào khoảng 10^oC, nhưng khi chỉ nhiệt độ cơ thể người thì từ "*rất lạnh*" lại mang ngữ nghĩa vào khoảng 35^oC.

- Các phương pháp tìm kiếm tối ưu tham số mờ kết quả khó phản ánh ngữ nghĩa của giá trị ngôn ngữ tương ứng, hơn nữa nó có thể tạo ra không gian tìm kiếm rất lớn các tham số. Điều này làm giảm tốc độ hội tụ của quá trình tìm kiếm cũng như giảm hiệu quả của phương pháp.

Mặt khác, về phía ĐSGT, việc áp dụng phương pháp định lượng ngữ nghĩa theo điểm sẽ không còn phù hợp trong các mô hình ứng dụng phân lớp. Miền dữ liệu của các thuộc tính của bài toán thường liên tục trong khi hệ các luật mờ được xây dựng lại rời rạc, do đó cần một phương pháp định lượng ngữ nghĩa các giá trị

ngôn ngữ trong ĐSGT phải liên tục trong miền ngữ nghĩa của nó. Hơn nữa, khi 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 là đủ nhưng chỉ áp dụng ở một mức (các giá trị ngôn ngữ có số lượng gia tử giống nhau), sẽ bỏ qua các giá trị ngôn ngữ mức dưới (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.

Để khắc phục những vấn đề trên, lần đầu tiên, trong luận án này đề xuất phương pháp ứng dụng ĐSGT vào xây dựng các mô hình cho bài toán phân lớp trong lĩnh vực khai phá dữ liệu. Trong ĐSGT, với tính chất sánh được của các giá trị ngôn ngữ đã tạo nên ràng buộc về ngữ nghĩa trong các phương pháp tìm kiếm tối ưu tham số, không làm biến dị tập mờ của chúng. Thông thường, thực tế các mô hình ứng dụng cho bài toán phân lớp với số lượng các giá trị ngôn ngữ không nhiều, số gia tử ít hoặc thậm chí không sử dụng gia tử [50], [10], [42]. Và để giảm bớt không gian tìm kiếm tối ưu các tham số cho mô hình cũng như đảm bảo tính bình đẳng trong việc xem xét các giá trị ngôn ngữ, những cải tiến về một số vấn đề trong ĐSGT được đề xuất nhằm đem lại ứng dụng đạt hiệu quả cao.

Với ý nghĩa như vậy, luận án đặt ra những mục tiêu nghiên cứu cụ thể sau đây:

1) Khảo sát các tính chất, đặc trưng của các giá trị ngôn ngữ cũng như các vấn đề trong ĐSGT nhằm ứng dụng vào việc xây dựng các luật mờ cho bài toán phân lớp.

2) Với những yêu cầu đặt ra đối với việc xây dựng hệ luật mờ cho bài toán phân lớp, luận án sẽ thiết kế các phương pháp tìm kiếm tối ưu xấp xỉ để lựa chọn bộ tham số mờ gia tử đủ tốt và tìm kiếm hệ luật mờ đủ tốt cho ứng dụng.

3) Chọn một số bài toán phân lớp từ đơn giản đến phức tạp để ứng dụng và kiểm chứng cho phương pháp được xây dựng thông qua việc đánh giá và so sánh với các phương pháp khác.

Với nhiệm vụ đặt ra, luận án đã đạt được một số kết quả đóng góp vào việc nghiên cứu mở rộng ứng dụng cho ĐSGT. Có thể khái quát các kết quả chính như sau:

- Nghiên cứu sâu về đại số 2 gia tử (ĐS2GT), tức là ĐSGT chỉ gồm một gia tử dương và một gia tử âm, và khảo sát các tính chất của nó. Khảo sát tính chất kế thừa ngữ nghĩa và quan hệ ngữ nghĩa của các giá trị ngôn ngữ. Giới thiệu khái niệm khoảng tương tự của các giá trị ngôn ngữ và xây dựng hệ khoảng tương tự cho một tập các giá trị ngôn ngữ. Trên cơ sở ĐS2GT, chúng ta khẳng định hệ khoảng tương tự luôn tồn tại và có thể ứng dụng xấp xỉ cho mọi quá trình thực.

- Xây dựng hai phương pháp sinh luật mờ trực tiếp từ tập dữ liệu mẫu cho bài toán phân lớp. Một thuật toán dựa trên hệ khoảng tính mờ và một thuật toán dựa trên hệ khoảng tương tự của các giá trị ngôn ngữ. Các luật sinh ra trong cả hai phương pháp này đều thực hiện theo "*vết*" dữ liệu mang ngữ nghĩa của các giá trị ngôn ngữ. Trên cơ sở quan hệ ngữ nghĩa của các giá trị ngôn ngữ, luận án đã đưa ra phép kết nhập các luật mờ áp dụng cho việc rút gọn hệ luật. Bên cạnh đó, phương pháp sàng dựa trên các tiêu chuẩn đánh giá cũng được áp dụng để rút gọn hệ luật.

- Xây dựng phương pháp thiết kế ngôn ngữ cho bài toán thông qua việc tìm kiếm tối ưu tham số mờ gia tử cho mô hình dựa trên giải thuật di truyền (*Genetic Algorithm - GA*) kết hợp thuật toán mô phỏng tôi luyện (*Simulated Annealing - SA*), từ kết quả đó áp dụng phương pháp sinh tập luật mờ phân lớp và thiết kế tiếp thuật toán tìm kiếm hệ luật tối ưu trên tập luật này.

- Ứng dụng mô hình vào 4 bài toán phân lớp rất đặc trưng với tập dữ liệu cung cấp bởi Đại học California - Irvin, được nhiều tác giả dùng để thử nghiệm cho các mô hình phân lớp. Đánh giá và so sánh kết quả với các phương pháp khác cho thấy tính hiệu quả của mô hình trong luận án.

Về bố cục, luận án bao gồm phần mở đầu, 4 chương, phần kết luận và tài liệu tham khảo.

Chương 1: Trình bày các vấn đề cơ bản dùng trong luận án như tập mờ và các phép toán trong lôgíc mờ, khái niệm về biến ngôn ngữ, mô hình hệ mờ dạng luật và tóm tắt phương pháp lập luận xấp xỉ truyền thống trên mô hình đó. Trình bày các khái niệm, tính chất trong ĐSGT, vấn đề định lượng ngữ nghĩa theo điểm các giá trị ngôn ngữ và ứng dụng vào việc xây dựng phương pháp lập luận xấp xỉ bằng nội suy gia tử dựa trên mạng nơron. Cũng trong chương này, giới thiệu tổng quan về bài toán phân lớp trong khai phá dữ liệu và phương pháp giải bài toán bằng mô hình hệ mờ dạng luật.

Chương 2: Khảo sát các tính chất của ĐS2GT và xây dựng hệ khoảng tương tự cho tập các giá trị ngôn ngữ. Trong ĐS2GT, luận án khẳng định luôn tồn tại hệ khoảng tương tự như vậy và có thể ứng dụng xấp xỉ cho mọi quá trình thực. Trên cơ sở của hệ khoảng tương tự, luận án đã đề xuất phương pháp xây dựng hệ luật mờ ứng dụng cho bài toán phân lớp (thuật toán ***IFRG*2**). Bên cạnh đó, đối với ĐSGT tuyến tính thông thường (không hạn chế số gia tử), luận án cũng đề xuất thêm phương pháp xây dựng hệ luật mờ phân lớp dựa trên hệ khoảng tính mờ của các giá trị ngôn ngữ (thuật toán ***IFRG*1**). Cả hai phương pháp xây dựng hệ luật mờ này đều được khẳng định là có độ phức tạp đa thức đối với kích thước của tập dữ liệu mẫu trong bài toán. Cũng trong chương này, luận án khảo sát tính chất kế thừa ngữ nghĩa và quan hệ ngữ nghĩa của các giá trị ngôn ngữ và xây dựng phép kết nhập để rút gọn hệ luật mờ. Bên cạnh đó, phương pháp sàng theo tiêu chuẩn đánh giá trên luật để rút gọn hệ luật cũng được áp dụng trong chương này. Các phương pháp xây dựng và rút gọn hệ luật mờ đều được minh họa bằng các ví dụ khá trực quan để kiểm tra đánh giá.

Chương 3: Trong chương này, luận án xem xét bài toán tối ưu tham số cũng như tối ưu hệ luật. Dựa trên giải thuật di truyền kết hợp thuật toán mô phỏng tôi luyện, thiết kế hai phương pháp tối ưu: Thứ nhất là thuật toán ***FPO-SGA*** để tìm kiếm bộ tham số mờ gia tử tối ưu cho mô hình được đề xuất đối với một bài toán ứng dụng. Thứ hai là thuật toán ***RBO-SGA*** để tìm kiếm hệ luật tối ưu. Ở đây, các ví dụ minh họa cho phương pháp tối ưu được sử dụng để đánh giá, so sánh kết quả với

trường hợp không tối ưu trong Chương 2 cho thấy tính ưu việt của phương pháp tối ưu cũng như so sánh với kết quả của các tác giả khác.

Chương 4: Lựa chọn 4 bài toán phân lớp từ đơn giản đến phức tạp để ứng dụng cho mô hình trong luận án. Bài toán phân lớp các loại hoa (*IRIS*) đơn giản nhất trong số 4 bài toán này, áp dụng cả hai phương pháp xây dựng hệ luật mờ (***IFRG*1** và ***IFRG*2**). Các bài toán còn lại gồm phân lớp các loại rượu (*WINE*), phân lớp các loại kính (*GLASS*) và phân lớp các loại men sinh học (*YEAST*) đều áp dụng phương pháp xây dựng hệ luật dựa trên ĐS2GT (thuật toán ***IFRG*2**) bởi số thuộc tính và số mẫu dữ liệu khá nhiều, sự phức tạp trong phân bố dữ liệu giữa các lớp. Các kết quả ứng dụng được thiết kế trong nhiều kịch bản khác nhau, nhằm minh chứng cho sự ổn định, tính hiệu quả của phương pháp. Các kết quả này được so sánh với các kết quả của các tác giả khác và đều cho thấy hiệu quả rõ rệt của mô hình trong luận án.

Các kết quả chính của luận án đã được công bố trong các công trình [1], [2], [3], [4], [5], [6], [7] (trang 136), các kết quả này cũng được báo cáo và thảo luận tại các hội nghị, hội thảo, Seminar.

CHƯƠNG 1
TỔNG QUAN VÀ NHỮNG KIẾN THỨC CƠ SỞ

1.1 Kiến thức cơ sở về lập luận mờ

1.1.1 Khái niệm mờ và hình thức hóa toán học bằng tập mờ

Thực tế cho thấy 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*,... Hơn nữa, trong [72] B. Russel đã viết: "Tất cả logíc cổ điển luôn giả sử rằng các đối tượng được sử dụng là rõ ràng. Vì thế nó không thể ứng dụng tốt trong cuộc sống trên trái đất này...". Như vậy, rất cần một tiếp cận nghiên cứu mới so với logíc cổ điển.

L. A. Zadeh đã đề xuất hình thức hóa toán học của khái niệm mờ vào năm 1965, từ đó lý thuyết tập mờ được hình thành và ngày càng thu hút nhiều nghiên cứu của các tác giả cũng như phát triển ứng dụng. Bằng các phương pháp tiếp cận khác nhau, các nhà nghiên cứu như Dubois, Prade, Mamdani, Tagaki, Sugeno, Ishibuchi, Herrera… đã đưa ra những kết quả cả về lý thuyết và ứng dụng trong các bài toán điều khiển mờ, khai phá dữ liệu mờ, cơ sở dữ liệu mờ, các hệ hỗ trợ quyết định,... [15], [18], [22], [36], [48], [57], [72], [78], [81].

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*,… tìm cách biểu diễn chúng bằng một khái niệm toán học, được gọi là tập mờ và được định nghĩa như sau.

Định nghĩa 1.1. [82] Cho một tập vũ trụ U với các phần tử ký hiệu bởi x, $U=\{x\}$. Một tập mờ A trên U là tập được đặc trưng bởi một hàm $\mu_A(x)$ mà nó liên kết mỗi phần tử $x \in U$ với một số thực trong đoạn [0,1]. Giá trị hàm $\mu_A(x)$ biểu diễn mức độ thuộc của x trong A. $\mu_A(x)$ là một ánh xạ từ U vào [0,1] và được gọi là hàm thuộc của tập mờ A.

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 số hàm thuộc thông dụng trong ứng dụng của lý thuyết tập mờ:

- Dạng tam giác: $\mu_A(x) = \max(\min((x-a)/(b-a),(c-x)/(c-b)),0)$,

- Dạng hình thang: $\mu_A(x) = \max(\min((x-a)/(b-a),(d-x)/(d-c),1),0)$,

- Dạng Gauss: $\mu_A(x) = exp(-(c-x)^2/(2\sigma^2))$,... trong đó a, b, c, d, σ,... là các tham số của hàm thuộc tương ứng.

Các khái niệm, tính chất, phép toán trong lý thuyết tập kinh điển cũng được mở rộng cho các tập mờ [2], [15], [18], [22], [81]. Theo đó, các phép toán như *t-norm*, *t-conorm*, *negation* và phép *kéo theo* (*implication*),... trong lôgíc mờ được đề xuất, nghiên cứu chi tiết cung cấp cho các mô hình ứng dụng giải các bài toán thực tế.

Một khái niệm quan trọng trong việc tiếp cận giải bài toán phân lớp về sau trong luận án đó là phân hoạch mờ (*fuzzy partition*). Về hình thức, chúng ta định nghĩa như sau.

Định nghĩa 1.2. [70], [49] Cho p điểm cố định $m_1 < m_2 < ... < m_p$ trong tập $U = [a, b] \subset R$. Khi đó tập Φ gồm p tập mờ $A_1, A_2,..., A_p$ (với $\mu_{A_1}, \mu_{A_2},..., \mu_{A_p}$ là các hàm thuộc tương ứng) định nghĩa trên U được gọi là một phân hoạch mờ của U 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) $\forall x \in U$, $\exists k$, sao cho $\mu_{A_k}(x) > 0$ (tất cả mọi điểm trong U đều thuộc một lớp của phân hoạch này với độ thuộc nào đó khác không).

Ngoài ra, các tác giả trong [49] đưa thêm một số điều kiện để đảm bảo phân hoạch mờ là đều và mạnh.

Như vậy, theo định nghĩa, tập các tập mờ là không gian $\mathcal{F}(U,[0,1])$ các hàm từ U vào đoạn [0,1], một không gian tương đối giàu về cấu trúc tính toán mà nhiều nhà nghiên cứu đã sử dụng cho việc mô phỏng phương pháp lập luận của con người.

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. Một lớp rộng các khái niệm mờ có thể mô hình qua các tập mờ mà L. A. Zadeh đã đưa ra gọi là biến ngôn ngữ.

1.1.2 Biến ngôn ngữ

Biến ngôn ngữ, như Zadeh đã viết [81], là các biến mà giá trị của chúng không phải là số mà là các từ hoặc các câu trong ngôn ngữ tự nhiên hoặc nhân tạo. Các giá trị của biến ngôn ngữ được sử dụng "*khi có sự thiếu hụt tính chính xác bề ngoài của những vấn đề phức tạp cố hữu*" (Zadeh [9]).

Về hình thức, biến ngôn ngữ được được định nghĩa như sau.

Định nghĩa 1.3. [81] Biến ngôn ngữ là một bộ năm $(X,T(X),U,R,M)$, trong đó X là tên biến, $T(X)$ là tập các giá trị ngôn ngữ của biến X, U là không gian tham chiếu hay còn gọi là miền cơ sở của biến X, R là một quy tắc cú pháp sinh các giá trị ngôn ngữ trong $T(X)$, M là quy tắc gán ngữ nghĩa biểu thị bằng tập mờ trên U cho các từ ngôn ngữ trong $T(X)$.

Ví dụ 1.1. Cho X là biến ngôn ngữ có tên *AGE*, miền tham chiếu của X là U=[0,120]. Tập các giá trị ngôn ngữ $T(AGE)$={*very old, old, possible old, less old, less young, quite young, more young,...*}. Chẳng hạn với giá trị nguyên thủy *old*, quy tắc gán ngữ nghĩa M cho *old* bằng tập mờ sau:

$$M(old) = \{(u,\mu_{old}(u)) : u \in [0,120]\},$$

trong đó $\mu_{old}(u)$ = max(min(1,(*u*-50)/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 các gia tử *very*, *more or less*,... tương ứng với các phép bình phương *CON*, căn bậc hai *DIL*,... [81]. Ngoài ra, các giá trị ngôn ngữ có chứa liên từ *AND*, *OR*, *NOT* thì chúng được tính toán bởi các toán tử *t-norm*, *t-conorm*, *negation* [2], [15], [22], [72], [81], [82].

Từ những nghiên cứu về biến ngôn ngữ, các tác giả đã đưa ra những đặc trưng cơ bản của chúng gồm [81], [9]:

- Tính phổ quát: các biến ngôn ngữ khác nhau về các giá trị nguyên thủy nhưng ý nghĩa về mặt cấu trúc miền giá trị của chúng vẫn được giữ. Nói cách khác, cấu trúc miền giá trị của hai biến ngôn ngữ cho trước tồn tại một "*đẳng cấu*" sai khác nhau bởi giá trị sinh nguyên thủy.

- Tính độc lập ngữ cảnh của gia tử và liên từ như *AND*, *OR*,...: ngữ nghĩa của các gia tử và liên từ như *AND*, *OR*,... 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 như *AND*, *OR*,... 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 lý thuyết tập mờ cùng với biến ngôn ngữ, các tác giả đã phát triển lý thuyết lập luận xấp xỉ nhằm mô phỏng quá trình suy luận của con người. Trong đó mô hình hệ mờ dạng luật là phương pháp được nghiên cứu và ứng dụng rất mạnh mẽ [15], [18], [22], [48], [52], [57], [72], [75].

1.1.3 Hệ mờ dạng luật và phương pháp lập luận xấp xỉ truyền thống

Hệ mờ áp dụng cho lập luận xấp xỉ được phát triển dựa trên lý thuyết tập mờ, với những ràng buộc nhất định, được xem như là một bộ xấp xỉ vạn năng [55]. Hơn nữa, thế mạnh của hệ mờ là có thể xấp xỉ các hành vi hệ thống mà ở đó các hàm giải tích hoặc các quan hệ dạng số không tồn tại. Vì vậy, hệ mờ có tiềm năng to lớn để ứng dụng vào việc giải quyết các vấn đề của các hệ thống phức tạp như hệ sinh học, hệ xã hội, hệ kinh tế và hệ thống chính trị. Mặt khác, hệ mờ còn có thể ứng dụng trong các hệ thống ít phức tạp, ở đó không cần một giải pháp chính xác mà chỉ cần một giải pháp xấp xỉ nhưng nhanh hơn, hiệu quả hơn và giảm chi phí tính toán.

Trong mô hình hệ mờ dạng luật, mỗi luật mờ thể hiện một tri thức của con người về một bài toán ứng dụng và được biểu diễn dưới dạng "If *Antecedents* then *Consequents*", trong đó *Antecedents* là các điều kiện chứa các từ ngôn ngữ thường được liên kết bởi liên từ "and" và *Consequents* là phần kết luận biểu thị qua các vị từ mờ chứa khái niệm mờ hoặc vị từ kinh điển. Nếu kết luận của luật là khái niệm mờ thì hệ mờ ở dạng Mamdani, ngược lại kết luận là giá trị rõ thì hệ mờ dạng Sugeno [57], [72]. Ví dụ về hai dạng luật mờ tương ứng:

If x_1 is *Large* and x_2 is *Very Small* then y is *Normal*,

If x_1 is *Small* and x_2 is *Large* then y = "Iris-Setosa".

Dưới dạng tổng quát, một hệ mờ dạng luật có *n* đầu vào 1 đầu ra (MISO) thường phát biểu như sau:

If x_1 is A_{i1} and ... and x_n is A_{in} then y is B_i, (1.1)

trong đó x_1, x_2, …, x_n và y là các biến ngôn ngữ thuộc không gian tham chiếu U_1, U_2, ..., U_n và V, A_{ij}, B_i (i = 1,…, M; j = 1,…, n) là các giá trị ngôn ngữ tương ứng.

Các luật mờ này được xây dựng hoặc dựa trên ý kiến chuyên gia về bài toán ứng dụng hoặc sử dụng các kỹ thuật học máy để sinh trực tiếp từ các mẫu dữ liệu thu thập được. Tuy nhiên, không phải bài toán nào cũng có chuyên gia với các ý

kiến đủ để xây dựng một hệ luật, thường phải kết hợp các phương pháp sinh luật đảm bảo tính đầy đủ cho hệ luật đó.

Giải bài toán lập luận xấp xỉ theo mô hình (1.1) là xây dựng một phương pháp lập luận dựa trên các luật mờ để tính toán đầu ra từ các dữ liệu đầu vào tương ứng, tức tìm kết quả B' của γ khi biết giá trị $A'_1, A'_2, ..., A'_n$ tương ứng với các biến $x_1, x_2, \ldots, x_n$. Vì chúng ta đang ở trong môi trường thông tin mờ, không chắc chắn, nên không có một phương pháp lập luận chính xác và duy nhất. Mỗi phương pháp sẽ xuất phát từ một quan sát trực quan nào đó.

Theo phương pháp truyền thống, quy tắc *modus ponens* tổng quát hóa được áp dụng cho hệ mờ dạng (1.1) cùng với việc sử dụng các phép toán lôgíc mờ đã được nhiều tác giả đề cập chi tiết trong [2], [15], [57], [72]. Ở đây chúng ta tóm tắt như sau:

Xét mỗi luật mờ trong (1.1) là một quan hệ mờ R_i trên miền tích Đề-các $\mathcal{U} = U_1 \times U_2 \times ... \times U_n \times V$ với hàm thuộc được xác định bởi:

$$\mu_{Ri} = I(T^n(\mu_{Ai,1}, ..., \mu_{Ai,n}), \mu_{Bi}) \tag{1.2}$$

trong đó $\mu_{Ai,j}$, μ_{Bi} là các hàm thuộc tương ứng với $A_{i,j}$, B_i, T^n là phép *t-norm n*-ngôi và I là phép *kéo theo*. Kết nhập các luật mờ R_i (i = 1, ..., m) của hệ bằng phép *t-conorm* với hàm thuộc μ_R và áp dụng quy tắc suy diễn hợp thành ta có kết quả:

$$\mu_{B'} = \sup_{(u_1,\ldots,u_n,v)\in\mathcal{U}} \left\{ \left[\nabla_{j=1}^{n} \mu_{A'_j}(u_j) \right] \circ \left[\Delta_{i=1}^{M} \left(I\left(\nabla_{j=1}^{n} \mu_{A_{i,j}}(u_j), \mu_{B_i}(v) \right) \right) \right] \right\} \tag{1.3}$$

ở đây ∇ là phép *t-norm*, Δ là phép *t-conorm* và $\circ$ là *min* hoặc *prod*.

Công thức (1.3) cho thấy phương pháp lập luận này với những cách chọn các phép *t-norm*, *t-conorm* hay *kéo theo I* dẫn đến những kết quả tính toán tập mờ B' khác nhau. Điều này phù hợp với đặc trưng của lập luận xấp xỉ. Câu hỏi về cách chọn các phép trên như thế nào để có một phương pháp lập luận tốt nói chung không có câu trả lời khẳng định mà phụ thuộc vào từng tình huống ứng dụng cụ thể và được kiểm chứng qua kết quả thực nghiệm.

Mặt khác, hệ luật mờ dạng Sugeno với phần kết luận của các luật là một mệnh đề kinh điển chứa hằng cá thể sẽ trở thành một trường hợp riêng của dạng (1.1) khi chọn đầu ra B_i có hàm thuộc ở dạng đơn tử [72]. Tuy nhiên, luật mờ dạng Sugeno với ưu điểm có thể thể hiện các hành vi cục bộ của hệ thống được ứng dụng và không cần giải mờ sau khi lập luận [57]. Hơn nữa, trong nhiều nghiên cứu của các tác giả như Ishibuchi H., Herrera F., Khotanzad A., Mansoori E.G.,... [42]-[46], [30]-[28], [60], [50] với việc sử dụng các luật mờ có phần kết luận chỉ chứa các giá trị hằng cá thể đã đem lại kết quả rất khả quan. Đây là những lý do thúc đẩy những nghiên cứu hơn nữa về các mô hình ứng dụng hệ luật mờ, đặc biệt trường hợp luật mờ có kết luận chỉ chứa giá trị hằng cá thể sẽ được trình bày tiếp ở những phần sau.

1.2 Đại số gia tử: một số vấn đề cơ bản

1.2.1 Các khái niệm cơ bản về đại số gia tử

Phương pháp lập luận tính toán nhằm giải quyết vấn đề mô phỏng tư duy, lập luận của con người chính là việc chúng ta mượn cấu trúc tính toán rất phong phú của tập tất cả các hàm $\mathcal{F}(U,[0,1])$ để mô phỏng các cách lập luận của con người mà chúng ta thường được thực hiện trên nền ngôn ngữ tự nhiên. Tuy nhiên, trong [2], các tác giả đã chỉ ra rằng tập các giá trị ngôn ngữ của một biến ngôn ngữ sẽ là một cấu trúc đại số đủ giàu để tính toán và nghiên cứu các phương pháp lập luận. Như vậy thay vì mượn cấu trúc của $\mathcal{F}(U,[0,1])$, chúng ta có một khả năng lựa chọn khác là sử dụng cấu trúc đại số của chính các tập các giá trị ngôn ngữ.

Đại số gia tử (ĐSGT) được ra đời do đề xuất của N.C. Ho và W. Wechler vào năm 1990 [37], đến nay đã có nhiều nghiên cứu phát triển và ứng dụng thành công của các tác giả [3], [6]-[9], [36]-[39].

Trong [37], các tác giả đã chứng minh miền ngôn ngữ $\boldsymbol{X} = Dom(x)$ của một biến ngôn ngữ x có thể được tiên đề hóa và được gọi là đại số gia tử và được ký hiệu là $\mathcal{A}x = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \leq)$ trong đó $\boldsymbol{G}$ là tập các phần tử sinh, $\boldsymbol{H}$ là tập các gia tử

(*hedge*) còn "≤" là quan hệ cảm sinh ngữ nghĩa trên $\boldsymbol{X}$. Giả thiết trong $\boldsymbol{G}$ có chứa các phần tử hằng **0**, **1**, $\boldsymbol{W}$ với ý nghĩa là phần tử bé nhất, phần tử lớn nhất và phần tử trung hòa (*neutral*) trong $\boldsymbol{X}$. Ta gọi mỗi giá trị ngôn ngữ $x \in \boldsymbol{X}$ là một hạng từ (*term*) trong ĐSGT.

Nếu tập $\boldsymbol{X}$ và $\boldsymbol{H}$ là các tập sắp thứ tự tuyến tính, khi đó $\mathcal{A}x = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \leq)$ là ĐSGT tuyến tính. Hơn nữa, nếu được trang bị thêm hai gia tử tới hạn là $\boldsymbol{\sum}$ và $\boldsymbol{\Phi}$ với ngữ nghĩa là cận trên đúng và cận dưới đúng của tập $\boldsymbol{H}(x)$ khi tác động lên x, thì ta được ĐSGT truyến tính đầy đủ, ký hiệu $\underline{\mathcal{A}x} = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{\sum}, \boldsymbol{\Phi}, \leq)$. Vì trong luận án chỉ quan tâm đến ĐSGT tuyến tính, kể từ đây nói ĐSGT cũng có nghĩa là ĐSGT tuyến tính.

Khi tác động gia tử $h \in \boldsymbol{H}$ vào phần tử $x \in \boldsymbol{X}$, thì thu được phần tử ký hiệu hx. Với mỗi $x \in \boldsymbol{X}$, ký hiệu $\boldsymbol{H}(x)$ là tập tất cả các hạng từ $u \in \boldsymbol{X}$ sinh từ x bằng cách áp dụng các gia tử trong $\boldsymbol{H}$ và viết $u = h_n \ldots h_1 x$, với $h_n, \ldots, h_1 \in \boldsymbol{H}$.

Tập $\boldsymbol{H}$ gồm các gia tử dương $\boldsymbol{H}^+$ và gia tử âm $\boldsymbol{H}^-$. Các gia tử dương làm tăng ngữ nghĩa của một hạng từ mà nó tác động, còn gia tử âm làm giảm ngữ nghĩa của hạng từ. Không mất tính tổng quát, ta luôn giả thiết rằng $\boldsymbol{H}^- = \{h_{-1} < h_{-2} < ... < h_{-q}\}$ và $\boldsymbol{H}^+ = \{h_1 < h_2 < ... < h_p\}$.

Để ý rằng biểu thức $h_n...h_1u$ được gọi là một biểu diễn chính tắc của một hạng từ x đối với u nếu $x = h_n...h_1u$ và $h_i...h_1u \neq h_{i-1}...h_1u$ với i nguyên và $i \leq n$. Ta gọi 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 $l(x)$.

Ví dụ 1.2. Cho biến ngôn ngữ TRUTH, có $\boldsymbol{G}$ = {**0**, *FALSE*, $\boldsymbol{W}$, *TRUE*, **1**}, $\boldsymbol{H}^-$ = { *Possible* < *Little* } và $\boldsymbol{H}^+$ = { *More* < *Very* }. Khi đó TRUE < *More TRUE* < *Very TRUE*, *Little* TRUE < TRUE,...

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

Định lý 1.1. [37] Cho tập $\boldsymbol{H}^-$ và $\boldsymbol{H}^+$ là các tập sắp thứ tự tuyến tính của ĐSGT $\mathcal{A}X = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \leq)$. Khi đó ta có các khẳng định sau:

(1) Với mỗi $u \in \boldsymbol{X}$ thì $\boldsymbol{H}(u)$ là tập sắp thứ tự tuyến tính.

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

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

Định lý 1.2. [38] 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 $h_{j'} = k_{j'}$ với mọi $j' < j$ (ở đây nếu $j = \min\{n, m\} + 1$ thì hoặc h_j là toán tử đơn vị $\boldsymbol{I}$, $h_j = \boldsymbol{I}$, $j = n + 1 \leq m$ hoặc $k_j = \boldsymbol{I}$, $j = m + 1 \leq n$) 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 so sánh được với nhau khi và chỉ khi $h_j x_j$ và $k_j x_j$ là không so sánh được với nhau.

Trong phần tiếp theo, chúng ta trình bày một số vấn đề của đại số gia tử làm cơ sở cho việc nghiên cứu và phát triển một số mô hình lập luận và ứng dụng về sau.

1.2.2 Vấn đề định lượng ngữ nghĩa trong đại số gia tử

Trong phần này chúng ta xem xét ba vấn đề cơ bản đó là độ đo tính mờ của các giá trị ngôn ngữ (hạng từ), phương pháp định lượng ngữ nghĩa và khoảng tính mờ của các khái niệm mờ.

Tính mờ của các giá trị ngôn ngữ xuất phát từ thực tế rằng một giá trị ngôn ngữ mang ý nghĩa mô tả cho nhiều sự vật và hiện tượng trong thế giới thực, với lý

do tập hữu hạn các giá trị ngôn ngữ không đủ để phản ánh thế giới vô hạn các sự vật hiện tượng [34]. Như vậy khái niệm tính mờ và độ đo tính mờ của một giá trị ngôn ngữ được hình thành và nó là một khái niệm rất khó xác định, đặc biệt trong lý thuyết tập mờ [8]. Tuy nhiên, trong ĐSGT các tác giả đã cho thấy độ đo tính mờ được xác định một cách hợp lý: "*tính mờ của một hạng từ x được hiểu như là ngữ nghĩa của nó vẫn có thể được thay đổi khi tác động vào nó bằng các gia tử*" [34], [35]. Do đó, tập các hạng từ sinh từ x bằng các gia tử sẽ thể hiện cho tính mờ của x và do đó, $\boldsymbol{H}(x)$ có thể sử dụng như là một mô hình biểu thị tính mờ của x và kích thước tập $\boldsymbol{H}(x)$ được xem như độ đo tính mờ của x. Ta có định nghĩa sau về độ đo tính mờ.

Định nghĩa 1.4. [35] Cho $\underline{AX}$ = ($\boldsymbol{X}$, $\boldsymbol{G}$, $\boldsymbol{H}$, $\boldsymbol{\sum}$, $\boldsymbol{\Phi}$, ≤) là một ĐSGT tuyến tính đầy đủ. Ánh xạ $fm : \boldsymbol{X} \to [0,1]$ được gọi là một đo tính mờ của các hạng từ trong $\boldsymbol{X}$ nếu:

(1) fm là đầy đủ, tức là $fm(\boldsymbol{c}^-) + fm(\boldsymbol{c}^+) = 1$ và $\sum_{h\in \boldsymbol{H}} fm(hu) = fm(u)$, $\forall u \in \boldsymbol{X}$;

(2) $fm(x) = 0$, với các x thỏa $\boldsymbol{H}(x) = \{x\}$. Đặc biệt, $fm(\boldsymbol{0}) = fm(\boldsymbol{W}) = fm(\boldsymbol{1}) = 0$;

(3) $\forall x,y \in \boldsymbol{X}$, $h \in \boldsymbol{H}$, $\dfrac{fm(hx)}{fm(x)} = \dfrac{fm(hy)}{fm(y)}$, tỷ số này không phụ thuộc vào x và y,

vì vậy nó được gọi là độ đo tính mờ của các gia tử và được ký hiệu bởi $\mu(h)$.

Trong đó, điều kiện (1) thể hiện tính đầy đủ của các phần tử sinh và các gia tử cho việc biểu diễn ngữ nghĩa của miền thực đối với các biến. (2) thể hiện tính rõ của các hạng từ và (3) có thể được chấp nhận vì chúng ta đã chấp nhận giả thiết rằng các gia tử là độc lập với ngữ cảnh và, do vậy, khi áp dụng một gia tử h lên các hạng từ thì hiệu quả tác động tương đối làm thay đổi ngữ nghĩa của các hạng từ đó là như nhau. Hình vẽ sau (Hình 1.1) minh họa rõ hơn cho khái niệm độ đo tính mờ của biến ngôn ngữ TRUTH (đã xét trong Ví dụ 1.2).

Các tính chất của độ đo tính mờ của các hạng từ và gia tử được thể hiện qua mệnh đề sau:

Mệnh đề 1.1. [35] Với độ đo tính mờ *fm* và μ đã được định nghĩa trong *Định nghĩa 1.4*, ta có:

(1) $fm(\boldsymbol{c}^-) + fm(\boldsymbol{c}^+) = 1$ và $\sum_{h\in H} fm(hx) = fm(x)$;

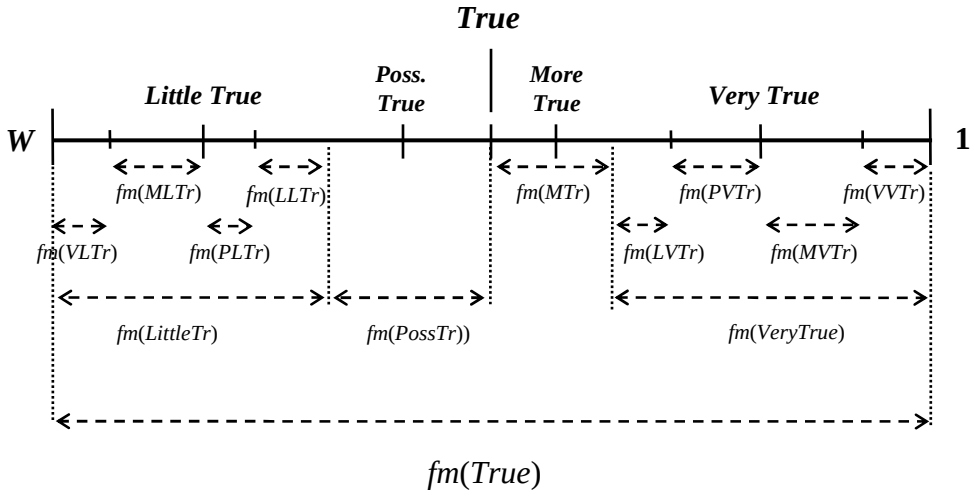
(2) $\sum_{j=-q}^{-1} \mu(h_j) = \alpha$, $\sum_{j=1}^{p} \mu(h_j) = \beta$, với $\alpha, \beta > 0$ và $\alpha + \beta = 1$;

(3) $\sum_{x\in X_k} fm(x) = 1$, trong đó $\boldsymbol{X}_k$ là tập các hạng từ có độ dài đúng k;

(4) $fm(hx) = \mu(h).fm(x)$, và $\forall x\in \boldsymbol{X}$, $fm(\mathcal{W}x) = fm(\boldsymbol{\Phi}x) = 0$;

(5) Cho $fm(\boldsymbol{c}^-)$, $fm(\boldsymbol{c}^+)$ và $\mu(h)$ với $\forall h\in \boldsymbol{H}$, khi đó với $x = h_n...h_1\boldsymbol{c}^{\varepsilon}$, $\varepsilon \in \{-,+\}$, dễ dàng tính được độ đo tính mờ của x như sau:

$$fm(x) = \mu(h_n)...\mu(h_1)fm(\boldsymbol{c}^{\varepsilon}).$$



Hình 1.1: Độ đo tính mờ của biến TRUTH

Thông thường, ngữ nghĩa của các hạng từ thuần túy mang tính định tính. Tuy nhiên, trong nhiều ứng dụng, chúng ta cần giá trị định lượng của các hạng từ này cho việc tính toán và xử lý. Theo tiếp cận của tập mờ, việc định lượng hóa các khái niệm mờ được thực hiện qua các phương pháp khử mờ (*defuzzification*). Đối với ĐSGT, giá trị định lượng của các hạng từ được định nghĩa dựa trên cấu trúc thứ tự ngữ nghĩa của miền giá trị của các biến ngôn ngữ, cụ thể là độ đo tính mờ của các

hạng từ và gia tử. Tuy có nhiều phương pháp xác định giá trị định lượng của các hạng từ dựa trên các tham số này nhưng phải thỏa mãn một số ràng buộc nhất định và được thể hiện trong định nghĩa sau.

Định nghĩa 1.5. [35] Cho $\underline{AX}$ = (***X***, ***G***, ***H***, $\sum$, ***Φ***, ≤) là một ĐSGT tuyến tính đầy đủ. Ánh xạ υ : ***X*** → [0,1] được gọi là một hàm định lượng ngữ nghĩa (*SQM*) của $\underline{AX}$ nếu:

(1) υ là ánh xạ 1-1 từ tập ***X*** vào đoạn [0,1] và đảm bảo thứ tự trên ***X***, tức là $\forall x,y \in \boldsymbol{X}$, $x < y \Rightarrow \upsilon(x) < \upsilon(y)$ và $\upsilon(\mathbf{0}) = 0$, $\upsilon(\mathbf{1}) = 1$.

(2) υ liên tục: $\forall x \in \boldsymbol{X}$, $\upsilon(\boldsymbol{\Phi}x)$ = *infimum* $\upsilon(\boldsymbol{H}(x))$ và $\upsilon(\sum x)$ = *supremum* $\upsilon(\boldsymbol{H}(x))$.

Điều kiện (1) là bắt buộc tối thiểu đối với bất kỳ phương pháp định lượng nào, còn điều kiện (2) đảm bảo tính trù mật của ***H***(***G***) trong ***X***. Dựa trên những ràng buộc này, các tác giả trong [35] đã xây dựng một phương pháp định lượng ngữ nghĩa của các hạng từ trong ĐSGT. Trước hết chúng ta xét định nghĩa về dấu của các hạng từ như sau.

Định nghĩa 1.6. [35] Một hàm dấu *Sign* : ***X*** → {-1,0,1} là một ánh xạ được định nghĩa đệ qui như sau, trong đó $h, h' \in \boldsymbol{H}$ và $c \in \{\boldsymbol{c}^-, \boldsymbol{c}^+\}$:

(1) $Sign(\boldsymbol{c}^-) = -1$, $Sign(\boldsymbol{c}^+) = 1$;

(2) $Sign(h\boldsymbol{c}) = -Sign(\boldsymbol{c})$ nếu h âm đối với c; $Sign(h\boldsymbol{c}) = Sign(\boldsymbol{c})$ nếu h dương đối với ***c***;

(3) $Sign(h'hx) = -Sign(hx)$, nếu $h'hx \neq hx$ và h' âm đối với h; $Sign(h'hx) = Sign(hx)$, nếu $h'hx \neq hx$ và h' dương đối với h;

(4) $Sign(h'hx) = 0$, nếu $h'hx = hx$.

Dựa trên hàm dấu này, chúng ta có tiêu chuẩn để so sánh hx và x.

Mệnh đề 1.2. [35] Với bất kỳ h và x, nếu $Sign(hx) = 1$ thì $hx > x$; nếu $Sign(hx) = -1$ thì $hx < x$ và nếu $Sign(hx) = 0$ thì $hx = x$.

Định nghĩa 1.7. [35] Cho $\underline{AX}$ là một ĐSGT tuyến tính đầy đủ và *fm* là một độ đo tính mờ trên $\boldsymbol{X}$. Ta nói ánh xạ $v : \boldsymbol{X} \to [0,1]$ được cảm sinh bởi độ đo tính mờ *fm* nếu được định nghĩa bằng đệ qui như sau:

(1) $v(\boldsymbol{W}) = \theta = fm(\boldsymbol{c}^-)$, $v(\boldsymbol{c}^-) = \theta - \alpha.fm(\boldsymbol{c}^-) = \beta.fm(\boldsymbol{c}^-)$, $v(\boldsymbol{c}^+) = \theta + \alpha.fm(\boldsymbol{c}^+)$;

(2) $v(h_j x) = v(x) + Sign(h_j x)\left\{\sum_{i=Sign(j)}^{j-Sign(j)} \mu(h_i)\, fm(x) - \omega(h_j x)\mu(h_j x)\, fm(x)\right\}$,

với mọi j, $-q \le j \le p$ và $j \ne 0$, trong đó:

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

(3) $v(\Phi\boldsymbol{c}^-) = 0$, $v(\Sigma\boldsymbol{c}^-) = \theta = v(\Phi\boldsymbol{c}^+)$, $v(\Sigma\boldsymbol{c}^+) = 1$, và với mọi j thỏa $-q \le j \le p$, $j \ne 0$, ta có:

$$v(\Phi h_j x) = v(x) + Sign\,(h_j x)\left\{\sum_{i=Sign\,(j)}^{j-Sign\,(j)} \mu(h_i)\, fm(x)\right\} - \frac{1}{2}\left(1 - Sign\,(h_j x)\right)\mu(h_j)\, fm(x),$$

$$v(\Sigma h_j x) = v(x) + Sign\,(h_j x)\left\{\sum_{i=Sign\,(j)}^{j-Sign\,(j)} \mu(h_i)\, fm(x)\right\} + \frac{1}{2}\left(1 - Sign\,(h_j x)\right)\mu(h_j)\, fm(x).$$

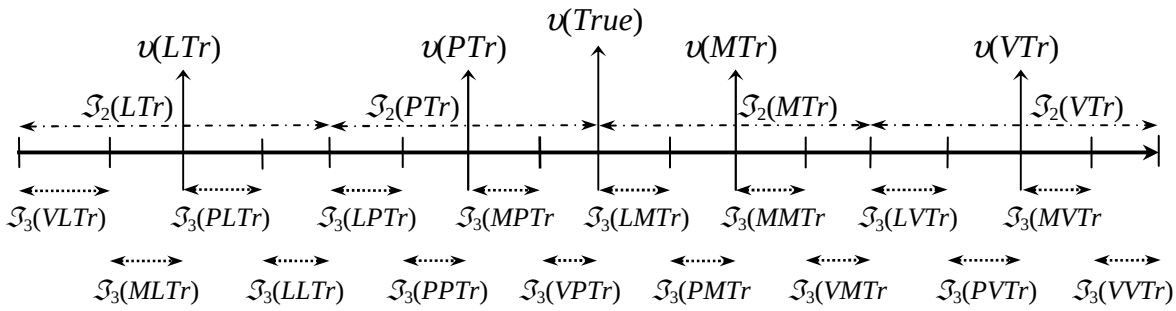
Với định nghĩa này, các tác giả trong [35] đã chứng minh nó thỏa mãn các yêu cầu của một hàm định lượng ngữ nghĩa và đảm bảo tính trù mật của nó đối với các hạng từ của $\underline{AX}$ trong đoạn [0,1] (xem Định lý 1.3).

Một khái niệm rất quan trọng làm cơ sở cho việc nghiên cứu và xây dựng các mô hình ứng dụng về sau đó là khoảng tính mờ (*fuzziness interval*) của các khái niệm mờ. Trong ĐSGT, dựa trên độ đo tính mờ *fm*, chúng ta sẽ định nghĩa khoảng tính mờ của các hạng từ. Gọi $\mathcal{I}t\mathcal{V}([0,1])$ là họ các đoạn con của đoạn [0,1], ký hiệu $|\bullet|$ là độ dài của đoạn "$\bullet$".

Định nghĩa 1.8. Khoảng tính mờ của các hạng từ $x \in \boldsymbol{X}$, ký hiệu $\mathcal{I}_{fm}(x)$, là một đoạn con của [0,1], $\mathcal{I}_{fm}(x) \in \mathcal{Itv}([0,1])$, nếu nó có độ dài bằng độ đo tính mờ, $|\mathcal{I}_{fm}(x)| = fm(x)$, và được xác định bằng qui nạp theo độ dài của x như sau:

(1) Với độ dài của x bằng 1 ($l(x)=1$), tức là $x \in \{\boldsymbol{c}^-, \boldsymbol{c}^+\}$, khi đó $|\mathcal{I}_{fm}(\boldsymbol{c}^-)| = fm(\boldsymbol{c}^-)$, $|\mathcal{I}_{fm}(\boldsymbol{c}^+)| = fm(\boldsymbol{c}^+)$ và $\mathcal{I}_{fm}(\boldsymbol{c}^-) \le \mathcal{I}_{fm}(\boldsymbol{c}^+)$;

(2) Giả sử x có độ dài n ($l(x)=n$) và khoảng tính mờ $\mathcal{I}_{fm}(x)$ đã được định nghĩa với $|\mathcal{I}_{fm}(x)| = fm(x)$. Khi đó tập các khoảng tính mờ $\{\mathcal{I}_{fm}(h_j x)$: $-q \le j \le p$ và $j \ne 0\} \subset \mathcal{Itv}([0,1])$ được xây dựng sao cho nó là một phân hoạch của $\mathcal{I}_{fm}(x)$, và thỏa mãn $|\mathcal{I}_{fm}(h_j x)| = fm(h_j x)$ và có thứ tự tuyến tính tương ứng với thứ tự của tập $\{h_{-q}x, h_{-q+1}x, ..., h_p x\}$, tức là nếu $h_{-q}x > h_{-q+1}x > ... > h_p x$ thì $\mathcal{I}_{fm}(h_{-q}x) > \mathcal{I}_{fm}(h_{-q+1}x) > ... > \mathcal{I}_{fm}(h_p x)$ và ngược lại (xem Hình 1.2). Dễ dàng thấy rằng hệ phân hoạch như vậy luôn tồn tại dựa vào tính chất (1) trong Mệnh đề 1.1.



Hình 1.2: Khoảng tính mờ của các hạng từ của biến TRUTH

Trường hợp độ dài của x bằng k, $l(x) = k$, ta ký hiệu $\mathcal{I}_k(x)$ thay cho $\mathcal{I}_{fm}(x)$, khi đó ta nói khoảng tính mờ của x có độ sâu k (hay khoảng tính mờ mức k). Để thuận tiện về sau, ta ký hiệu:

$\boldsymbol{X}_k$ là tập các hạng từ có độ dài đúng k,

$\boldsymbol{X}_{(k)} = \bigcup_{l=1,...,k} \boldsymbol{X}_l$ là tập tất cả các hạng từ có độ dài từ 1 đến k.

Rõ ràng $\boldsymbol{X} = \bigcup_{k=1}^{\infty} \boldsymbol{X}_k$, và

$\boldsymbol{I}_k = \{\mathcal{I}_k(x)$: $x \in \boldsymbol{X}_k\}$ là tập tất cả các khoảng tính mờ độ sâu k,

$\boldsymbol{I} = \{\mathfrak{I}(x): x \in \boldsymbol{X}\} = \bigcup_{k=1}^{\infty} \boldsymbol{I}_k$.

Tương tự ta cũng có tập $\boldsymbol{I}_{(k)} = \bigcup_{l=1,...,k} \boldsymbol{I}_l$.

Tiếp theo chúng ta xem xét một số tính chất của khoảng tính mờ cũng như cấu trúc của họ tất cả các khoảng tính mờ trong mệnh đề sau. Họ các khoảng tính mờ đóng một vai trò quan trọng trong việc xem xét quan hệ tương tự đối với dữ liệu trong miền tham chiếu của các biến. Ở đây, ta sử dụng khái niệm tựa phân hoạch tức là phân hoạch mà hai tập bất kỳ của nó có nhiều nhất một điểm chung.

Mệnh đề 1.3. Cho $\underline{AX} = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \boldsymbol{\sum}, \boldsymbol{\Phi}, \leq)$ là một ĐSGT tuyến tính đầy đủ:

(1) Nếu $Sign(h_p x') = 1$, thì ta có $\mathfrak{I}(h_{-q}x') \leq \mathfrak{I}(h_{-q+1}x') \leq ... \leq \mathfrak{I}(h_{-1}x') \leq \mathfrak{I}(h_1x') \leq \mathfrak{I}(h_2x') \leq ... \leq \mathfrak{I}(h_px')$, và nếu $Sign(h_p x') = -1$, thì ta có $\mathfrak{I}(h_px') \leq \mathfrak{I}(h_{p-1}x') \leq ... \leq \mathfrak{I}(h_1x') \leq \mathfrak{I}(h_{-1}x') \leq \mathfrak{I}(h_{-2}x') \leq ... \leq \mathfrak{I}(h_{-q}x')$;

(2) Tập $\boldsymbol{I}_k = \{\mathfrak{I}(x): x \in \boldsymbol{X}_k\}$ là một tựa phân hoạch của đoạn [0,1];

(3) Cho một số m, tập $\{\mathfrak{I}(y): y = k_m... k_1x, \forall k_m,... , k_1 \in \boldsymbol{H}\}$ là một tựa phân hoạch của khoảng tính mờ $\mathfrak{I}(x)$;

(4) Tập $\boldsymbol{I}_k = \{\mathfrak{I}(x): x \in \boldsymbol{X}_k\}$ "*mịn*" hơn tập $\boldsymbol{I}_{k-1} = \{\mathfrak{I}(x): x \in \boldsymbol{X}_{k-1}\}$, tức là bất kỳ một khoảng tính mờ trong $\boldsymbol{I}_k$ chắc chắn được chứa bên trong một khoảng của $\boldsymbol{I}_{k-1}$;

(5) Với $x < y$ và $l(x) = l(y)$, thì $\mathfrak{I}(x) \leq \mathfrak{I}(y)$ và $\mathfrak{I}(x) \neq \mathfrak{I}(y)$.

Chứng minh. Các tính chất (2) đến (5) đã được chứng minh trong [35], ở đây ta chứng minh (1). Theo Mệnh đề 1.2, nếu $Sign(h_p x') = 1$ thì ta có $x' \leq h_p x'$. Vì các gia tử trong $\boldsymbol{H}^+$ là so sánh được và $\boldsymbol{H}^+$ và $\boldsymbol{H}^-$ là đối ngược nhau, nên $h_{-q}x' \leq h_{-q+1}x' \leq ... \leq h_{-1}x' \leq x' \leq h_1x' \leq h_2x' \leq ... \leq h_px'$. Từ Định nghĩa 1.8 của khoảng tính mờ ta suy ra $\mathfrak{I}(h_{-q}x') \leq \mathfrak{I}(h_{-q+1}x') \leq ... \leq \mathfrak{I}(h_{-1}x') \leq \mathfrak{I}(h_1x') \leq \mathfrak{I}(h_2x') \leq ... \leq \mathfrak{I}(h_px')$. Chứng minh tương tự với trường hợp $Sign(h_p x') = -1$.■

Dễ dàng suy ra từ mệnh đề trên trong trường hợp các khoảng tính mờ được xét ở dạng nửa đóng, tức là $\mathfrak{I}(x) = (lmp(\mathfrak{I}(x)), rmp(\mathfrak{I}(x))]$, và khoảng tính mờ của hạng từ bé nhất trong phân hoạch ở dạng đóng thì các tựa phân hoạch trong (2), (3)

trở thành các phân hoạch thực sự. Trong đó, *lmp* và *rmp* là điểm mút trái và điểm mút phải của khoảng tính mờ.

Để ý rằng dựa trên cấu trúc thứ tự của $\boldsymbol{X}$, phần tử x nằm ở giữa hai tập $\{h_{-i}x: -q \le i \le -1\}$ và $\{h_jx: 1 \le j \le p\}$, hơn nữa ta có

$$\sum_{i\in[-q,-1]} |\mathfrak{I}(h_ix)| = fm(x).\ \sum_{i\in[-q,-1]} \mu(h_i) = \alpha.fm(x) = \alpha.|\mathfrak{I}(x)|$$

Điều này cho thấy điểm cuối chung của hai khoảng tính mờ $\mathfrak{I}(h_{-1}x)$ và $\mathfrak{I}(h_1x)$ chính là giá trị định lượng ngữ nghĩa $\upsilon(x)$ (xem [35]) của hạng từ x. Giá trị này chia đôi khoảng tính mờ $\mathfrak{I}(x)$ theo tỷ lệ $\alpha:\beta$ nếu $Sign(h_px) = 1$, hoặc tỷ lệ $\beta:\alpha$ nếu $Sign(h_px) = -1$ (xem (1) của Mệnh đề 1.3).

Theo Định nghĩa 1.7 và 1.8, có một mối liên hệ giữa ánh xạ định lượng ngữ nghĩa và khoảng tính mờ của của hạng từ trong một ĐSGT, được thể hiện bằng định lý sau.

Định lý 1.3. [35] Cho $\underline{\mathcal{AX}} = (\boldsymbol{X}, \boldsymbol{G}, \boldsymbol{H}, \sum, \boldsymbol{\Phi}, \le)$ là một ĐSGT tuyến tính đầy đủ và hàm υ được định nghĩa trong *Định nghĩa 1.7*. Khi đó υ là một ánh xạ định lượng ngữ nghĩa và tập các giá trị của υ đối với $\boldsymbol{H}(x)$, viết là $\upsilon(\boldsymbol{H}(x))$, trù mật trong đoạn $[\upsilon(\boldsymbol{\Phi}x), \upsilon(\sum x)]$, $\forall x \in \boldsymbol{X}$. Hơn nữa,

$$\upsilon(\boldsymbol{\Phi}x) = infimum\ \upsilon(\boldsymbol{H}(x)),\ \upsilon(\sum x) = supremum\ \upsilon(\boldsymbol{H}(x)) \text{ và}$$

$$fm(x) = \upsilon(\sum x) - \upsilon(\boldsymbol{\Phi}x),$$

và như vậy $fm(x) = d(\upsilon(\boldsymbol{H}(x)))$, trong đó $d(A)$ là đường kính của $A \subseteq [0,1]$. Kết quả, $\upsilon(\boldsymbol{H}(\boldsymbol{G}))$ trù mật trong đoạn [0,1].

Định lý này cũng khẳng định rằng ĐSGT $\underline{\mathcal{AX}}$ cùng với hàm định lượng ngữ nghĩa υ có thể ứng dụng trong mọi quá trình thực.

Từ những kết quả trên cho thấy giá trị định lượng ngữ nghĩa $\upsilon(x)$ của một hạng từ x cũng như khoảng tính mờ $\mathfrak{I}(x)$, $\forall x \in \boldsymbol{X}$, phụ thuộc đầy đủ vào các tham số mờ gia tử $fm(\boldsymbol{c}^-)$, $fm(\boldsymbol{c}^+)$, $\mu(h)$ $\forall h \in \boldsymbol{H}$.

1.2.3 Phương pháp lập luận xấp xỉ bằng nội suy theo tiếp cận đại số gia tử

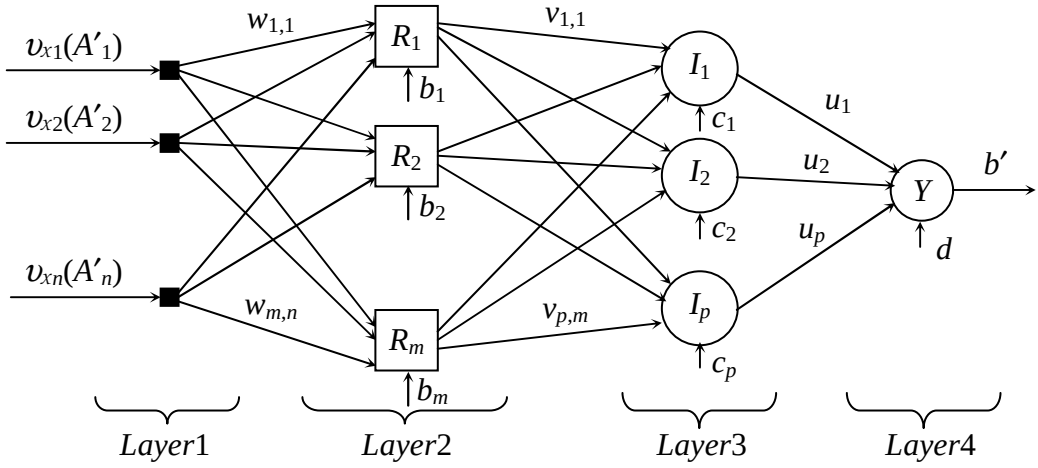
Trong Mục 1.1.3 chúng ta đã xem xét phương pháp lập luận xấp xỉ truyền thống theo mô hình hệ các luật mờ dạng (1.1). Với tiếp cận ĐSGT, các tác giả đã xây dựng phương pháp lập luận mới [7], [8], [35], [39]. Ở đây chúng ta xem xét một số vấn đề chính đối với bài toán lập luận xấp xỉ theo tiếp cận ĐSGT.

Mỗi luật mờ trong (1.1) sẽ xác định một điểm trong không gian tích Đề-các $\mathcal{D}_L = Dom(x_1) \times ... \times Dom(x_n) \times Dom(y)$, trong đó $Dom(x_j)$, $Dom(y)$ là các miền ngôn ngữ của các biến ngôn ngữ x_j, y (j=1,... , n) và chúng được xem như các ĐSGT. Như vậy, mô hình hệ luật mờ (1.1) định nghĩa một siêu mặt ngôn ngữ S_L trong không gian $\mathcal{D}_L$, cho nên giải bài toán lập luận xấp xỉ có nghĩa là tìm kết quả B' ứng với đầu vào (A'_1, ..., A'_n) bằng cách nội suy trên siêu mặt $S_{L,n+1}$.

Trong ĐSGT, chúng ta sử dụng các hàm định lượng ngữ nghĩa υ_{xi}, υ_y (*Định nghĩa 1.7*) để chuyển siêu mặt ngôn ngữ $S_{L,n+1}$ về siêu mặt thực $S_{R,n+1}$ trong không gian tham chiếu $\mathcal{U} = U_1 \times ... \times U_n \times V$ của các biến. Một số tác giả trong [8], [35] đề xuất sử dụng một phép kết nhập $\mathcal{Ag}$ để chuyển siêu mặt $S_{R,n+1}$ về dạng $S_{R,2}$ trong không gian thực hai chiều, sau đó áp dụng một phương pháp nội suy để tìm kết quả b' ứng với đầu vào $a' = \mathcal{Ag}(\upsilon_{x1}(A'_1), ..., \upsilon_{xn}(A'_n))$. Đối với một số bài toán cần kết quả lập luận là giá trị ngôn ngữ, trong [8] đã đề xuất hàm ngược của hàm định lượng ngữ nghĩa υ^{-1} để xác định giá trị ngôn ngữ của b'. Rõ ràng, phép kết nhập $\mathcal{Ag}$ có thể làm mất thông tin. Như vậy, kết quả lập luận ngoài phụ thuộc các tham số mờ gia tử của hàm định lượng ngữ nghĩa, còn phụ thuộc rất lớn đến phép kết nhập $\mathcal{Ag}$ cũng như phương pháp nội suy.

Một phương pháp lập luận thực hiện nội suy trực tiếp trên siêu mặt thực $S_{R,n+1}$ đã được đề xuất nhằm hạn chế sự mất mát thông tin của phép kết nhập. Trong [4], các tác giả sử dụng mạng nơron RBF để nội suy tìm kết quả của bài toán lập luận. Với thế mạnh của mạng nơron truyền tới đa lớp (FF) là công cụ xấp xỉ vạn năng [25], [55], cùng với thuật toán học lan truyền ngược sai số (BP). Trong bước đầu

nghiên cứu, luận án đã đề xuất mô hình mạng nơron FF để nội suy trên $S_{R,n+1}$ gồm 4 lớp (Hình 1.3), lớp vào (*Layer*1) có *n* nơron tương ứng với *n* đầu vào của hệ (1.1), lớp ẩn thứ nhất (*Layer*2) đóng vai trò của các luật mờ có *m* nơron, lớp ẩn thứ hai (*Layer*3) có *p* nơron và lớp ra (*Layer*4) một nơron.



Hình 1.3: Mô hình mạng nơron FF ứng dụng nội suy để lập luận

Các tham số liên kết giữa các lớp nơron và độ lệch của các nơron ký hiệu là $PAR_{net} = \{ w_{i,j}, b_i, v_{k,i}, c_k, u_k, d \mid j=1...n, i=1...m, k=1...p \}$, hàm kích hoạt tại các nơron được chọn ở dạng hàm *Gauss*. Đầu ra của mạng $b' = O(PAR_{net}, (\upsilon_{x1}(A'_1), ..., \upsilon_{xn}(A'_n)))$ là kết quả lập luận của bài toán đối với mỗi đầu vào $(\upsilon_{x1}(A'_1), ..., \upsilon_{xn}(A'_n))$. Thuật toán BP được áp dụng để điều chỉnh tham số mạng PAR_{net} sao cho kết quả lập luận đạt hiệu quả cao. Hàm đánh giá hiệu quả lập luận của phương pháp chính là ước lượng sai số giữa kết quả lập luận của mạng và siêu mặt $S_{R,n+1}$ như sau:

$$E_{net} = \frac{1}{n}\sqrt{\sum_{i=1}^{m}(b'_i - \upsilon_y(B_i))^2}\,, \tag{1.4}$$

trong đó $b'_i = O(PAR_{net}, (\upsilon_{x1}(A_{i,1}), ..., \upsilon_{xn}(A_{i,n})))$ là kết quả lập luận của mạng với đầu vào $(\upsilon_{x1}(A_{i,1}), ..., \upsilon_{xn}(A_{i,n}))$ tương ứng luật thứ *i* trong mô hình (1.1).

Số nơron tại lớp *Layer*3 được chọn phù hợp theo từng bài toán ứng dụng. Ngoài ra, chúng tôi đã thiết kế phương pháp tối ưu dựa trên giải thuật di truyền để tìm kiếm bộ tham số mờ gia tử tối ưu, sẽ trình bày chi tiết ở phần sau.

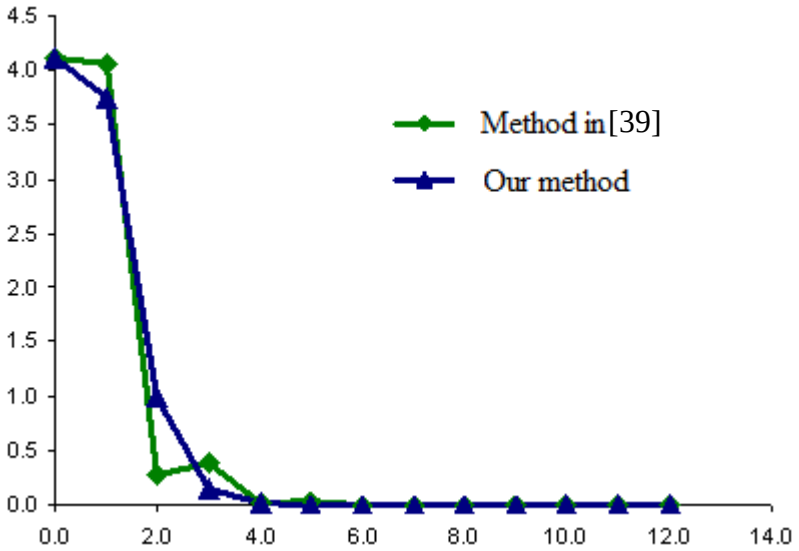
Áp dụng thử nghiệm phương pháp lập luận này vào bài toán điều khiển con lắc ngược đã được một số tác giả xem xét trong [4], [8]. Ở đây chúng ta tóm tắt bài toán ở dạng bảng các luật mờ (FAM) với giá trị ngôn ngữ trong ĐSGT và kết quả đạt được của phương pháp (Hình 1.4).

Bảng 1.1: Bảng các luật mờ dạng ngôn ngữ của bài toán điều khiển

X_1 \ X_2	***Large***	***W***	***Small***
Large	*More Large*	*Possible Large*	*W*
W	*Possible Large*	*W*	*Possible Small*
Small	*W*	*Possible Small*	*More Small*

Kết quả này, một lần nữa, thể hiện những nghiên cứu khởi đầu của luận án, đề xuất một phương pháp lập luận xấp xỉ theo tiếp cận ĐSGT dựa trên mạng nơron để nội suy trực tiếp và đạt kết quả khả quan.

Tuy nhiên, trong luận án này chúng tôi tập trung nghiên cứu và đề xuất phương pháp xây dựng hệ mờ dạng luật với ngữ nghĩa dựa trên ĐSGT. Một tiếp cận mới trong nghiên cứu và ứng dụng ĐSGT vào các bài toán khai phá dữ liệu. Phần tiếp theo sẽ giới thiệu bài toán phân lớp và một số mô hình đã và đang được nghiên cứu nhiều trong nước cũng như trên thế giới.



Hình 1.4: Kết quả sai số điều khiển của phương pháp và so sánh với [39]

1.3 Bài toán phân lớp trong khai phá dữ liệu

1.3.1 Giới thiệu bài toán phân lớp

Bài toán phân lớp (*classification*) là một trong những bài toán đặc trưng của lĩnh vực khai phá dữ liệu, được nhiều tác giả nghiên cứu và ứng dụng như Ishibuchi, Herrera, Abonyi, Chen, Khotanzad, Mansoori, Olson,... Trong đó, các phương pháp được biết đến như là cây quyết định, mạng nơron, phương pháp Bayes, SVM, boosting, random forest,... [54], [63]. Trong khi các phương pháp này tập trung giải quyết bài toán với mục tiêu đạt hiệu quả phân lớp cao nhất thì phương pháp dựa trên hệ mờ dạng luật (*fuzzy rule-based classification systems - FRBCS*), ngoài việc đạt hiệu quả phân lớp cao còn được nghiên cứu để đáp ứng cho người dùng một mô hình phân lớp dễ hiểu và trực quan. Người dùng có thể sử dụng các luật mờ trong mô hình như là các tri thức của mình để chủ động áp dụng trong thực tế. Phương pháp FRBCS được nhiều tác giả nghiên cứu sử dụng để giải bài toán (chẳng hạn trong [10], [17], [31], [40]-[46], [50], [60], [77]) và chúng ta gọi đây là bài toán phân lớp mờ.

Bài toán phân lớp mờ có thể được phát biểu như sau: cho một tập các mẫu dữ liệu $\boldsymbol{D}$ = { ($\boldsymbol{P}$; $\boldsymbol{C}$) }, trong đó $\boldsymbol{P}$ = { $p_i = (d_{i,1}, ..., d_{i,n})$ | i=1, ..., N } là tập dữ liệu, $\boldsymbol{C}$ = {C_1, ..., C_m} là tập các nhãn của các lớp, $p_i \in \mathcal{U}$ là dữ liệu thứ i với $\mathcal{U} = U_1 \times ... \times U_n$ là tích Đề-các của các miền của n thuộc tính x_1, ..., x_n tương ứng, m là số lớp và N là số mẫu dữ liệu, để ý rằng $\boldsymbol{P} \subset \mathcal{U}$. Mỗi dữ liệu $p_i \in \boldsymbol{P}$ thuộc một lớp $c_i \in \boldsymbol{C}$ tương ứng tạo thành từng cặp $(p_i, c_i) \in \boldsymbol{D}$. Giải bài toán bằng *FRBCS* chính là xây dựng một hệ các luật mờ, ký hiệu $\boldsymbol{S}$, để phân lớp đóng vai trò như một ánh xạ từ tập dữ liệu vào tập nhãn:

$$\boldsymbol{S} : \mathcal{U} \rightarrow \boldsymbol{C}. \tag{1.5}$$

Hệ các luật mờ này biểu diễn cho tri thức về bài toán, nó không chỉ phản ánh đúng với tập dữ liệu mẫu mà còn có khả năng dự đoán và cung cấp giúp cho người dùng phán đoán, ra quyết định. Do đó, hệ luật phải tường minh, dễ hiểu đối với người dùng.

Như vậy, hệ $\boldsymbol{S}$ phải đạt các mục tiêu như hiệu quả phân lớp cao, tức là sai số phân lớp cho các dữ liệu ít nhất có thể, số lượng các luật nhỏ cũng như số điều kiện tham gia trong vế trái mỗi luật ít. Mục tiêu về hiệu quả phân lớp nhằm đáp ứng tính đúng đắn của của hệ đối với tập dữ liệu mẫu được cho của bài toán, còn hai mục tiêu sau với mong muốn hệ luật phải tường minh, các luật mờ trong $\boldsymbol{S}$ phải đơn giản và dễ hiểu đối với người dùng. Nếu $f_p(\boldsymbol{S})$ là hàm đánh giá hiệu quả phân lớp, $f_n(\boldsymbol{S})$ là số luật và $f_a(\boldsymbol{S})$ là độ dài (hay số điều kiện tham gia) trung bình của vế trái trong hệ luật $\boldsymbol{S}$ thì mục tiêu là xây dựng hệ luật sao cho:

$$f_p(\boldsymbol{S}) \to max,\ f_n(\boldsymbol{S}) \text{ và } f_a(\boldsymbol{S}) \to min. \tag{1.6}$$

Ba mục tiêu trên không thể đạt được đồng thời. Khi số luật giảm đồng nghĩa với lượng tri thức về bài toán giảm thì nguy cơ phân lớp sai tăng lên, nhưng khi có quá nhiều luật cũng có thể gây ra sự nhiễu loạn thông tin trong quá trình phân lớp. Bên cạnh đó, số điều kiện của mỗi luật ảnh hưởng đến tính phổ quát hay cá thể của luật, cụ thể nếu số điều kiện ít sẽ làm tăng tính phổ quát và ngược lại số điều kiện tăng sẽ làm tăng tính cá thể của luật đó. Tính phổ quát sẽ làm tăng khả năng dự đoán của luật nhưng nguy cơ gây sai số lớn, trong khi tính cá thể giảm khả năng dự đoán nhưng lại tăng tính đúng đắn của luật. Các phương pháp giải quyết bài toán đều phải thỏa hiệp giữa các mục tiêu này để đạt được kết quả cuối cùng.

Các tác giả trong [50] sử dụng hệ luật mờ như dạng (1.1) cho bài toán phân lớp, khi đó kết quả lập luận đầu ra của hệ là một tập mờ B' đối với một mẫu dữ liệu, chúng ta cần giải mờ để xác định nhãn phân lớp cho mẫu dữ liệu tương ứng. Nhiều tác giả [10], [17], [23], [30]-[33], [40]-[46], [53], [59], [60], [74], [77] thì sử dụng các luật mờ có phần kết luận của mỗi luật là một giá trị hằng tương ứng với nhãn của một lớp, có dạng như sau:

$$\text{If } x_1 \text{ is } A_{q1} \text{ and ... and } x_n \text{ is } A_{qn} \quad \text{then} \quad \text{Class } C_q \quad \text{with} \quad CF_q, \tag{1.7}$$

trong đó $A_{q,j}$ là giá trị ngôn ngữ của các biến ngôn ngữ tương ứng với các thuộc tính, C_q là nhãn phân lớp và CF_q là trọng số của luật, q=1, ..., M với M là số luật,

j=1, ..., n. Thông thường, trọng số của luật là số thực trong khoảng đơn vị, $CF_q \in [0,1]$.

Đối với tập dữ liệu mẫu của bài toán phân lớp được cho dưới dạng số, tức là $\mathcal{U} \subset \mathcal{R}^n$, thì việc xây dựng một hệ luật mờ $\boldsymbol{S}$ thường gồm hai bước sau:

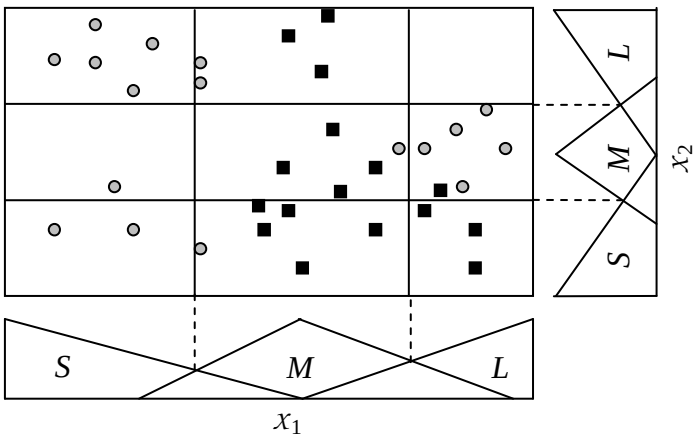
(B1) Phân hoạch mờ (*fuzzy partition*) trên miền của các thuộc tính bằng tập các giá trị ngôn ngữ của các biến ngôn ngữ - $Dom(x_i)$, mỗi giá trị ngôn ngữ được gán một hàm thuộc tương ứng.

(B2) Xác định các luật mờ từ các phân hoạch ở trên tạo thành hệ $\boldsymbol{S}$.

Bước phân hoạch mờ dựa trên các tập mờ tương ứng với các trị ngôn ngữ trên miền của các thuộc tính. Có hai phương pháp thường áp dụng đó là phân hoạch dưới dạng lưới (*grid-partition*) và phân hoạch theo sự phân bố dữ liệu (*scatter-partition*) (Hình 1.5 và 1.6). Để minh họa rõ hơn ta lấy ví dụ như sau.

Ví dụ 1.3. Cho bài toán phân lớp với tập mẫu có thuộc tính x_1, x_2 và hai lớp $\{C_1, C_2\}$ biểu thị bằng chấm tròn và vuông (Hình 1.5).

Theo phương pháp *grid-partition*, phân hoạch mờ trên miền của 2 thuộc tính thành các tập mờ dạng tam giác tương ứng với giá trị ngôn ngữ là {S(*small*), M(*medium*), L(*large*)} sẽ tạo thành một lưới phân hoạch mờ như Hình vẽ 1.5.



Hình 1.5: Lưới phân hoạch mờ trên miền của 2 thuộc tính

Lưới phân hoạch mờ này chia không gian tích Đề-các của các miền của thuộc tính tạo thành không gian các siêu hộp (*hyper-box*), ký hiệu $\mathcal{H}_S$, các luật mờ sẽ

được hình thành từ các tổ hợp của các giá trị ngôn ngữ trong không gian phân hoạch tương ứng với mỗi siêu hộp mà tại đó có hỗ trợ bởi các mẫu dữ liệu [42].

Tuy nhiên, các mẫu dữ liệu của các lớp khác nhau có thể thuộc cùng một siêu hộp, đây là một thách thức lớn đối với bất kỳ phương pháp xây dựng hệ luật mờ phân lớp nào. Trực quan từ ví dụ trong Hình 1.5, các hệ luật có thể được chọn:

- Hệ $\boldsymbol{S^1}$ gồm 7 luật mờ sau:

If x_1 is *Small* and x_2 is *Small* then Class C_1,
If x_1 is *Small* and x_2 is *Large* then Class C_1,
If x_1 is *Large* and x_2 is *Medium* then Class C_1,
If x_1 is *Large* and x_2 is *Small* then Class C_2,
If x_1 is *Medium* and x_2 is *Small* then Class C_2,
If x_1 is *Medium* and x_2 is *Medium* then Class C_2,
If x_1 is *Medium* and x_2 is *Large* then Class C_2.

- Hệ $\boldsymbol{S^2}$ gồm 4 luật mờ sau:

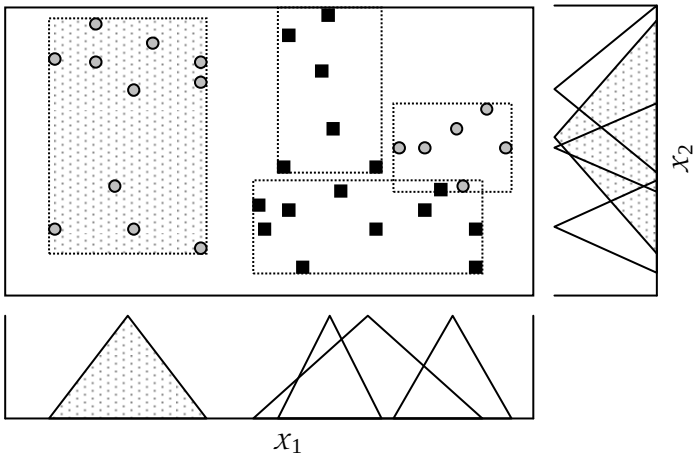
If x_1 is *Small* then Class C_1,
If x_1 is *Large* and x_2 is *Medium* then Class C_1,
If x_1 is *Medium* then Class C_2,
If x_1 is *Large* and x_2 is *Small* then Class C_2.

Giả sử rằng các luật mờ này có trọng số $CF = 1$.

Theo phương pháp *scatter-partition*, phân hoạch mờ dựa trên sự phân tích dữ liệu của bài toán. Thông thường được thực hiện bằng các phương pháp học máy (*machine learning*), chẳng hạn sử dụng giải thuật di truyền [14], [61] và được gắn với phương pháp điều chỉnh tham số mờ cho hệ mờ. Hình vẽ 1.6 minh họa phương pháp *scatter-partition*. Trong đó, trên miền của mỗi thuộc tính sẽ chọn các giá trị ngôn ngữ cùng với hàm thuộc tương ứng dựa trên sự phân tán của dữ liệu. Chẳng hạn hình chữ nhật tô màu chứa các dữ liệu với phân hoạch bởi các hàm thuộc dạng tam giác có màu tương ứng trên x_1, x_2.

Rõ ràng phương pháp giải bài toán phân lớp mờ phụ thuộc vào các yếu tố như chọn tập mờ cho các giá trị ngôn ngữ để phân hoạch trên miền của các thuộc tính

cũng như số lượng các giá trị ngôn ngữ, phương pháp lựa chọn, xác định các luật mờ từ không gian các siêu hộp $\mathcal{H}_S$ để đạt các mục tiêu trong (1.6). Trong phần tiếp theo sẽ trình bày chi tiết hơn phương pháp xây dựng hệ luật mờ cho bài toán phân lớp.



Hình 1.6: Phương pháp phân hoạch mờ *scatter-partition*

1.3.2 Mô hình hệ mờ dạng luật giải bài toán phân lớp

Mô hình hệ luật mờ dưới dạng (1.7) được nhiều tác giả nghiên cứu và áp dụng giải bài toán phân lớp trong các kết quả [10], [17], [23], [30]-[29], [40]-[47], [53], [59], [60], [74], [77]. Luật mờ dạng (1.7) có thể viết gọn lại như sau:

$$A_q \Rightarrow C_q \text{ with } CF_q, \tag{1.8}$$

trong đó $A_q = (A_{q,1}, ..., A_{q,n})$.

Tương tự trong khai phá luật kết hợp [63], [27], luật mờ (1.8) được đánh giá qua độ tin cậy $c(A_q \Rightarrow C_q)$ và độ hỗ trợ $s(A_q \Rightarrow C_q)$ bằng công thức (1.9) và (1.10):

$$c(A_q \Rightarrow C_q) = \frac{\sum_{p_i \in Class\, C_q} \mu_{Aq}(p_i)}{\sum_{i=1}^{N} \mu_{Aq}(p_i)}, \tag{1.9}$$

$$s(A_q \Rightarrow C_q) = \frac{\sum_{p_i \in Class\, C_q} \mu_{Aq}(p_i)}{N}. \tag{1.10}$$

Có thể sử dụng một phép toán *t-norm* bất kỳ để tính mức đốt cháy của mẫu dữ liệu p_i đối với điều kiện A_q của luật mờ, thông thường các tác giả áp dụng *t-norm* dạng tích khi đó:

$$\mu_{Aq}(p_i) = \mu_{Aq,1}(d_{i,1}) \,.\, \mu_{Aq,2}(d_{i,2}) \,.\, \ldots \,.\, \mu_{Aq,n}(d_{i,n}). \tag{1.11}$$

Để tiện về sau ký hiệu c_q và s_q là độ tin cậy và hỗ trợ của luật dạng (1.8).

Để đánh giá trọng số (*CF*) của luật dạng (1.8), nhóm tác giả H. Ishibuchi [43], [44] dựa trên độ tin cậy của luật đã đề xuất các phương pháp đánh giá trọng số luật như sau:

$$CF^1(A_q \Rightarrow C_q) = c_q, \tag{1.12}$$

$$CF^2(A_q \Rightarrow C_q) = c_q - c_{q,Ave}, \tag{1.13}$$

$$CF^3(A_q \Rightarrow C_q) = c_q - c_{q,2nd}, \tag{1.14}$$

$$CF^4(A_q \Rightarrow C_q) = c_q - c_{q,Sum}, \tag{1.15}$$

trong đó $c_{q,Ave}$ là độ tin cậy trung bình của các luật có cùng điều kiện A_q nhưng kết luận khác C_q:

$$c_{q,Ave} = \frac{1}{m-1} \sum_{\substack{h=1 \\ C_h \neq C_q}}^{m} c(A_q \Rightarrow C_h), \tag{1.16}$$

$c_{q,2nd}$ là độ tin cậy lớn nhất của các luật có cùng điều kiện A_q nhưng kết luận là lớp khác với C_q:

$$c_{q,2nd} = \max\{c(A_q \Rightarrow C_h) \mid h = 1, \ldots, m;\ C_h \neq C_q\}, \tag{1.17}$$

$c_{q,Sum}$ là tổng các độ tin cậy của các luật có cùng điều kiện A_q nhưng kết luận là lớp khác với C_q:

$$c_{q,Sum} = \sum_{\substack{h=1 \\ C_h \neq C_q}}^{m} c(A_q \Rightarrow C_h). \tag{1.18}$$

Để ý rằng nếu số lớp m=2 thì các công thức (1.16), (1.17) và (1.18) sẽ đồng nhất với nhau, $c_{q,Ave} = c_{q,2nd} = c_{q,Sum}$.

Các tác giả cũng đã phân tích và minh họa bằng một số kết quả thực nghiệm rằng với trọng số luật đánh giá theo CF^3 (công thức 1.14 và 1.17) cho kết quả tốt hơn so với CF^1 và CF^2. Một số trường hợp thì trọng số luật theo CF^4 lại cho kết quả tốt nhất. Tuy nhiên trọng số CF^4 tính theo công thức (1.15) và (1.18) có nguy cơ rất lớn cho giá trị âm, vì $c_{q,Sum}$ có thể lớn hơn c_q. Do đó các tác giả [42]-[47], [30] hầu hết áp dụng trọng số CF^3 cho các bài toán ứng dụng. Trong luận án cũng có sử dụng luật với trọng số $CF^0 = 1$ để minh họa.

Với hệ luật mờ $\boldsymbol{S}$ dạng (1.8), có thể áp dụng hai phương pháp lập luận [43], [44] để phân lớp cho một dữ liệu $p' = (d'_1, ..., d'_n) \in \mathcal{U}$. Thứ nhất là phương pháp chọn luật có mức đốt cháy lớn nhất đối với dữ liệu đưa vào và phân lớp tương ứng với kết luận của luật đó (*single winner rule - SWR*):

$$Classify_{SWR}(p') = \arg\max_{C_h}\{\mu_{A_h}(p').CF^{\,w} \mid A_h \Rightarrow C_h \in \boldsymbol{S}\}, \tag{1.19}$$

trong đó w là chỉ số tương ứng trọng số luật được chọn, $w \in \{1,2,3,4\}$, hoặc có thể áp dụng với trọng số đồng nhất bằng 1 cho mọi luật, ký hiệu $CF^0 = 1$.

Phương pháp lập luận thứ hai, tính tổng mức đốt cháy của các luật có cùng kết luận đối với dữ liệu p' và chọn lớp nào có tổng này lớn nhất (*weighted vote - WV*):

$$Classify_{WV}(p') = \arg\max_{C_h}\{V_h(p') \mid h = 1,...,m\}, \tag{1.20}$$

trong đó $V_h(p')$ là tổng mức đốt cháy (*vote*) của các luật đối với mẫu dữ liệu p',

$$V_h(p') = \sum_{\substack{A_q \Rightarrow C_q \in S \\ C_q = C_h}} \mu_{Aq}(p').CF^{\,w}\,. \tag{1.21}$$

Ở đây, các tác giả đã phân tích và minh họa rằng hiệu quả phân lớp của phương pháp lập luận *weighted vote* cao hơn *single winner rule* trong một số ví dụ mẫu. Tuy nhiên, phương pháp lập luận *single winner rule* luôn đem lại tính trực quan cao hơn, hơn nữa phương pháp này cho phép giảm (rút gọn) số luật trong hệ luật một cách dễ dàng [44].

Tiếp theo chúng ta xem xét phương pháp sinh luật dựa trên lưới phân hoạch mờ của miền các thuộc tính [23], [42]-[47], [53]. Trong không gian các siêu hộp $\mathcal{H}_S$, mỗi $(A_{q,1}, ..., A_{q,n}) \in \mathcal{H}_S$ sẽ dùng để xây dựng một luật mờ bằng cách đặt điều kiện của luật tương ứng với siêu hộp đó $A_q = (A_{q,1}, ..., A_{q,n})$, phần kết luận được chọn là nhãn phân lớp sao cho luật đạt độ tin cậy lớn nhất:

$$C_q = \arg\max_{C_h}\{c(A_q \Rightarrow C_h) \mid s(A_q \Rightarrow C_h) > 0, h = 1,..., m\}. \quad (1.22)$$

Trường hợp có nhiều lớp cùng thỏa mãn (1.22), trong [44] đề nghị loại bỏ luật có điều kiện A_q này. Tuy nhiên, chúng ta có thể chọn một trong số chúng theo cách ngẫu nhiên nhằm hạn chế sự mất mát thông tin.

Phương pháp sinh luật này sẽ đảm bảo các công thức đánh giá trọng số của luật là (1.12) và (1.14) luôn dương. Tuy nhiên khi áp dụng trọng số luật CF^4 (công thức (1.15)) có thể cho giá trị âm, khi đó luật tương ứng sẽ không được xem xét và bị loại bỏ.

Như vậy, theo phương pháp phân hoạch mờ trên miền của các thuộc tính đã trình bày trong phần trước, nếu số các giá trị ngôn ngữ để phân hoạch trên mỗi thuộc tính là K_j, thì kích thước không gian các siêu hộp trong lưới phân hoạch mờ là $|\mathcal{H}_S| = \prod_{j=1,...,n} K_j$. Trường hợp chọn số các giá trị ngôn ngữ giống nhau và bằng K cho mọi thuộc tính, $|\mathcal{H}_S| = K^n$.

Mặt khác, với các mục tiêu trong (1.6), các tác giả trong [43] sử dụng thêm một giá trị "*Don't Care*" (*DC*) trong phân hoạch các giá trị ngôn ngữ cho các thuộc tính để biểu thị thuộc tính đó không xét đến trong luật mờ chứa nó, tức là hàm thuộc của giá trị này đồng nhất bằng 1 ($\mu_{DC}(d) = 1, \forall d$). Như vậy kích thước không gian các siêu hộp là $|\mathcal{H}_S| = (K_1+1) \times (K_2+1) \times ... \times (K_n+1)$, hoặc $|\mathcal{H}_S| = (K+1)^n$ khi $K_j = K, \forall j$. Một không gian rất lớn đối với việc lựa chọn, xác định các luật mờ vì số thuộc tính của bài toán phân lớp thường không nhỏ, cho dù số phân hoạch K có thể nhỏ. Chẳng hạn với bài toán có $n = 4$, nếu chọn $K = 3$ thì $|\mathcal{H}_S| = (3+1)^4 = 256$,

thậm chí bài toán có n = 13, kích thước không gian các siêu hộp để sinh luật sẽ là $|\mathcal{H}_S|$ = (3+1)^{13} = 67108864, một con số rất lớn.

Ký hiệu $\boldsymbol{S_0}$ là tập tất cả các luật mờ được sinh ra từ không gian $\mathcal{H}_S$, kích thước tập $\boldsymbol{S_0}$ có khả năng rất lớn, có thể $|\boldsymbol{S_0}| = |\mathcal{H}_S|$ trong trường hợp cực đoan. Do đó các tác giả trong [43], [44] đề xuất phương pháp sàng nhằm rút gọn hệ luật. Mỗi luật trong $\boldsymbol{S_0}$ sẽ được đánh giá tiêu chuẩn lựa chọn (hay tiêu chuẩn sàng), ký hiệu là *SR*, như sau:

$$SR^1(A_q \Rightarrow C_q) = c_q, \tag{1.23}$$

$$SR^2(A_q \Rightarrow C_q) = s_q, \tag{1.24}$$

$$SR^3(A_q \Rightarrow C_q) = c_q \,.\, s_q. \tag{1.25}$$

Phương pháp sàng dựa trên một trong các tiêu chuẩn trên được thực hiện như sau: Sắp xếp các luật mờ trong tập $\boldsymbol{S_0}$ theo nhóm của các lớp trong phần kết luận với tiêu chuẩn (SR^z) được chọn giảm dần, $z \in \{1,2,3\}$. Giả sử cần chọn ra mỗi lớp M_c luật, khi đó lấy từ trên xuống đúng M_c luật trong mỗi nhóm được sắp.

Các tác giả cũng đã minh họa bằng một số kết quả thực nghiệm rằng với tiêu chuẩn SR^3 (công thức 1.25) sẽ cho kết quả hệ luật với hiệu quả phân lớp tốt hơn so với hai tiêu chuẩn còn lại [44].

Một phương pháp khác được các tác giả áp dụng là thiết kế các thuật toán tìm kiếm hệ luật tối ưu dựa trên giải thuật di truyền (*GA*) [28], [43]. Trong đó các luật mờ được mã hóa bằng các cá thể trong *GA* bởi một trong hai phương pháp là Michigan hoặc Pittsburgh. Phương pháp Michigan mã hóa mỗi luật mờ thành một cá thể trong khi phương pháp Pittsburgh mã hóa tập các luật mờ thành một cá thể. Như vậy, lời giải cho bài toán tìm kiếm hệ luật không thể xác định trực tiếp từ một cá thể có độ phù hợp tốt nhất trong phương pháp Michigan, nhưng với Pittsburgh chúng ta có thể xác định lời giải dựa trên cá thể có độ phù tốt nhất trong quần thể. Hầu hết các tác giả sử dụng phương pháp Pittsburgh, tuy nhiên trong [46] đã kết hợp cả phương pháp Michigan vào giải thuật di truyền. Hơn nữa, nhóm nghiên cứu

của H. Ishibuchi đã tiếp cận phương pháp tìm kiếm tối ưu *Pareto* dựa trên giải thuật di truyền, gọi là giải thuật NSGA-II được đề xuất bởi K. Deb [21].

Ngoài ra, một số tác giả đã đề xuất các phương pháp điều chỉnh trọng số của luật (*CF*) để tăng hiệu quả phân lớp cho hệ luật. S.M. Fakhrahmad trong [23] xuất phát với trọng số *CF*=1, áp dụng phương pháp giảm gradient theo hàm đánh giá sai số phân lớp. E. G. Mansoori trong [60] sử dụng một hàm thay đổi giá trị trọng số của luật theo các trường hợp khác nhau đối với từng mẫu dữ liệu đưa vào.

Tất cả các phương pháp trên đều nỗ lực tìm kiếm một hệ luật mờ phân lớp nhằm đạt các mục tiêu trong (1.6). Một yếu tố rất quan trọng và ảnh hưởng lớn đến kết quả của các phương pháp là việc chọn tập các giá trị ngôn ngữ cùng với hàm thuộc tương ứng để phân hoạch mờ trên miền của các thuộc tính. Trực quan chúng ta thấy rằng với bất kỳ một bài toán phân lớp, phân hoạch mờ trên miền của các thuộc tính không giống nhau. Điều này do đặc trưng và sự phân bố dữ liệu của bài toán. Vì vậy, áp dụng cùng một kiểu phân hoạch mờ cho các thuộc tính của bài toán, hay thậm chí của các bài toán khác nhau như trong [23], [30], [42]-[46], [60], [74], [77] sẽ làm giảm hiệu quả của phương pháp. Một số tác giả trong [10], [17], [40], [59] đã sử dụng các giải thuật tìm kiếm tối ưu phân hoạch mờ chủ yếu dựa trên *GA*. Tuy nhiên phương pháp tìm kiếm phân hoạch tối ưu sẽ làm cho các tập mờ tương ứng với các giá trị ngôn ngữ trở nên không thực tế (xem Hình vẽ 1.6 ở trên), có thể hai giá trị ngôn ngữ khác nhau nhưng hàm thuộc gần như đồng nhất. Do vậy chúng ta phải thỏa hiệp giữa hiệu quả phân lớp và tính đơn giản dễ hiểu của hệ luật trong các phương pháp giải bài toán phân lớp mờ.

1.4 Kết luận Chương 1

Trong chương này 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. Trong lý thuyết tập mờ và lôgíc mờ, mô hình hệ mờ dạng luật (*fuzzy rule-based systems*) và các phương pháp lập luận xấp xỉ trên mô hình này đã và đang được nhiều nhà nghiên

cứu phát triển và ứng dụng trong các bài toán thực tế. Luận án đã tóm tắt phương pháp lập luận dựa trên quy tắc *modus ponens* tổng quát hóa và qua đó thấy rằng có rất nhiều cách chọn các phép toán lôgíc mờ áp dụng vào lập luận. Mỗi cách chọn sẽ cho một kết quả lập luận khác nhau và không thể khẳng định chắc chắn chọn như thế nào thì sẽ có một phương pháp lập luận tốt. Điều này phụ thuộc vào từng tình huống ứng dụng cụ thể và được kiểm chứng qua kết quả thực nghiệm.

Tiếp theo, trong Mục 1.2 trình bày các khái niệm cơ bản về đại số gia tử, vấn đề định lượng ngữ nghĩa và khái niệm về khoảng tính mờ của các giá trị ngôn ngữ. Trên cơ sở đó một số tác giả đã đề xuất các phương pháp lập luận xấp xỉ theo tiếp cận ĐSGT và ứng dụng thành công. Bước đầu nghiên cứu trong luận án đã giới thiệu một phương pháp lập luận bằng nội suy trực tiếp trên siêu mặt $S_{R,n+1}$ sử dụng mạng nơron truyền tới đa lớp và ứng dụng cho kết quả tốt. Điều này cho thấy tiềm năng và thế mạnh của các phương pháp lập luận theo tiếp cận ĐSGT.

Trong Mục 1.3 trình bày về bài toán phân lớp và các phương pháp tiếp cận giải bài toán dựa trên mô hình hệ các luật mờ của nhiều tác giả. Tuy nhiên các phương pháp này gặp trở ngại là số luật sinh ra có thể rất lớn do vậy đòi hỏi một khối lượng tính toán khổng lồ, hoặc các hàm thuộc của các giá trị ngôn ngữ có thể gần như đồng nhất với nhau khi sử dụng các biện pháp điều chỉnh tham số.

Từ những nghiên cứu về ĐSGT và các phương pháp lập luận, trong luận án này tập trung nghiên cứu và phát triển phương pháp xây dựng hệ mờ dạng luật với ngữ nghĩa dựa trên ĐSGT và ứng dụng vào bài toán phân lớp trong khai phá dữ liệu, một bài toán đang được nghiên cứu khá mạnh mẽ. Trong chương tiếp theo sẽ đi sâu nghiên cứu, đề xuất phương pháp xây dựng hệ mờ dạng luật với ngữ nghĩa dựa trên ĐSGT.

CHƯƠNG 2
PHƯƠNG PHÁP SINH LUẬT MỜ VỚI NGỮ NGHĨA CÁC TỪ NGÔN NGỮ DỰA TRÊN ĐSGT

Chương 1 đã trình bày phương pháp xây dựng hệ mờ dạng luật dựa trên phân hoạch bởi các tập mờ. Tuy nhiên, tồn tại trong đó một sự tách biệt giữa cú pháp của các giá trị ngôn ngữ và ngữ nghĩa của chúng biểu diễn bằng tập mờ. Điều này dẫn đến các quá trình sinh hệ luật mờ và tối ưu tham số tập mờ làm biến dạng ngữ nghĩa của các giá trị ngôn ngữ, hay các tập mờ khó phản ánh bản chất ngữ nghĩa của các giá trị ngôn ngữ. Chẳng hạn các tác giả trong [10], [50] sử dụng phương pháp điều chỉnh tham số tập mờ nhưng với khoảng cách này thì các tập mờ kết quả (Hình 3.1) rất khó nhận biết ngữ nghĩa cũng như giá trị ngôn ngữ mà chúng phản ánh. Một tập mờ có thể gần như nằm bên trong một tập mờ khác. Các luật mờ kết quả không trực quan, khó hiểu đối với người dùng trong khi người dùng cần đến các tri thức này để phán đoán, ra quyết định trong ứng dụng thực tế.

Mặt khác, một số tác giả trong [42]-[47], [23],[53],[60] chọn cố định tập các giá trị ngôn ngữ cùng với tập mờ biểu diễn ngữ nghĩa của chúng sẽ rất khó phù hợp đối với ngữ cảnh của các bài toán khác nhau. Tuy nhiên, sử dụng ngữ nghĩa để xác định tập các giá trị ngôn ngữ tương ứng với các tập mờ này cũng rất khó khăn. Chẳng hạn trong [43] sử dụng tập mờ dạng $\mu_{S2}(x) = 1\text{-}x$ trên miền [0,1], trực quan rất khó xác định giá trị ngôn ngữ với ngữ nghĩa của tập mờ này.

Đại số gia tử với những đặc trưng của mình giúp giữ mối liên kết về cú pháp của các giá trị ngôn ngữ và ngữ nghĩa của chúng biểu diễn bằng các tập mờ, tạo ra những ràng buộc nhất định để ngữ nghĩa của các giá trị ngôn ngữ không bị biến dạng trong các quá trình xử lý. Như vậy, việc sử dụng ĐSGT trong quá trình xây dựng hệ mờ dạng luật sẽ giúp khắc phục được những hạn chế đã đề cập ở trên. Kết quả hệ luật mờ sẽ phản ánh đúng bản chất của ngôn ngữ, qua đó người dùng có thể dễ dàng trong việc sử dụng các tri thức dạng luật này.

Trước hết, trong phần tiếp theo sẽ thiết kế phương pháp xây dựng hệ luật mờ từ tập dữ liệu mẫu của bài toán với ngữ nghĩa dựa trên ĐSGT.

2.1 Lược đồ xây dựng hệ luật mờ dựa trên ĐSGT

Trước hết chúng ta nhắc lại bài toán xây dựng hệ luật mờ phân lớp (trong Mục 1.3.1). Bài toán cho một tập các mẫu dữ liệu $\boldsymbol{D} = \{ (\boldsymbol{P}; \boldsymbol{C}) \}$, trong đó $\boldsymbol{P} = \{ p_i = (d_{i,1}, ..., d_{i,n}) \mid i=1, ..., N \}$ là tập dữ liệu, $\boldsymbol{C} = \{C_1, ..., C_m\}$ là tập các nhãn của các lớp, $p_i \in \mathcal{U}$ là dữ liệu thứ i, $\mathcal{U} = U_1 \times ... \times U_n$ là tích Đề-các của các miền của n thuộc tính $x_1, ..., x_n$ tương ứng, m là số lớp và N là số mẫu dữ liệu, để ý rằng $\boldsymbol{P} \subset \mathcal{U}$. Mỗi dữ liệu $p_i \in \boldsymbol{P}$ được gán nhãn phân lớp $c_i \in \boldsymbol{C}$ tương ứng tạo thành từng cặp $(p_i, c_i) \in \boldsymbol{D}$. Thông thường miền của các thuộc tính là miền thực, tức là $\mathcal{U} \subset \mathcal{R}^n$. Lược đồ xây dựng hệ luật mờ phân lớp cho tập dữ liệu mẫu $\boldsymbol{D}$ thường gồm hai hai bước chính như sau:

(*Step*1) Phân hoạch mờ (*fuzzy partition*) trên miền của các thuộc tính dựa trên tập các giá trị ngôn ngữ của các biến ngôn ngữ - $Dom(x_i)$, mỗi giá trị ngôn ngữ được thiết kế một hàm định lượng ngữ nghĩa tương ứng.

(*Step*2) Xác định các luật mờ từ các phân hoạch ở trên tạo thành hệ luật mờ $\boldsymbol{S}$ dạng (1.7).

Dựa trên ĐSGT, trong bước 1 chúng ta có hai phương pháp phân hoạch mờ. Thứ nhất, chúng ta áp dụng phương pháp lưới phân hoạch mờ dựa trên hệ các khoảng tính mờ $\boldsymbol{I_k}$ của một tập hạng từ mức k ($\boldsymbol{X_k}$). Ký hiệu ĐSGT cho miền ngôn ngữ của mỗi thuộc tính x_j là $\mathfrak{AX}_j$. Theo Mệnh đề 1.3 với khoảng tính mờ xét ở dạng nửa đóng, hệ khoảng tính mờ $\boldsymbol{I_k}$ là một phân hoạch của [0,1]. Bằng cách chọn mức phân hoạch k_j thích hợp đối với mỗi thuộc tính, khi đó miền của mỗi thuộc tính được phân hoạch bởi $\boldsymbol{I}_{kj}$ và tương ứng là tập giá trị ngôn ngữ $\boldsymbol{X}_{kj}$. Mặt khác miền của các thuộc tính thường là miền thực, $U_j = [a_j, b_j] \subset \mathcal{R}$, chúng ta chuẩn hóa về miền đơn vị [0,1] bằng các hàm chuyển như sau:

$$f_j(v) = \frac{v - a_j}{b_j - a_j},\ \forall v \in U_j,\ j = 1, ..., n. \quad (2.1)$$

Phương pháp phân hoạch dựa trên hệ các khoảng tính mờ như trên sẽ tạo nên một không gian gồm các siêu hộp $B_i \in \mathcal{H}_S = \boldsymbol{I}_{k1} \times ... \times \boldsymbol{I}_{kn}$ tương ứng với không gian tích Đề-các của các giá trị ngôn ngữ $\mathcal{L}_S = \boldsymbol{X}_{k1} \times ... \times \boldsymbol{X}_{kn}$. Hệ các luật mờ với điều kiện là các giá trị ngôn ngữ sẽ được xây dựng dựa trên không gian $\mathcal{H}_S$. Mỗi siêu hộp $B_q \in \mathcal{H}_S$ ứng với một tập giá trị ngôn ngữ $\boldsymbol{A}_q = (A_{q1}, ..., A_{qn}) \in \mathcal{L}_S$ xác định điều kiện vế trái của tuyển một luật mờ. Chúng ta sinh luật mờ này nếu B_q có chứa ít nhất một mẫu dữ liệu trong $\boldsymbol{D}$. Khi đó, phần kết luận vế phải của luật là nhãn phân lớp được chọn sao cho luật sinh ra đạt độ tin cậy cao nhất. Ta có luật mờ sinh theo dạng sau, được gọi là luật cơ sở:

$$\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_q, \quad (2.2)$$

trong đó

$$\boldsymbol{C}_q = \arg\max_{C_h} \{\ c(\boldsymbol{A}_q \Rightarrow C_h) \mid h = 1,...,m\ \} \quad (2.3)$$

với c là độ tin cậy được tính theo công thức (1.9), (1.10). Trường hợp có nhiều lớp cùng thỏa mãn (2.3) thì chọn ngẫu nhiên một trong chúng.

Theo tính chất phân hoạch của hệ khoảng tính mờ mức k trong ĐSGT - $\boldsymbol{I}_k$, mỗi mẫu dữ liệu $p_i \in \boldsymbol{P}$ xác định duy nhất một siêu hộp B_i. Chúng ta chỉ xem xét sinh hệ luật từ những siêu hộp có chứa mẫu dữ liệu, do đó số luật tối đa được sinh là N trong trường hợp cực đoan, tức là bất kỳ hai mẫu dữ liệu đều không cùng thuộc một siêu hộp trong $\mathcal{H}_S$. Lược đồ sinh luật này giảm thiểu tính toán và xem xét đến các khả năng sinh luật từ không gian các phân hoạch $\mathcal{H}_S$, nhỏ hơn nhiều so với phương pháp của Ishibuchi có số khả năng sinh các luật là $|\mathcal{H}_S|$.

Hạn chế phương pháp phân hoạch dựa trên hệ khoảng tính mờ là chỉ sinh luật với tập các hạng từ có độ dài đúng k_j trong ĐSGT của mỗi thuộc tính x_j, việc bỏ qua các hạng từ độ dài nhỏ hơn k_j về trực quan không hợp lý và có thể làm giảm hiệu

năng của hệ mờ sinh ra. Một cách rất tự nhiên nhằm khắc phục hạn chế trên là áp dụng phương pháp lưới phân hoạch mờ bởi tập các hạng từ có độ dài không quá k ($\boldsymbol{X}_{(k)}$) thay cho tập hạng từ độ dài đúng k ($\boldsymbol{X}_k$). Như vậy chúng ta cần xây dựng một hệ phân hoạch (kiểu như hệ khoảng tính mờ) của tập các hạng từ $\boldsymbol{X}_{(k)}$ và đây là phương pháp phân hoạch thứ 2 trong bước 1 của lược đồ trên. Điều này sẽ được trình bày chi tiết trong Mục 2.3.

Đối với các bài toán có số các thuộc tính lớn, để đảm bảo tính đơn giản và dễ hiểu đối với hệ luật mờ sinh ra và hơn nữa thực tế các thuộc tính có những vai trò khác nhau quyết định đến việc phân lớp, do đó chúng ta mong muốn các luật sinh ra chỉ chứa điều kiện của một số ít các thuộc tính có vai trò quyết định đến phân lớp. Theo tiếp cận của H. Ishibuchi và các cộng sự [43], chúng ta sử dụng thêm một giá trị ngôn ngữ "*Don't Care*" (*DC*) trong phân hoạch để chỉ sự không quan tâm đối với các thuộc tính được loại bỏ trong vế trái của mỗi luật, hàm định lượng ngữ nghĩa của giá trị ngôn ngữ này đồng nhất bằng 1 trên miền của thuộc tính ($\mu_{DC}(v) = 1, \forall v$). Khi đó mỗi luật cơ sở dạng $\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_q$ sẽ được sinh các luật thứ cấp dạng $\boldsymbol{A}_q(i) \Rightarrow \boldsymbol{C}_q(i)$, với $\boldsymbol{A}_q(i)$ được chọn từ các điều kiện trong $\boldsymbol{A}_q$ theo tổ hợp số các thuộc tính ($\boldsymbol{A}_q(i) \subseteq \boldsymbol{A}_q$, $|\boldsymbol{A}_q(i)| \leq L$), $\boldsymbol{C}_q(i)$ được xác định theo công thức (2.3). L là độ dài (số các giá trị ngôn ngữ khác *DC* trong điều kiện ở vế trái) tối đa của luật cần sinh và được cho trước, trường hợp số thuộc tính n nhỏ có thể chọn $L = n$.

Đây có thể xem như một phương pháp rút gọn vế trái của các luật mờ nhằm thực hiện mục tiêu giảm thiểu số điều kiện vế trái của các luật trong (1.6).

Lược đồ xây dựng hệ luật mờ dựa trên ĐSGT phụ thuộc các tham số mờ gia tử, số lượng các tham số này ít hơn nhiều so với các phương pháp dựa trên tập mờ trong [29], [50]. Số các tham số mờ cho mỗi thuộc tính gồm $fm(\boldsymbol{c}^-)$ và $\mu(h)$ $\forall h \in \boldsymbol{H}$, nếu sử dụng ĐSGT với 4 gia tử $\boldsymbol{H}$ = { *Little*, *Possible*, *More*, *Very* } thì tổng các tham số của bài toán là $4n+n = 5n$, để ý rằng $fm(\boldsymbol{c}^+) = 1\text{-}\ fm(\boldsymbol{c}^-)$, n là số thuộc tính. Đặc biệt, nếu áp dụng ĐSGT với 2 gia tử thì số lượng tham số chỉ là $2n$, vì $\mu(V) = 1\text{-}\mu(L)$. Trong khi đó, các phương pháp trong [29], [50] với hàm thuộc dạng tam giác có tổng các tham số mờ là $(3K)n$, trong đó K là số các tập mờ trong phân hoạch

cho mỗi thuộc tính. Phương pháp trong [74] có tổng các tham số mờ là $(K+1)n$. Điều này cho thấy mô hình dựa trên ĐSGT sẽ giảm độ phức tạp trong các quá trình tìm kiếm tối ưu tham số mờ.

2.2 Phương pháp sinh luật mờ dựa trên hệ khoảng tính mờ

2.2.1 Hệ khoảng tính mờ và quan hệ ngữ nghĩa của các hạng từ

Chương 1 chúng ta ký hiệu $\boldsymbol{X}_k$ là tập các hạng từ độ dài k trong ĐSGT, $\boldsymbol{I}_k = \{\mathfrak{I}(x) : x \in \boldsymbol{X}_k\}$ là tập các khoảng tính mờ của các hạng từ trong $\boldsymbol{X}_k$ và là một phân hoạch của [0,1] (theo (2) của Mệnh đề 1.3). Ta gọi $\boldsymbol{I_k}$ là hệ phân hoạch khoảng tính mờ mức k (hay độ sâu k). Nếu đặt $x_{k,0}$ là hạng từ bé nhất trong tập $\boldsymbol{X}_k$, thì $\upsilon(\boldsymbol{\Phi}x_{k,0}) = 0$. Theo Định lý 1.3 và Định nghĩa 1.8, chúng ta có $\mathfrak{I}(x_{k,0}) = [\upsilon(\boldsymbol{\Phi}x_{k,0}), \upsilon(\boldsymbol{\sum}x_{k,0})]$ và $\mathfrak{I}(x) = (\upsilon(\boldsymbol{\Phi}x), \upsilon(\boldsymbol{\sum}x)]$ cho $\forall x \in \boldsymbol{X}_k$, $x \neq x_{k,0}$, trong đó quy ước khoảng tính mờ luôn đóng ở điểm mút phải. Hơn nữa, nếu ký hiệu λ_k là độ dài lớn nhất của các khoảng tính mờ trong $\boldsymbol{I}_k$ và η là độ đo tính mờ lớn nhất của các gia tử trong $\boldsymbol{H}$, thì theo (4) của Mệnh đề 1.1 ta có $\lambda_{k+1} \leq \eta\lambda_k \leq \eta^k\lambda_1$. Do $\eta < 1$ nên ta luôn tìm được khoảng tính mờ của x cho dù khoảng cần tìm bé đến mức nào.

Điều này cho phép xây dựng các thuật toán xác định các khoảng tính mờ của mọi hạng từ trong ĐSGT. Theo (3) của Mệnh đề 1.3, $\forall x \in \boldsymbol{X}$, $\{\mathfrak{I}(hx): \forall h \in \boldsymbol{H}\}$ là một phân hoạch của khoảng tính mờ $\mathfrak{I}(x)$ và được tính toán bằng thuật toán sau.

Thuật toán 2.1. Tính phân hoạch các khoảng tính mờ độ sâu k+1 của khoảng tính mờ độ sâu k ($\mathfrak{I}_k(x)$).

Inputs: $x \in \boldsymbol{X}_k$, $\mathfrak{I}_k(x)$ và $\mu(h)$ $\forall h \in \boldsymbol{H} = \{h_{-q}, h_{-q+1}, ..., h_{-1}, h_1, h_2, ..., h_p\}$

Outputs: $\{\mathfrak{I}(hx): \forall h \in \boldsymbol{H}\}$ tập phân hoạch các khoảng tính mờ độ sâu k+1 của $\mathfrak{I}_k(x)$ và tương ứng là tập $\{hx: \forall h \in \boldsymbol{H}\}$

Actions:

(*Step*1) Đặt tập chỉ số $\boldsymbol{J}$,

$$\boldsymbol{J} = \begin{cases} \{-q, -q+1, \ldots, -1, 1, \ldots, p\} & \text{nếu } Sign(h_p x)=1, \\ \{p, p-1, \ldots, 1, -1, \ldots, -q\} & \text{ngược lại} \end{cases}$$

Không xét chỉ số 0, ký hiệu $j_i \in \boldsymbol{J}$ với i=1,..., p+q.

(*Step*2) Tính khoảng tính mờ xuất phát,

$$\mathfrak{I}(h_{j1}x) = (lmp(\mathfrak{I}_k(x)), lmp(\mathfrak{I}_k(x) + \mu(h_{j1}).|\mathfrak{I}_k(x)|],$$

nếu $\mathfrak{I}_k(x)$ là khoảng tính mờ đóng trái thì $\mathfrak{I}_k(h_{j1}x)$ cũng đóng trái.

(*Step*3) Đặt $\boldsymbol{I}_{k+1}(x) = \{\mathfrak{I}_{k+1}(h_{j1}x)\}$.

(*Step*4) Lặp theo i = 2,..., p+q, để tính khoảng tính mờ tiếp theo

$$\mathfrak{I}_{k+1}(h_{ji}x) = (rmp(\mathfrak{I}_{k+1}(h_{ji-1}x)), rmp(\mathfrak{I}_{k+1}(h_{ji-1}x) + \mu(h_{ji}).|\mathfrak{I}_k(x)|].$$

Return: Tập phân hoạch $\{\mathfrak{I}_{k+1}(hx) : \forall h \in \boldsymbol{H}\}$ và tập $\{hx: \forall h \in \boldsymbol{H}\}$.

End.□

Trong đó *rmp* và *lmp* là điểm mút phải và điểm mút trái của khoảng tính mờ. Kết quả tập phân hoạch $\{\mathfrak{I}_{k+1}(hx) : \forall h \in \boldsymbol{H}\}$ gồm các khoảng tính mờ độ sâu k+1 có thứ tự tương ứng với thứ tự ngữ nghĩa các hạng từ sinh bằng cách tác động các gia tử lên x. Bước 3 của thuật toán 2.1 trên lặp trên các gia tử trong $\boldsymbol{H}$ theo thứ tự tương ứng với thứ tự ngữ nghĩa của các hạng từ sinh $\{hx : \forall h \in \boldsymbol{H}\}$ (xác định bởi bước 1). Điểm mút trái của khoảng tính mờ tiếp theo chính là điểm mút phải của khoảng tính mờ trước đó, khoảng tính mờ xuất phát tương ứng với hạng từ có ngữ nghĩa bé nhất được tính trong bước 2.

Dựa trên Mệnh đề 1.3 cùng với Định nghĩa 1.8 về khoảng tính mờ, dễ dàng chứng minh được mệnh đề sau.

Mệnh đề 2.1. Thuật toán 2.1 là đúng đắn với độ phức tạp được giới hạn bởi $O(|\boldsymbol{H}|)$.

Tiếp theo chúng ta thiết kế thuật toán tính tập tất cả các khoảng tính mờ của các hạng từ từ mức 1 đến mức k_{max}, tức $\boldsymbol{I}_{(\boldsymbol{k_{max}})}$, với $k_{max} \geq 1$ cho trước.

Thuật toán 2.2. Tính tập các khoảng tính mờ độ sâu từ 1 đến k_{max} - $\boldsymbol{I}_{(k_{max})}$

Inputs: Các tham số mờ $fm(\boldsymbol{c}^-)$, $\mu(h)$, $\forall h \in \boldsymbol{H}$, và số nguyên $k \geq 1$

Outputs: Tập tất cả các khoảng tính mờ $\boldsymbol{I}_{(k_{max})}$ và tập $\boldsymbol{X}_{(k_{max})}$ tương ứng

Actions:

(*Step*1) Đặt $\boldsymbol{X}_1 = \{\boldsymbol{c}^-, \boldsymbol{c}^+\}$, tính

$$\boldsymbol{I}_1 = \{\mathfrak{I}_1(\boldsymbol{c}^-), \mathfrak{I}_1(\boldsymbol{c}^+)\},$$

với $\mathfrak{I}_1(\boldsymbol{c}^-) = [0, fm(\boldsymbol{c}^-)]$ và $\mathfrak{I}_1(\boldsymbol{c}^+) = (fm(\boldsymbol{c}^-), 1]$,

(*Step*2) Lặp với $k = 2$ đến k_{max}, tính

$$\boldsymbol{I}_k = \cup_{x \in \boldsymbol{X}_{k-1}} \{\mathfrak{I}_k(hx): \forall h \in \boldsymbol{H}\} \text{ và tập } \boldsymbol{X}_k \text{ tương ứng,}$$

trong đó tập $\{\mathfrak{I}_k(hx): \forall h \in \boldsymbol{H}\}$ với mỗi $x \in \boldsymbol{X}_{k-1}$ được tính dựa trên $\mathfrak{I}_{k-1}(x) \in \boldsymbol{I}_{k-1}$ theo thuật toán 2.1.

Return: Tập $\boldsymbol{I}_{(k_{max})} = \bigcup_{k=1\ldots k_{max}} \{\mathfrak{I}_k(x) : \forall x \in \boldsymbol{X}_k\}$ và tập $\boldsymbol{X}_{(k_{max})} = \bigcup_{k=1\ldots k_{max}} \boldsymbol{X}_k$.

End.□

Mệnh đề 2.2. Thuật toán 2.2 là đúng đắn với độ phức tạp được giới hạn bởi $O(|\boldsymbol{H}|^{k-1})$.

Chứng minh. Theo (2) của Mệnh đề 1.3 và Định nghĩa 1.8, hiển nhiên bước 1 của thuật toán tính đúng phân hoạch mức k=1, $\boldsymbol{I}_1$. Bước 2 của thuật toán lặp theo mức khoảng tính mờ k từ 2 đến k_{max}, tại mỗi mức sử dụng Thuật toán 2.1 để tính phân hoạch tất cả các khoảng tính mờ mức k ($\boldsymbol{I}_k$) dựa trên phân hoạch $\boldsymbol{I}_{k-1}$ đã tính trước đó với mỗi hạng từ $x \in \boldsymbol{X}_{k-1}$. Khi tập $\boldsymbol{I}_{k-1}$ là phân hoạch của [0,1] và $\{\mathfrak{I}_k(hx): \forall h \in \boldsymbol{H}\}$ là phân hoạch của $\mathfrak{I}_{k-1}(x) \in \boldsymbol{I}_{k-1}$, $\forall x \in \boldsymbol{X}_{k-1}$ (thuật toán 2.1), do đó tập $\boldsymbol{I}_{k-1}$ cũng là phân hoạch của [0,1]. Bằng phương pháp quy nạp, xuất phát từ $\boldsymbol{I}_1$, mỗi lần lặp trong bước 2 tính đúng phân hoạch mức k ($\boldsymbol{I}_k$) dựa trên phân hoạch mức k-1 ($\boldsymbol{I}_{k-1}$).

Mặt khác, theo Mệnh đề 2.1 ta có số bước để tính phân hoạch mức k là $|\boldsymbol{X}_{k-1}|.|\boldsymbol{H}|$, khi đó số bước để tính tất cả các phân hoạch theo giới hạn mức k_{max} là $2.|\boldsymbol{H}| + ... + 2.|\boldsymbol{H}|^{k_{max}-1} \Rightarrow$ độ phức tạp của Thuật toán 2.2 được giới hạn bởi $O(|\boldsymbol{H}|^{k-1})$.■

Như vậy, từ hai Thuật toán 2.1 và 2.2, nếu cho một ĐSGT với các tham số mờ $fm(\boldsymbol{c}^-)$, $fm(\boldsymbol{c}^+)$, $\mu(h)$: $\forall h \in \boldsymbol{H}$ và một số $k \geq 1$ nguyên, thì chúng ta có thể liệt kê tất cả các hạng từ có độ dài từ 1 đến k cùng với tập tất cả các khoảng tính mờ của chúng. Thực tế các mô hình ứng dụng thường áp dụng các giá trị ngôn ngữ với độ dài không quá lớn, do vậy khi cố định độ dài tối đa k, hai thuật toán trên có thể áp dụng cho việc tính toán trong các mô hình ứng dụng dựa trên ĐSGT.

Tiếp theo chúng ta khảo sát đặc trưng quan hệ ngữ nghĩa của các hạng từ. Dựa trên các tính chất trong ĐSGT, một hạng từ mang ngữ nghĩa của một hạng từ khác được dùng để sinh ra nó. Chẳng hạn *Very old* mang ngữ nghĩa của *old*, và *Very Little old* cũng mang ngữ nghĩa của *old*. Hai hạng từ *Very old* và *Very Little old* đều mang ngữ nghĩa (hay kế thừa ngữ nghĩa) của *old*, ta gọi *Very old* và *Very Little old* có quan hệ ngữ nghĩa. Đây là đặc trưng có quan hệ ngữ nghĩa của các hạng từ trong ĐSGT và được hình thức hóa bằng định nghĩa sau.

Định nghĩa 2.1. Với $\forall x,y \in \boldsymbol{X}$ được gọi là có quan hệ ngữ nghĩa với nhau (ký hiệu $x \,\Xi\, y$) nếu $\exists z \in \boldsymbol{X}$, $x = h_{i_n}...h_{i_1}z$, $y = h_{j_m}...h_{j_1}z$.

Khi x và y có quan hệ ngữ nghĩa, ta nói rằng khoảng tính mờ $\mathfrak{I}(z)$ bao hàm hai khoảng tính mờ $\mathfrak{I}(x)$ và $\mathfrak{I}(y)$, hay z bao hàm ngữ nghĩa của x và y (ký hiệu $z = \Xi(x,y)$). Theo định nghĩa, rõ ràng hai hạng từ x, y có quan hệ ngữ nghĩa nếu chúng có dạng $x = h_{i_n}...h_{i_1}\boldsymbol{c}$, $y = h_{j_m}...h_{j_1}\boldsymbol{c}$. Nếu $h_{i_1} = h_{j_1} = h'_1$, $h_{i_2} = h_{j_2} = h'_2$,... thì ta có $z = h'_1\boldsymbol{c}$ hoặc $z = h'_1h'_2\boldsymbol{c}$ 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, chúng ta cần phải xác định z sao cho ngữ nghĩa càng gần với x, y càng tốt. Tức z phải được chọn sao cho có độ dài lớn nhất.

Để tiện về sau ta ký hiệu $z = \Xi_{max}(x,y)$ là hạng từ bao hàm ngữ nghĩa của x và y với độ dài lớn nhất.

Tiếp theo chúng ta sẽ hình thức hóa mức độ gần nhau của hai hạng từ, xét ở góc độ ngữ nghĩa, bằng định nghĩa sau.

Định nghĩa 2.2. Với bất kỳ $x,y \in \boldsymbol{X}$ và $x\ \Xi\ y$, mức độ gần nhau của x và y theo quan hệ ngữ nghĩa được định nghĩa như sau:

$$sm(x,y) = \frac{m}{\max(k,l)}\left(1 - \left|\upsilon(x) - \upsilon(y)\right|\right),$$

trong đó $m = len(z)$ là độ dài (*length*) của hạng từ $z = \Xi_{max}(x,y)$, tương tự $k = len(x)$ và $l = len(y)$.

Sau đây chúng ta khảo sát một số tính chất của hàm *sm* trong định nghĩa trên.

Mệnh đề 2.3. Xét mức độ gần nhau *sm* của các hạng từ (Định nghĩa 2.2), ta có:

(1) Hàm *sm* là đối xứng, tức là $sm(x,y) = sm(y,x)$,

(2) x, y không có quan hệ ngữ nghĩa $\Leftrightarrow sm(x,y) = 0$,

(3) $sm(x,y) = 1 \Leftrightarrow x = y$,

(4) $\boldsymbol{c}, \boldsymbol{c}' \in \boldsymbol{G}, \boldsymbol{c} \neq \boldsymbol{c}', x \in \boldsymbol{H}(\boldsymbol{c}), y \in \boldsymbol{H}(\boldsymbol{c}') \Leftrightarrow sm(x,y) = 0$, và $\boldsymbol{c} \in \boldsymbol{G}, x,y \in \boldsymbol{H}(\boldsymbol{c}) \Leftrightarrow sm(x,y) > 0$,

(5) $\forall x, y, z \in \boldsymbol{X_k}, x \le y \le z \Rightarrow sm(x,z) \le sm(x,y)$ và $sm(x,z) \le sm(y,z)$.

Chứng minh. (1) dễ dàng suy ra từ định nghĩa.

Chứng minh (2), hiển nhiên ta có $m = 0$ khi và chỉ khi x, y không có quan hệ ngữ nghĩa. Vậy, nếu x và y không có quan hệ ngữ nghĩa $\Rightarrow m = 0 \Rightarrow sm(x,y) = 0$. Ngược lại, khi $sm(x,y) = 0 \Rightarrow m = 0$ hoặc $(1-|\upsilon(x)-\upsilon(y)|) = 0$. Trường hợp $m = 0$ tức x, y không có quan hệ ngữ nghĩa, trường hợp $(1-|\upsilon(x)-\upsilon(y)|) = 0 \Rightarrow |\upsilon(x)-\upsilon(y)| = 1 \Rightarrow x = \mathbf{0}, y = \mathbf{1}$ hoặc $x = \mathbf{1}, y = \mathbf{0} \Rightarrow x, y$ không có quan hệ ngữ nghĩa.

Chứng minh (3), do $m \le max(k,l)$ và $0 \le \upsilon(x), \upsilon(y) \le 1$ nên $sm(x,y) = 1 \Leftrightarrow m = max(k,l)$ và $|\upsilon(x)-\upsilon(y)| = 0 \Leftrightarrow x = y$, ta có (3). (4) được suy ra từ (2) vì theo Định nghĩa 2.1, $\boldsymbol{c} \neq \boldsymbol{c}', x \in \boldsymbol{H}(\boldsymbol{c}), y \in \boldsymbol{H}(\boldsymbol{c}') \Leftrightarrow x, y$ không có quan hệ ngữ nghĩa, và $x,y \in \boldsymbol{H}(\boldsymbol{c}) \Leftrightarrow x,y$ có quan hệ ngữ nghĩa.

Chứng minh (5), theo giả thiết $x \le y \le z \Rightarrow \upsilon(x) \le \upsilon(y) \le \upsilon(z) \Rightarrow 1\text{-}|\upsilon(x)\text{-}\upsilon(z)| \le 1\text{-}|\upsilon(x)\text{-}\upsilon(y)|$ và $1\text{-}|\upsilon(x)\text{-}\upsilon(z)| \le 1\text{-}|\upsilon(y)\text{-}\upsilon(z)|$. Mặt khác, cũng theo giả thiết $x,y,z \in \boldsymbol{X_k} \Rightarrow len(x) = len(y) = len(z) = k$. 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 $x \le y \le z$, dễ dàng suy ra $len(w_1) \ge len(w_3)$ và $len(w_2) \ge len(w_3)$. Vậy theo Định nghĩa 2.2 ta có $sm(x,y) \le sm(x,z)$ và $sm(y,z) \le sm(x,z) \Rightarrow đpcm$.■

Định nghĩa 2.2 ở trên cho phép xem xét hai hạng từ x, y có thể sử dụng hạng từ z đại diện mà không làm mất nhiều ngữ nghĩa của x, y bằng cách dựa vào sm. Nếu $sm = 0$ thì không thể kết nhập vì chúng không có quan hệ ngữ nghĩa, ngược lại chúng ta chọn $z = \sim_{max}(x,y)$ là hạng từ xác định khoảng tính mờ bé nhất (hay mức cao nhất) bao hàm $\mathfrak{I}(x)$ và $\mathfrak{I}(y)$. Đặc biệt tính chất (5) của Mệnh đề 2.3 cho thấy phép sử dụng thay thế dựa trên hàm sm giúp giảm thiểu sự mất mát ngữ nghĩa của các hạng từ. Trường hợp $k = len(x) < l = len(y)$ và y kế thừa ngữ nghĩa từ x hay $y = h_m...h_1x$, khi đó $z = \Xi_{max}(x,y) = x$. Việc sử dụng z thay thế cho x và y được xem như phép kết nhập x, y thành hạng từ z.

Như bất kỳ phép kết nhập nào, phép kết nhập dựa trên Định nghĩa 2.2 cũng có thể làm mất thông tin trong các mô hình ứng dụng. Tuy nhiên, về mặt ngữ nghĩa của các giá trị ngôn ngữ thì phép kết nhập này sẽ giảm bớt sự mất mát ngữ nghĩa của chúng, vì ở đây chúng ta mở rộng các khái niệm mờ. Rõ ràng, sm càng bé thì phép kết nhập càng làm tăng tính mờ của các khái niệm mờ được kết nhập, và dẫn đến nó phủ các khái niệm mờ khác. Khi $sm = 1$, có nghĩa $x = y$, dẫn đến $\mathfrak{I}_m(z) = \mathfrak{I}_k(x) = \mathfrak{I}_l(y)$. Trong các mô hình ứng dụng, chúng ta có thể đặt ngưỡng đối với giá trị sm để tránh mất mát thông tin quá nhiều, hơn nữa có thể gia cố thêm các hàm đo mật độ thông tin tập trung trong các khoảng tính mờ của khái niệm mờ để xác định mức độ kết nhập.

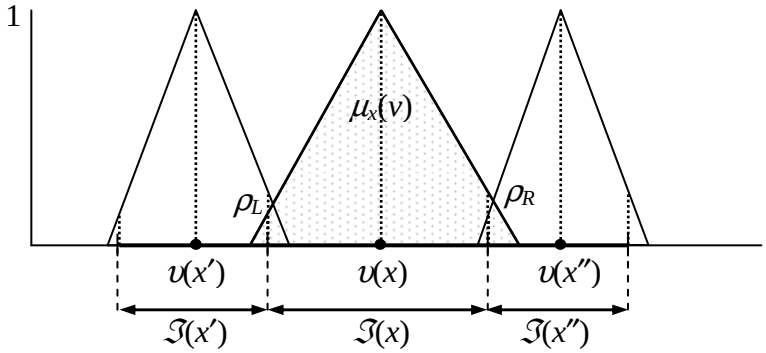
2.2.2 Thuật toán sinh luật mờ dựa trên hệ khoảng tính mờ

Dựa trên hệ khoảng tính mờ, miền của mỗi thuộc tính x_j được phân hoạch bởi một tập hạng từ mức k_j, tức $\boldsymbol{X}_{k_j} = \{... \le x_{k_j}' \le x_{k_j} \le x_{k_j}'' \le ... \}$ trong ĐSGT $\mathfrak{AX}_j$. Mỗi

hạng từ $x \in \boldsymbol{X_{kj}}$ được thiết kế hàm định lượng ngữ nghĩa dựa trên nguyên tắc càng gần tâm ($\upsilon(x_{kj})$ - *upsilon*) giá trị hàm càng lớn và bằng 1 tại tâm, hàm sẽ giảm dần về hai phía và không vượt khỏi tâm của hai hạng từ láng giềng $\upsilon(x_{kj}')$ và $\upsilon(x_{kj}'')$. Điều này nhằm đảm bảo tính thứ tự ngữ nghĩa của các hạng từ trong ĐSGT. Có thể thiết kế hàm dạng tam giác hay dạng hình chuông. Tuy nhiên thực tế để mềm dẻo và dễ dàng trong khi áp dụng, luận án xây dựng hàm dạng tam giác (Hình 2.1) với hai tham số ρ_L, ρ_R để xác định giá trị hàm tại hai điểm đầu mút của khoảng tính mờ tương ứng (ρ_L, $\rho_R > 0$), công thức tính như sau:

$$\mu_x(v) = \min\left(\begin{array}{l} \max\left(\dfrac{v.(1-\rho_L) - lmp(\Im(x)) + \upsilon(x).\rho_L}{\upsilon(x) - lmp(\Im(x))}, 0\right), \\ \max\left(\dfrac{rmp(\Im(x)) + v.(\rho_R - 1) - \rho_R.\upsilon(x)}{rmp(\Im(x)) - \upsilon(x)}, 0\right) \end{array}\right), \tag{2.4}$$

trong đó ρ_L và ρ_R là hai tham số xác định giá trị hàm $\mu_x(v)$ tại điểm mút trái $v = lmp(\Im(x))$ và điểm mút phải $v = rmp(\Im(x))$ của khoảng tính mờ $\Im(x)$.



Hình 2.1: Hàm định lượng dạng tam giác của các hạng từ

Áp dụng lược đồ đã trình bày trong Mục 2.1 để xây dựng một hệ luật mờ phân lớp, ta gọi hệ này là hệ luật khởi đầu. Trong phần này, dựa trên lưới phân hoạch hệ các khoảng tính mờ $\boldsymbol{I_k}$, chúng ta sẽ thiết kế thuật toán sinh hệ luật mờ như sau.

Thuật toán 2.3. Sinh các luật mờ từ tập dữ liệu mẫu dựa trên hệ phân hoạch các khoảng tính mờ (*Initial Fuzzy Rules Generation* - ***IFRG*1**).

Inputs:

- Tập dữ liệu mẫu $\boldsymbol{D} = \{ (p_i; c_i) \mid i=1, ..., N \}$, $p_i = (d_{i,1}, ..., d_{i,n}) \in \boldsymbol{P}$, $c_i \in \boldsymbol{C} = \{C_1, ..., C_m\}$, n là số thuộc tính, m là số lớp, N là số mẫu dữ liệu;

- Bộ các tham số mờ gia tử của ĐSGT cho mỗi thuộc tính $\boldsymbol{PAR}_j = \{ fm_j(\boldsymbol{c}^-), fm_j(\boldsymbol{c}^+), \mu_j(h) \ \forall h \in \boldsymbol{H} \}$, $j = 1, ..., n$;

- Mức phân hoạch k_j các khoảng tính mờ trên miền của các thuộc tính;

Outputs: Tập các luật mờ $\boldsymbol{S}_0 = \{R_1, ..., R_M\}$

Actions:

(*Step*1) Khởi tạo tập luật $\boldsymbol{S}_0 = \varnothing$,

(*Step*2) Tính phân hoạch các khoảng tính mờ trên miền các thuộc tính x_j theo tham số mờ gia tử $\boldsymbol{PAR}_j$ và mức phân hoạch k_j,

$$\boldsymbol{I}_{kj} = \{ \mathfrak{I}_{kj}(x_{kj,1}), \mathfrak{I}_{kj}(x_{kj,2}), ... \}, \boldsymbol{X}_{kj} = \{x_{kj,1}, x_{kj,2}, ... \}, j = 1, 2, ..., n.$$

(*Step*3) Lặp trên mỗi mẫu dữ liệu $(p_i; c_i) \in \boldsymbol{D}$, thực hiện:

(*Step*3.*a*) Xác định giá trị ngôn ngữ có khoảng tính mờ chứa $d_{i,j} \in p_i$,

$$\{ A_{i,j} = x_{kj,i^*} \mid x_{kj,i^*} \in \boldsymbol{X}_{kj} \text{ và } d_{i,j} \in \mathfrak{I}_{kj}(x_{kj,i^*}), j = 1, 2, ..., n \}$$

(*Step*3.*b*) Tạo một truyển vế trái gồm n giá trị ngôn ngữ trên

$$\boldsymbol{A}_q = (A_{i,1}, A_{i,2}, ..., A_{i,n}),$$

(*Step*3.*c*) Sinh luật mới theo $\boldsymbol{A}_q$ và thêm vào tập luật $\boldsymbol{S}_0$,

$$\boldsymbol{S}_0 = \boldsymbol{S}_0 \cup \{\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_q\},$$

trong đó $\boldsymbol{C}_q = \arg\max_{C_h} \{ c(\boldsymbol{A}_q \Rightarrow C_h) \mid h = 1,...,m \}$.

Return: Tập $\boldsymbol{S}_0$.

End.□

Mệnh đề 2.4. Độ phức tạp của thuật toán ***IFRG*1** được giới hạn bởi $O(n.|\boldsymbol{D}|^2)$.

Chứng minh. Dễ dàng nhận thấy thời gian tính toán phân hoạch mức k_j cho các thuộc tính ở bước 2 là $O(n.|\boldsymbol{X}_{(k^*)}|)$, trong đó $k^* = \max\{ k_j : j = 1,...,n \}$. Thông thường chúng ta có $|\boldsymbol{X}_{(k^*)}| << |\boldsymbol{D}|$ (*).

Trong bước 3, thời gian tính toán để sinh tuyển các điều kiện vế trái là $O(n.|\boldsymbol{D}|)$. Trường hợp cực đoan kích thước của tập các tuyển điều kiện vế trái là $|\boldsymbol{D}|$, tức mỗi mẫu dữ liệu xác định một tuyển vế trái. Mỗi tuyển vế trái được tính toán độ tin cậy và độ hỗ trợ theo lần lượt mỗi kết luận là một nhãn phân lớp trong tập dữ liệu, để chọn và sinh luật mờ có độ hỗ trợ cao nhất. Vậy thời gian của bước này là $O(m.|\boldsymbol{D}|^2)$ (**).

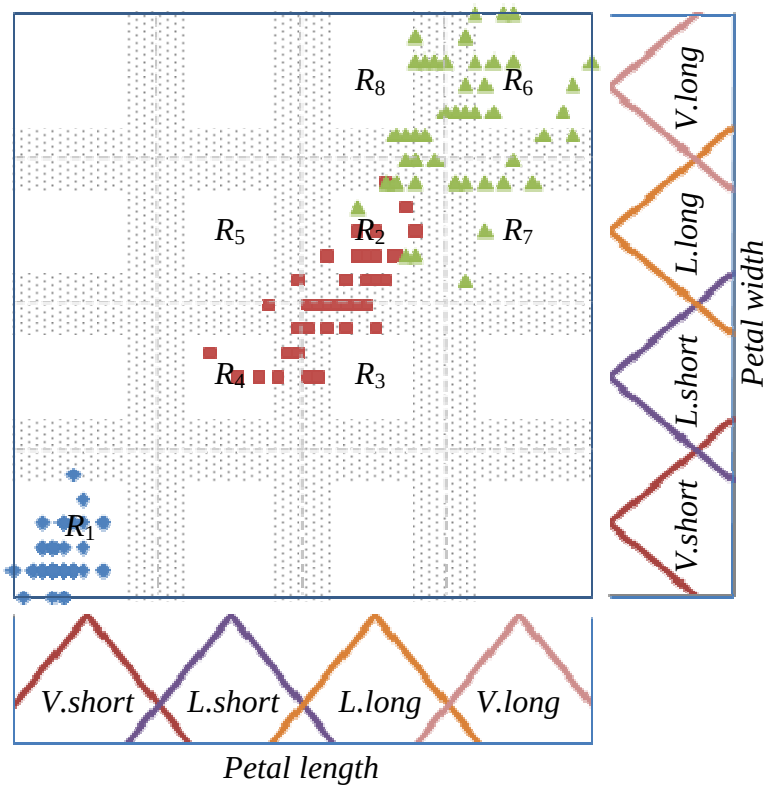
Từ (*) và (**) ta suy ra độ phức tạp của thuật toán ***IFRG*1** được giới hạn bởi $O(n.|\boldsymbol{D}|^2)$ (để ý rằng $m << |\boldsymbol{D}|$ và được xem như hằng số) $\Rightarrow$ *đpcm*. ■

Ví dụ 2.1. Minh họa phương pháp sinh luật của thuật toán ***IFRG*1** đối với tập dữ liệu mẫu của bài toán *IRIS* trong [76].

IRIS là một loài cây hoa. Tập dữ liệu mẫu phân lớp các loại hoa này được tạo bởi R.A. Fisher và đưa ra năm 1988, gồm 150 mẫu chia đều cho 3 loại hoa *Iris-setosa*, *Iris-versicolor* và *Iris-virginica*. Mỗi mẫu có 4 thuộc tính gồm độ dài đài hoa (*sepal length*), độ rộng đài hoa (*sepal width*), độ dài cánh hoa (*petal length*) và độ rộng cánh hoa (*petal width*). Hiện nay nó được rất nhiều tác giả sử dụng để thử nghiệm cho các mô hình giải bài toán phân lớp [17], [23], [30]-[32], [40], [42]-[43], [50], [53], [56], [60], [74], [77]. Để đơn giản cho việc minh họa phương pháp sinh luật của thuật toán trên, chúng tôi áp dụng hai thuộc tính đã được trích chọn trong [19] là *petal length* (*PL*) và *petal width* (*PW*) với phân bố dữ liệu theo từng lớp trong Hình 2.2. Gọi bài toán với hai thuộc tính này là *IRIS*2.

Ta ký hiệu đại số gia tử cho thuộc tính *PL*, *PW* là $\mathcal{AX}_1$, $\mathcal{AX}_2$, đều có cấu trúc như sau: $\boldsymbol{c}^- = short$, $\boldsymbol{c}^+ = long$, $\boldsymbol{H} = \{L, V\}$. Các tham số mờ gia tử đều cho giống nhau: $fm_j(\boldsymbol{c}^-) = fm_j(\boldsymbol{c}^+) = 0.5$, $\mu_j(L) = \mu_j(V) = 0.5$ và mức phân hoạch $k_j = 2$ đối với cả hai thuộc tính *PL*, *PW*. Tính toán hệ phân hoạch các khoảng tính mờ trên miền của hai thuộc tính *PL* và *PW* chúng ta có lưới phân hoạch trong Hình 2.2.

Áp dụng thuật toán ***IFRG1*** với hàm định lượng ngữ nghĩa gán cho các giá trị ngôn ngữ trong ĐSGT tính theo công thức (2.4), chọn $\rho_L = \rho_R = 0.3$ (Hình 2.3). Hệ luật sinh ra gồm 8 luật trong Bảng 2.1, mỗi luật được tính độ tin cậy, độ hỗ trợ và trọng số theo các công thức (1.15)-(1.18). Các luật R_1, R_4, R_5 và R_6 đều có độ tin cậy cũng như các trọng số bằng 1 vì miền quyết định (tính cả biên ngoài theo chân các hàm định lượng ngữ nghĩa của vùng phân hoạch - Hình 2.3) của chúng không chứa các mẫu dữ liệu khác lớp. Luật R_2 có trọng số CF^3, CF^4 nhỏ vì miền quyết định của nó chứa khá nhiều mẫu dữ liệu của lớp *Virginica*. Tất cả các luật đều có $CF^3 = CF^4$ vì miền quyết định của chúng đều chứa mẫu dữ liệu của 1 hoặc 2 lớp (xem công thức (1.17), (1.18), (1.20) và (1.21)).



Hình 2.2: Sơ đồ phân hoạch trên miền của thuộc tính *PL*, *PW*

Bảng 2.1: Danh sách luật sinh bởi thuật toán ***IFRG1*** cho bài toán *IRIS*2

R_1	**if** *petal-length* is *V.short*, and *petal-width* is *V.short* **then** *Setosa* ($\boldsymbol{c} = 1$, $\boldsymbol{s} = 0.15$, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)

R_2	**if** *petal-length* is *L.long*, and *petal-width* is *L.long* **then** *Versicolor* ($\boldsymbol{c}$ = 0.747, $\boldsymbol{s}$ = 0.099, CF^1=0.747, CF^2 = 0.62, CF^3 = CF^4 = 0.493)
R_3	**if** *petal-length* is *L.long*, and *petal-width* is *L.short* **then** *Versicolor* ($\boldsymbol{c}$ = 0.998, $\boldsymbol{s}$ = 0.034, CF^1 = 0.998, CF^2 = 0.997, CF^3 = CF^4 = 0.996)
R_4	**if** *petal-length* is *L.short*, and *petal-width* is *L.short* **then** *Versicolor* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.043, CF^1= CF^2 = CF^3 = CF^4 = 1)
R_5	**if** *petal-length* is *L.short*, and *petal-width* is *L.long* **then** *Versicolor* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.006, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_6	**if** *petal-length* is *V.long*, and *petal-width* is *V.long* **then** *Virginica* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.058, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_7	**if** *petal-length* is *V.long*, and *petal-width* is *L.long* **then** *Virginica* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.022, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_8	**if** *petal-length* is *L.long*, and *petal-width* is *V.long* **then** *Virginica* ($\boldsymbol{c}$ = 0.988, $\boldsymbol{s}$ = 0.034, CF^1 = 0.988, CF^2 = 0.982, CF^3 = CF^4 = 0.977)

Áp dụng cả hai phương pháp lập luận là *single winner rule* và *weighted vote*, kết quả tỷ lệ phân lớp đúng trên tập mẫu thể hiện trong Bảng 2.2. Cả hai phương pháp lập luận tại các trường hợp trọng số luật được tính toán cho tỷ lệ phân lớp đạt 94.67%, riêng luật không sử dụng trọng số (CF^0) cho kết quả thấp hơn. Điều này một lần nữa thể hiện tính hiệu quả của đánh giá trọng số luật trong các phương pháp lập luận dựa trên hệ luật mờ.

Bảng 2.2: Tỷ lệ (%) số mẫu phân lớp đúng của hệ luật trong bảng 2.1 theo các đánh giá trọng số luật với hai phương pháp lập luận

Trọng số luật được sử dụng	Phương pháp lập luận	
	single winner rule	*weighted vote*
Hệ luật sử dụng CF^0	92	92.67
Hệ luật sử dụng CF^1	94.67	94.67
Hệ luật sử dụng CF^2	94.67	94.67

Hệ luật sử dụng CF^3	94.67	94.67
Hệ luật sử dụng CF^4	94.67	94.67

2.2.3 Phương pháp rút gọn bằng phép hợp các luật mờ

Hệ luật khởi đầu sinh bởi thuật toán ***IFRG*1** có độ dài giống nhau và đúng bằng tập các thuộc tính. Mỗi luật mờ được sinh từ một siêu hộp trong không gian phân hoạch bởi hệ khoảng tính mờ, tương ứng là các giá trị ngôn ngữ. Khi chọn mức phân hoạch khoảng tính mờ thấp (tức k_j nhỏ) cơ hội mỗi siêu hộp chứa nhiều mẫu dữ liệu ở khác lớp nhau rất cao, do đó luật sinh ra không có tính phân biệt lớn giữa các lớp hay luật mờ có tính phổ quát cao. Thông thường chúng ta xuất phát với k_j lớn để sinh các luật có tính phân biệt (hay tính cá thể) cao, đảm bảo hiệu quả phân lớp đối với tập dữ liệu mẫu. Mặt khác theo mục tiêu (1.6) thì hệ luật phải tinh gọn và hiệu quả, đảm bảo tính phổ quát để thực hiện mục tiêu dự đoán. Phép hợp các luật mờ được đề xuất với mong muốn tìm những luật mang tính phổ quát hơn, tức bao trùm các luật khác nhưng đảm bảo hiệu quả phân lớp.

Dựa trên định nghĩa mức độ gần nhau của hai khoảng tính mờ (*Định nghĩa 2.2*), mỗi luật mờ xác định một tập các giá trị ngôn ngữ ở vế trái $A_q = (A_{q,1}, ..., A_{q,n})$ và tương ứng là tập các khoảng tính mờ. Chúng ta định nghĩa mức độ có thể hợp của hai luật mờ như sau:

Định nghĩa 2.3. Với hai luật $R_q = (A_q \Rightarrow C_q)$ và $R_p = (A_p \Rightarrow C_p)$, mức độ có thể hợp của hai luật R_q, R_p, ký hiệu $itg(R_q,R_p)$, được xác định dựa trên mức độ gần nhau của các thành phần như sau:

i) $itg(R_q, R_p) = 0$, nếu $C_q \neq C_p$, ngược lại

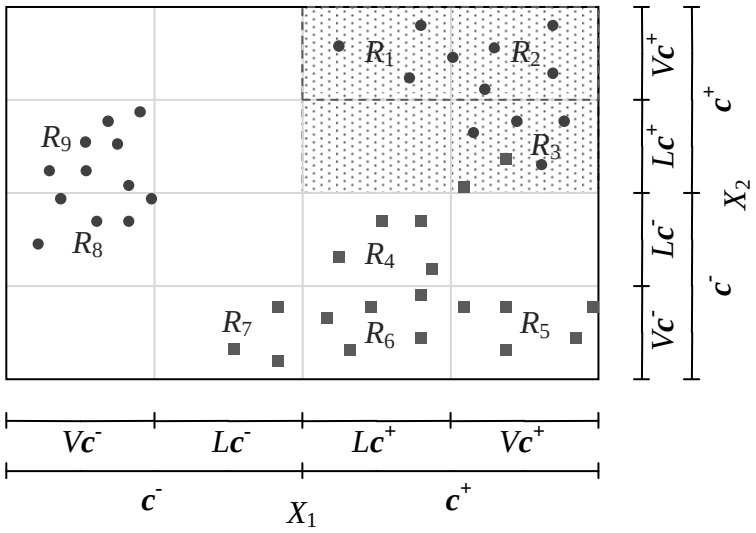
ii) $itg(R_q, R_p) = T_{ex}(sm(A_{p,1}, A_{q,1}), ..., sm(A_{p,n}, A_{q,n}))$,

trong đó, T_{ex} là một *t*-chuẩn mở rộng *n* ngôi, trong luận án này chúng tôi dùng toán tử *min*, hàm *sm* được xác định từ Định nghĩa 2.2.

Theo định nghĩa này hai luật có giá trị *itg* càng lớn thì mức độ có thể hợp càng cao, khi *itg* = 1 hai luật giống nhau và *itg* = 0 hai luật không thể hợp vì hoặc chúng ở hai lớp khác nhau hoặc các giá trị ngôn ngữ trong điều kiện luật không có quan hệ ngữ nghĩa. Luật mới được sinh ra có vế phải giống với hai luật trên, vế trái bao gồm các giá trị ngôn ngữ bao hàm ngữ nghĩa các cặp ($A_{q,j}$, $A_{p,j}$) và có độ dài lớn nhất.

Để đảm bảo hiệu quả phân lớp cũng như tránh mất mát thông tin quá nhiều, chúng ta đặt ngưỡng mức độ có thể hợp đối với một hệ luật. Khi đó, xét từng cặp luật nếu giá trị của hàm *itg* không nhỏ hơn ngưỡng θ_{itg} cho trước thì thay thế chúng bằng luật mới tương ứng, lặp lại quá trình này cho đến khi không có cặp luật nào trong hệ thỏa mãn.

Ví dụ 2.2. Minh họa phương pháp hợp các luật. Giả sử cho tập dữ liệu có 2 thuộc tính như trong Hình vẽ 2.3, với mức phân hoạch các khoảng tính mờ $k_j = 2$ và các tham số mờ gia tử đều bằng 0.5, hệ luật sinh bởi thuật toán ***IFRG*1** gồm 9 luật tương ứng 9 hình chữ nhật có chứa mẫu dữ liệu (Hình 2.3).



Hình 2.3: Minh họa phương pháp hợp các luật

Trực quan nhận thấy có thể hợp hai luật R_1 và R_2 hoặc R_2 và R_3, R_4 và R_6 hoặc R_5 và R_6, chẳng hạn hợp cặp $R_1 = ((Lc^+, Vc^+) \Rightarrow class$"•") và $R_2 = ((Vc^+, Vc^+) \Rightarrow class$"•") ta được $R_{12} = ((c^+, Vc^+) \Rightarrow class$"•"). Các luật R_8 và R_9 không thể vì trên thuộc tính X_2 cặp giá trị ngôn ngữ Lc^- và Lc^+ không có quan hệ ngữ nghĩa, tương tự với cặp luật R_6 và R_7 vì cặp giá trị ngôn ngữ Lc^- và Lc^+ trên thuộc tính X_1. Quá trình

hợp có thể được tiếp tục trên các luật mới, khi đó cặp luật R_{12} và R_3 có thể hợp thành một luật mới $R_{123} = ((\boldsymbol{c}^+, \boldsymbol{c}^+) \Rightarrow class$"•") tương ứng với miền quyết định được tô màu trong Hình 2.3.

Ví dụ 2.3. Áp dụng phương pháp hợp để rút gọn hệ luật trong Bảng 2.1. Đặt ngưỡng hợp $\theta_{itg} = 0.1$, kết quả hệ luật thu được gồm 6 luật (Bảng 2.3). Xét các cặp luật trong Bảng 2.1 có vế phải giống nhau, tính toán giá trị hàm đánh giá mức độ hợp $itg(R_6,R_7) = \min(sm(V.long,V.long), sm(V.long,L.long)) = ½.(1-|0.875-0.625|) = 0.125$ (trong đó $\upsilon(V.long) = 0.875$, $\upsilon(L.long) = 0.625$, hàm *sm* tính theo Định nghĩa 2.2), hợp hai luật này thành luật $R_{67} = ((V.long,long) \Rightarrow Virginica)$. Tương tự, cặp luật (R_{67},R_8) được hợp thành $R_{678} = ((long, long) \Rightarrow Virginica)$.

Bảng 2.3- Hệ 6 luật thu được sau khi hợp từ hệ luật trong bảng 2.1 của Ví dụ 2.1

R_1	**if** *petal-length* is *V.short*, and *petal-width* is *V.short* **then** *Setosa*
R_2	**if** *petal-length* is *L.long*, and *petal-width* is *L.long* **then** *Versicolor*
R_3	**if** *petal-length* is *L.long*, and *petal-width* is *L.short* **then** *Versicolor*
R_4	**if** *petal-length* is *L.short*, and *petal-width* is *L.short* **then** *Versicolor*
R_5	**if** *petal-length* is *L.short*, and *petal-width* is *L.long* **then** *Versicolor*
R_6	**if** *petal-length* is *long*, and *petal-width* is *long* **then** *Virginica*

Cho dù hệ luật thu được giảm 25% số luật nhưng hiệu quả phân lớp được nâng cao hơn so với trong Bảng 2.1, đạt 96.67% đối với tất cả các trường hợp áp dụng trọng số luật và phương pháp lập luận *single-winner-rule*, trong khi hệ luật trong Bảng 2.1 chỉ đạt 94.67%. Điều này minh họa cho khả năng ứng dụng của phương pháp hợp nhằm rút gọn hệ luật nhưng vẫn đảm bảo hiệu năng phân lớp.

Một hạn chế của thuật toán sinh luật ***IFRG*1** là việc bỏ qua xem xét ngữ nghĩa của các hạng từ độ dài nhỏ hơn so với mức phân hoạch k_j được chọn để sinh luật. Chẳng hạn Ví dụ 2.1 trên không xét các giá trị ngôn ngữ *short* và *long*. Điều này làm mất tính bình đẳng của các giá trị ngôn ngữ. Hơn nữa, phân hoạch trên Hình 2.2 trực quan ta thấy loại hoa *Setosa* có thể phân lớp chỉ bằng một thuộc tính *petal*

length hoặc *petal width* với ngữ nghĩa của giá trị ngôn ngữ *V.short*. Do đó luật R_1 trong Bảng 2.1 có thể loại bỏ điều kiện của một thuộc tính. Đây có thể coi là sự dư thừa các điều kiện trong vế trái luật.

Phần tiếp theo luận án sẽ đề xuất sử dụng đại số chỉ gồm 2 gia tử để xây dựng hệ khoảng tương tự cho tập hạng từ có độ dài không quá k ($\boldsymbol{X_{(k)}}$) và xây dựng phương pháp sinh luật dựa trên hệ khoảng tương tự.

2.3 Phương pháp sinh luật mờ dựa trên hệ khoảng tương tự

2.3.1 Đại số 2 gia tử

Thực tế các biến ngôn ngữ nói chung và theo tiếp cận của đại số gia tử nói riêng chỉ sai khác nhau các giá trị sinh nguyên thủy $\boldsymbol{G} = \{\boldsymbol{c}^-, \boldsymbol{c}^+\}$ và đây là đặc trưng mang tính phổ quát của chúng. Hơn nữa, tính độc lập ngữ cảnh của các gia tử và liên từ như *AND*, *OR*,... giúp chúng ta trong nghiên cứu và tìm kiếm mô hình cho các gia tử mà không quan tâm đến gia trị sinh nguyên thủy của các biến ngôn ngữ. Dựa trên những đặc trưng này, nhiều tác giả nghiên cứu đã xây dựng các mô hình ứng dụng với tập các gia tử hầu như giống nhau và chỉ gồm một số ít các gia tử. Chẳng hạn trong [10] không sử dụng gia tử, các tác giả trong [12], [20], [46], [50], [53], [57], [60], [74] giới thiệu dùng 1 gia tử *very* hoặc *medium*, các tác giả trong [18], [48], [8] dùng 2 gia tử *positive* và *negative* hoặc *very* và *less*, trong [35], [39], [52], [77] dùng từ 3 đến 4 gia tử.

Trong quá trình nghiên cứu và tiếp cận ĐSGT xây dựng mô hình giải bài toán phân lớp chúng tôi thấy rằng, ĐSGT chỉ gồm hai gia tử, một gia tử dương và một gia tử âm, sẽ có những đặc trưng rất quan trọng trong quá trình ứng dụng. Ta gọi đại số gia tử với hạn chế này là đại số 2 gia tử (ĐS2GT), ký hiệu $\mathcal{AX}^2$.

Bây giờ chúng ta khảo sát một số đặc trưng của ĐS2GT, theo hệ các tiên đề cho đại số gia tử [37], [38] và không mất tính tổng quát, đặt gia tử âm là $\boldsymbol{H}^- = \{L\}$ và gia tử dương là $\boldsymbol{H}^+ = \{V\}$. Hàm dấu của các hạng từ trong ĐS2GT có thể tính trực tiếp mà không sử dụng dạng truy hồi như trong *Định nghĩa 1.6*, vì gia tử V là

dương đối với L và V, ngược lại gia tử L là âm đối với V và L. Ta có công thức tính là:

$$Sign(x) = Sign(h_n...h_1\boldsymbol{c}) = (-1)^{NL(x)}Sign(\boldsymbol{c}), \quad (2.5)$$

trong đó $h_n, h_{n-1}, ..., h_1 \in \{L, V\}$ và $NL(x)$ là số lượng các gia tử L có trong hạng từ x.

Sau đây là mệnh đề khảo sát về kích thước của các tập $\boldsymbol{X_k}$, $\boldsymbol{X_{(k)}}$, $\boldsymbol{I_k}$ và $\boldsymbol{I_{(k)}}$.

Mệnh đề 2.5. Cho $\mathcal{AX}^2$, kích thước các tập $\boldsymbol{X_k}$, $\boldsymbol{X_{(k)}}$, $\boldsymbol{I_k}$, $\boldsymbol{I_{(k)}}$ được tính như sau:

(1) $|\boldsymbol{X}_1| = 5$.

(2) $|\boldsymbol{X}_k| = 2^k$, với $k > 1$.

(3) $|\boldsymbol{X}_{(k)}| = 1+2^{k+1}$.

Để ý rằng $|\boldsymbol{I}_k| = |\boldsymbol{X}_k|$, $|\boldsymbol{I}_{(k)}| = |\boldsymbol{X}_{(k)}|$.

Chứng minh. Trong ĐS2GT, hiển nhiên ta có tập $\boldsymbol{X}_1 = \{\boldsymbol{0}, \boldsymbol{c}^-, \boldsymbol{W}, \boldsymbol{c}^+, \boldsymbol{1}\} \Rightarrow$ (1), $\boldsymbol{X}_2 = \{V\boldsymbol{c}^-, L\boldsymbol{c}^-, L\boldsymbol{c}^+, V\boldsymbol{c}^+\}$. Với $k > 2$, $\boldsymbol{X}_k = \{Lx, Vx : x \in \boldsymbol{X}_{k-1}\} \Rightarrow |\boldsymbol{X}_k| = 2.|\boldsymbol{X}_{k-1}| \Rightarrow$ (2). Ta có $|\boldsymbol{X}_{(k)}| = \sum_{i=1...k} |\boldsymbol{X}_k| = 5 + \sum_{i=2...k} 2^k = 3+\sum_{i=1...k} 2^k = 3+(2.2^k-2) = 1+2^{k+1} \Rightarrow$ (3).■

Đặc trưng rất quan trọng của ĐS2GT là có thể xây dựng hệ phân hoạch các khoảng tương tự của tập các hạng từ $\boldsymbol{X}_{(k)}$ thay thế cho tập $\boldsymbol{X}_k$ và khẳng định được sự tồn tại của hệ này (sẽ trình bày chi tiết ở phần sau). Trên cơ sở phân hoạch hệ khoảng tương tự, phương pháp sinh hệ luật mờ được xây dựng với ngữ nghĩa gồm tập các hạng từ có độ dài không quá k. Điều này nhằm khắc phục được hạn chế của ĐSGT tuyến tính thông thường là chỉ áp dụng được với tập hạng từ độ dài đúng k (thuật toán ***IFRG*1**).

Một đặc điểm khác của ĐS2GT là việc áp dụng các phương pháp tìm kiếm tối ưu tham số mờ gia tử có lợi thế giảm không gian tìm kiếm, vì chúng ta chỉ định nghĩa không gian tìm kiếm cho tham số độ đo tính mờ của phần tử sinh âm ($fm(\boldsymbol{c}^-)$) và độ đo tính mờ của gia tử *Little* ($\mu(L)$), để ý rằng $fm(\boldsymbol{c}^+) = 1- fm(\boldsymbol{c}^-)$, $\mu(V) = 1-\mu(L)$, dẫn đến tốc độ tìm kiếm sẽ nhanh hơn và đạt hiệu quả cao trong ứng dụng.

Từ những khảo sát trên đây, chúng ta sẽ xem xét chi tiết về những vấn đề liên quan đến ĐS2GT và ứng dụng trong việc xây dựng các mô hình giải bài toán phân lớp dựa trên hệ luật mờ.

2.3.2 Hệ khoảng tương tự trong $\mathcal{AX}^2$

Trong ĐS2GT, chúng ta xét tính kề nhau của các hạng từ. Theo định nghĩa về khoảng tính mờ (*Định nghĩa 1.8*), hai khoảng tính mờ $\mathcal{I}(x)$ và $\mathcal{I}(y)$ được gọi là kề nhau nếu chúng có một điểm mút chung, tức là $lmp(\mathcal{I}(x)) = rmp(\mathcal{I}(y))$ hoặc $rmp(\mathcal{I}(x)) = lmp(\mathcal{I}(y))$, trong đó *lmp* và *rmp* là điểm mút trái và mút phải của khoảng tính mờ. Ta ký hiệu $\mathcal{I}(x) |< \mathcal{I}(y)$ có nghĩa khoảng tính mờ $\mathcal{I}(x)$ kề bên trái $\mathcal{I}(y)$, tức là $rmp(\mathcal{I}(x)) = lmp(\mathcal{I}(y))$. Để ý rằng chúng ta xét các khoảng tính mờ ở dạng nửa đóng.

Ký hiệu $\boldsymbol{X}_k = \{x_1 < x_2 < ... < x_i < ...\}$ và tương ứng tập $\boldsymbol{I}_k = \{\mathcal{I}_k(x_1) \le \mathcal{I}_k(x_2) \le ... \le \mathcal{I}_k(x_i) \le ...\}$ với mỗi $k = 1, 2,...$ Bây giờ chúng ta định nghĩa khoảng tính mờ tượng tự của các hạng từ như sau.

Định nghĩa 2.4. Cho $\mathcal{AX}^2$, $\forall x \in \boldsymbol{X}$ ($k = len(x)$ là độ dài hạng từ x, được hiểu bao gồm phần tử sinh và các gia tử), khoảng tương tự bậc g ($g \ge 1$) của x là khoảng được tạo bởi hai khoảng tính mờ kề nhau trong cùng phân hoạch $\boldsymbol{I}_m$ với $m = k+g$ chứa $\upsilon(x)$ làm điểm trong, ký hiệu $\mathcal{I}^g(x)$, được xác định như sau:

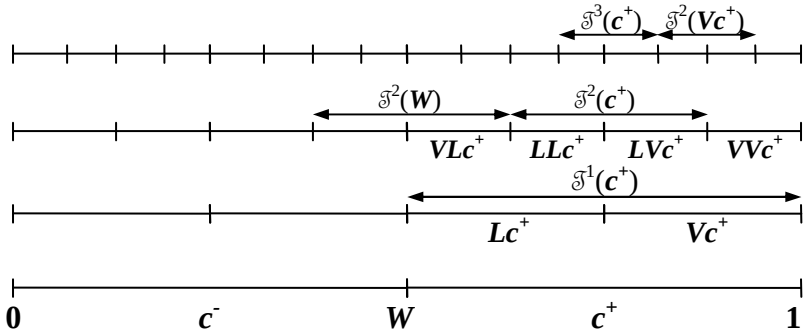
(1) Nếu $x = \mathbf{0}$ ($\upsilon(x) = 0$) thì $\mathcal{I}^g(x) = \mathcal{I}_m(y_1) \in \boldsymbol{I}_m$ và y_1 là hạng từ có ngữ nghĩa bé nhất trong $\boldsymbol{X}_m$ hay $0 \in \mathcal{I}_m(y_1)$;

(2) Nếu $x = \mathbf{1}$ ($\upsilon(x) = 1$) thì $\mathcal{I}^g(x) = \mathcal{I}_m(y_{2m}) \in \boldsymbol{I}_m$ và y_{2m} là hạng từ có ngữ nghĩa lớn nhất trong $\boldsymbol{X}_m$ hay $1 \in \mathcal{I}_m(y_{2m})$;

(3) Nếu $x \ne \mathbf{0}$ và $x \ne \mathbf{1}$ ($0 < \upsilon(x) < 1$) thì $\mathcal{I}^g(x) = \mathcal{I}_m(y_i) \oplus \mathcal{I}_m(y_{i+1})$, với $y_i, y_{i+1} \in \boldsymbol{X}_m$ và $\upsilon(x) = rmp(\mathcal{I}_m(y_i)) = lmp(\mathcal{I}_m(y_{i+1}))$, hay $\mathcal{I}_m(y_i) |< \mathcal{I}_m(y_{i+1})$,

trong đó $\oplus$ là phép nối hai khoảng tính mờ.

Định nghĩa này cho thấy một hạng từ có khoảng tượng tự bậc càng cao sẽ càng bị thu hẹp về tâm (giá trị v) của hạng từ đó: ta có $\forall x \in \boldsymbol{X}$, $g1 > g2$, $\mathcal{I}^{g1}(x) \subset \mathcal{I}^{g2}(x)$. Hình 2.4 minh họa rõ khoảng tượng tự bậc g trong định nghĩa, trong đó $\mathcal{I}^3(\boldsymbol{c}^+) \subset \mathcal{I}^2(\boldsymbol{c}^+) \subset \mathcal{I}^1(\boldsymbol{c}^+)$.

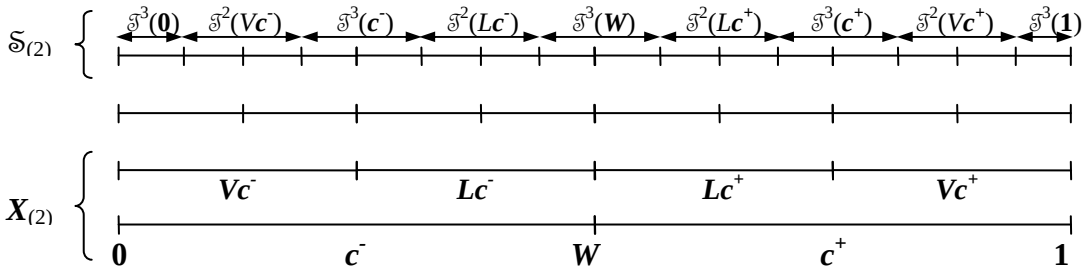


Hình 2.4: Các khoảng tính mờ tượng tự của các hạng từ

Dựa trên khái niệm khoảng tương tự của các hạng từ trong định nghĩa trên, tiếp theo chúng ta định nghĩa tập các khoảng tương tự của tập các hạng từ có độ dài không quá k và được gọi là hệ khoảng tương tự.

Định nghĩa 2.5. Cho $\mathcal{AX}^2$, một số $k \geq 1$ nguyên, hệ khoảng tương tự của tập $\boldsymbol{X}_{(k)}$, ký hiệu $\mathfrak{S}_{(k)}$, là tập các khoảng tương tự của tất cả các hạng từ trong $\boldsymbol{X}_{(k)}$, sao cho $\forall x \in \boldsymbol{X}_{(k)}$, $\mathcal{I}^g(x) \in \mathfrak{S}_{(k)}$, $g+len(x) = k^*$ không đổi (tức là $\forall x \in \boldsymbol{X}_{(k)}$, $\mathcal{I}^g(x)$ được tạo bởi các khoảng tính mờ cùng mức phân hoạch $\boldsymbol{I}_{k^*}$) và $\mathfrak{S}_{(k)}$ là một phân hoạch của [0,1].

Ta gọi $\mathfrak{S}_{(k)}$ là hệ khoảng tương tự mức k, tương ứng với tập $\boldsymbol{X}_{(k)}$. Hình 2.5 sau đây minh họa cho Định nghĩa 2.5 với k=2.



Hình 2.5: Hệ khoảng tương tự $\mathfrak{S}_{(2)}$ của tập $\boldsymbol{X}_{(2)}$

Bổ đề 2.1. Cho $\mathcal{AX}^2$, một số $k \geq 1$ nguyên, $\boldsymbol{X}_{(k)} = \{x_0 < ... < x_i < ... \, x_{2k+1}\}$, $\boldsymbol{X}_{k+1} = \{y_1 < y_2 < ... < y_i < ...\}$. Với $\forall y_i, y_{i+1} \in \boldsymbol{X}_{k+1}$, $\exists x_j \in \boldsymbol{X}_{(k)}$: $y_i < x_j < y_{i+1}$, $i = 1,2,...$ Khi đó, tập $\boldsymbol{X}_{(k+1)} = \boldsymbol{X}_{(k)} \cup \boldsymbol{X}_{k+1} = \{ x_0 < y_1 < x_1 < ... < x_{i-1} < y_i < x_i < ... < y_{2k+1} < x_{2k+1}\}$.

Chứng minh. Theo Mệnh đề 2.5, $|\boldsymbol{X}_{(k)}| = 1+2^{k+1}$, $|\boldsymbol{X}_{k+1}| = 2^{k+1}$, điều này thỏa mãn mỗi cặp theo thứ tự trong $\boldsymbol{X}_{(k)}$ tương ứng với một phần tử trong $\boldsymbol{X}_{k+1}$. Bây giờ ta chứng minh $\forall y_i \in \boldsymbol{X}_{k+1}$, $x_{i-1}, x_i \in \boldsymbol{X}_{(k)}$, $i = 1, 2, ..., 2^{k+1}$: $x_{i-1} < y_i < x_i$. Bằng phương pháp quy nạp, với trường hợp k=1, $\boldsymbol{X}_{(1)} = \{\mathbf{0}, \boldsymbol{c}^-, \boldsymbol{W}, \boldsymbol{c}^+, \mathbf{1}\}$ và $\boldsymbol{X}_2 = \{V\boldsymbol{c}^-, L\boldsymbol{c}^-, L\boldsymbol{c}^+, V\boldsymbol{c}^+\}$, rõ ràng $\boldsymbol{X}_{(2)} = \{\mathbf{0} < V\boldsymbol{c}^- < \boldsymbol{c}^- < L\boldsymbol{c}^- < \boldsymbol{W} < L\boldsymbol{c}^+ < \boldsymbol{c}^+ < V\boldsymbol{c}^+ < \mathbf{1}\}$, vậy $k = 1$ được khẳng định.

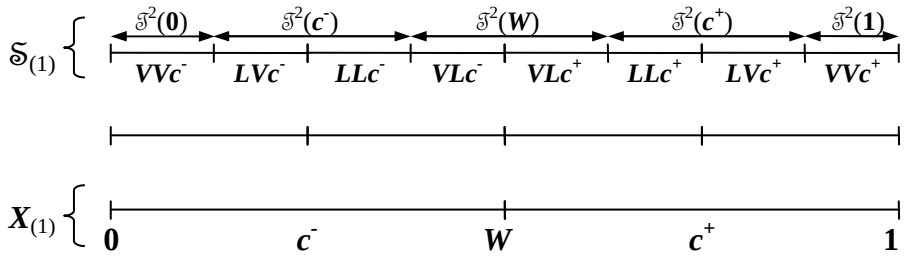
Giả sử trường hợp $k > 1$ đã khẳng định, ta chứng minh cho trường hợp k+1. Theo giả thiết, $\boldsymbol{X}_{(k)} = \{x_0 < ... < x_i < ... \, x_{2k+1}\}$, $\boldsymbol{X}_{k+1} = \{y_1 < y_2 < ... < y_i < ... < y_{2k+1}\}$ và $\boldsymbol{X}_{(k+1)} = \boldsymbol{X}_{(k)} \cup \boldsymbol{X}_{k+1} = \{x_0 < y_1 < x_1 < ... < x_{i-1} < y_i < x_i < ... < y_{2k+1} < x_{2k+1}\}$ (*). Trong $\mathcal{AX}^2$, tính $\boldsymbol{X}_{k+2} = \{Ly_i, Vy_i : \forall y_i \in \boldsymbol{X}_{k+1}\}$ (bằng cách dùng hai gia tử $\{L,V\}$ tác động lên mỗi hạng từ trong $\boldsymbol{X}_{k+1}$). Dựa trên các tính chất của ĐSGT [34], [35], $\forall y_i \in \boldsymbol{X}_{k+1}$, ta có hoặc $\boldsymbol{\Phi} y_i < Ly_i < y_i < Vy_i < \boldsymbol{\Sigma} y_i$, hoặc $\boldsymbol{\Phi} y_i < Vy_i < y_i < Ly_i < \boldsymbol{\Sigma} y_i$, hơn nữa trong $\boldsymbol{X}_{(k+1)}$ ta cũng có $x_{i-1} < \boldsymbol{\Phi} y_i < y_i < \boldsymbol{\Sigma} y_i < x_i$, để ý rằng $x_0 = \mathbf{0}$, $x_{2k+1} = \mathbf{1}$ (**). Từ (*) và (**) ta suy ra $\forall z, z' \in \boldsymbol{X}_{k+2}$, $\exists x_{i-1}, y_i, x_i \in \boldsymbol{X}_{(k+1)}$: $x_{i-1} < z < y_i < z' < x_i$, trong đó $z = Ly_i$, $z' = Vy_i$, hoặc $z = Vy_i$, $z' = Ly_i$, $i = 1,2, ..., 2^{k+1}$. Vậy k+1 được khẳng định $\Rightarrow$ *đpcm*. ■

Tiếp theo chúng ta sẽ xây dựng hệ các khoảng tương tự của các hạng từ trong ĐS2GT và khẳng định sự tồn tại của hệ này bằng định lý sau.

Định lý 2.1. Cho $\mathcal{AX}^2$, một số nguyên $k \geq 1$, với tập $\boldsymbol{X}_{(k)}$, tồn tại duy nhất một hệ các khoảng tương tự mức k, $\mathfrak{S}_{(k)}$ được tạo bởi phân hoạch các khoảng tính mờ mức k+2, tức là $\boldsymbol{I}_{k+2}$.

Chứng minh. Rõ ràng tập $\boldsymbol{I}_{k+2}$ là một phân hoạch của [0,1] (Mệnh đề 1.3), vậy nếu $\mathfrak{S}_{(k)}$ được tạo từ $\boldsymbol{I}_{k+2}$ thỏa mãn là một phân hoạch của [0,1] (thỏa Định nghĩa 2.5). Ta chỉ cần chứng minh $\mathfrak{S}_{(k)}$ được tạo duy nhất từ phân hoạch $\boldsymbol{I}_{k+2}$.

(*i*) Trước hết, chứng minh $\mathfrak{S}_{(k)}$ được tạo từ phân hoạch $\boldsymbol{I}_{k+2}$. Bằng phương pháp quy nạp, với trường hợp k=1, ta có $\boldsymbol{X}_{(1)} = \{\mathbf{0}, \boldsymbol{c}^-, \boldsymbol{W}, \boldsymbol{c}^+, \mathbf{1}\}$ và $\boldsymbol{I}_3 = \{\mathfrak{I}(VV\boldsymbol{c}^-), \mathfrak{I}(LV\boldsymbol{c}^-), \mathfrak{I}(LL\boldsymbol{c}^-), \mathfrak{I}(VL\boldsymbol{c}^-), \mathfrak{I}(VL\boldsymbol{c}^+), \mathfrak{I}(LL\boldsymbol{c}^+), \mathfrak{I}(LV\boldsymbol{c}^+), \mathfrak{I}(VV\boldsymbol{c}^+)\}$. Theo Định nghĩa 2.4, hệ khoảng tương tự của $\boldsymbol{X}_{(1)}$ được xây dựng như sau, $\mathfrak{S}_{(1)} = \{\mathfrak{I}^2(\mathbf{0}) = \mathfrak{I}(VV\boldsymbol{c}^-), \mathfrak{I}^2(\boldsymbol{c}^-) = \mathfrak{I}(LV\boldsymbol{c}^-) \oplus \mathfrak{I}(LL\boldsymbol{c}^-), \mathfrak{I}^2(\boldsymbol{W}) = \mathfrak{I}(VL\boldsymbol{c}^-) \oplus \mathfrak{I}(VL\boldsymbol{c}^+), \mathfrak{I}^2(\boldsymbol{c}^+) = \mathfrak{I}(LL\boldsymbol{c}^+) \oplus \mathfrak{I}(LV\boldsymbol{c}^+), \mathfrak{I}^2(\mathbf{1}) = \mathfrak{I}(VV\boldsymbol{c}^+)\}$ (Hình vẽ 2.6), vậy k=1 được khẳng định.



Hình 2.6: Hệ khoảng tương tự $\mathfrak{S}_{(1)}$ của $\boldsymbol{X}_{(1)}$

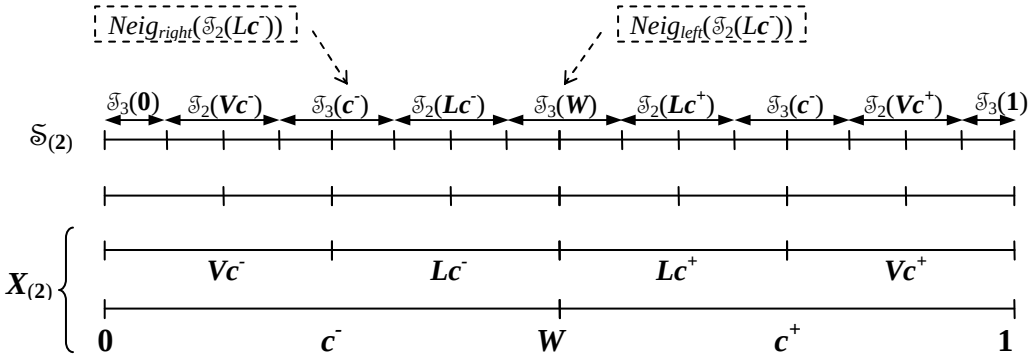
Giả sử trường hợp $k > 1$ đã khẳng định, tức là $\mathfrak{S}_{(k)}$ của $\boldsymbol{X}_{(k)}$ được tạo bởi $\boldsymbol{I}_{k+2}$. Ta chứng minh cho trường hợp k+1. Để ý rằng $\boldsymbol{I}_k = \{\mathfrak{I}(y_i) : \forall y_i \in \boldsymbol{X}_k\}$. Không mất tính tổng quát, theo tính chất của ĐSGT, giả sử các tập $\boldsymbol{X}_k$, $\boldsymbol{X}_{(k)}$ và $\boldsymbol{X}_{k+2}$ được sắp thứ tự. Hệ khoảng tương tự của $\boldsymbol{X}_{(k)}$ được xây dựng như sau, $\mathfrak{S}_{(k)} = \{\mathfrak{I}(w_i) = \mathfrak{I}(y_{2i}) \oplus \mathfrak{I}(y_{2i+1}) : w_i \in \boldsymbol{X}_{(k)}, y_{2i}, y_{2i+1} \in \boldsymbol{X}_{k+2}, i=1,2, ...,2^{k+1}\text{-}1\} \cup \{\mathfrak{I}(y_1), \mathfrak{I}(y_{2^{k+2}})\}$, điều này thỏa mãn vì theo Mệnh đề 2.5 ta có $|\mathfrak{S}_{(k)}| = |\boldsymbol{X}_{(k)}| = 1+2^{k+1}$ và $|\boldsymbol{X}_{k+2}| = 2^{k+2} \Rightarrow (|\boldsymbol{X}_{k+2}|\text{-}2)/2 = 2^{k+1}\text{-}1 = |\mathfrak{S}_{(k)}|\text{-}2$, để ý rằng trong $\mathfrak{S}_{(k)}$, $\mathfrak{I}(\mathbf{0}) = \mathfrak{I}(y_1)$, $\mathfrak{I}(\mathbf{1}) = \mathfrak{I}(y_{2^{k+2}})$. Trong ĐS2GT, sinh các tập $\boldsymbol{X}_{k+1} = \{Lw_i, Vw_i : \forall w_i \in \boldsymbol{X}_k\}$, $\boldsymbol{X}_{k+3} = \{Ly_i, Vy_i : \forall y_i \in \boldsymbol{X}_{k+2}\}$ (bằng cách dùng hai gia tử $\{L,V\}$ tác động lên mỗi hạng từ trong $\boldsymbol{X}_k$, $\boldsymbol{X}_{k+2}$), tương ứng ta có $\boldsymbol{I}_{k+3}$. Biết rằng $\boldsymbol{X}_{(k+1)} = \boldsymbol{X}_{(k)} \cup \boldsymbol{X}_{k+1}$, $\boldsymbol{X}_{(k+2)} = \boldsymbol{X}_{(k+1)} \cup \boldsymbol{X}_{k+2}$, $\boldsymbol{X}_{(k+3)} = \boldsymbol{X}_{(k+2)} \cup \boldsymbol{X}_{k+3}$. Theo Bổ đề 2.1, từ $\boldsymbol{X}_{(k)}$ và $\boldsymbol{X}_{k+1}$ ta có thứ tự tập $\boldsymbol{X}_{(k+1)} = \{ ... < w_{i-1} < x_i < w_i < x_{i+1} < w_{i+1} < ... \}$, trong đó $w_{i-1}, w_i, w_{i+1} \in \boldsymbol{X}_{(k)}$, $x_i, x_{i+1} \in \boldsymbol{X}_{k+1}$, i = 1,2, ..., 2^{k+1}-1. Tương tự, với $x_{j-1}, x_j, x_{j+1} \in \boldsymbol{X}_{(k+1)}$, $y_j, y_{j+1} \in \boldsymbol{X}_{k+2}$, j = 1,2, ..., 2^{k+2}-1, ta có thứ tự tập $\boldsymbol{X}_{(k+2)} = \{ ... < x_{j-1} < y_j < x_j < y_{j+1} < x_{j+1} < ... \}$, và $z_{2j-1}, z_{2j}, z_{2j+1}, z_{2(j+1)} \in \boldsymbol{X}_{k+3}$, suy ra thứ tự tập $\boldsymbol{X}_{(k+3)} = \{ ... < x_{j-1} < z_{2j-1} < y_j < z_{2j} < x_j < z_{2j+1} < y_{j+1} < z_{2(j+1)} < x_{j+1} < ... \}$. Xây dựng tập $\mathfrak{S}_{(k+1)} = \{\mathfrak{I}(x_j) =$

$\mathfrak{I}(z_{2j}) \oplus \mathfrak{I}(z_{2j+1}) : x_j \in \boldsymbol{X}_{(k+1)}, z_{2j}, z_{2j+1} \in \boldsymbol{X}_{k+3}, j = 1,2, ..., 2^{k+2}\text{-}1\} \cup \{\mathfrak{I}(z_1), \mathfrak{I}(z_{2^{k+3}})\}$, để ý rằng trong $\mathfrak{S}_{(k+1)}$ có $\mathfrak{I}(\mathbf{0}) = \mathfrak{I}(z_1)$, $\mathfrak{I}(\mathbf{1}) = \mathfrak{I}(z_{2^{k+3}})$. Vậy trường hợp k+1 được khẳng định.

(*ii*) Chứng minh không tồn tại $m \geq 1$, $m \neq 2$, để $\mathfrak{S}_{(k)}$ được tạo từ $\boldsymbol{I}_{k+m}$. Theo Mệnh đề 2.1, $|\boldsymbol{I}_{k+m}| = 2^{k+m}$, $|\mathfrak{S}_{(k)}| = |\boldsymbol{X}_{(k)}| = 1+2^{k+1}$, mặt khác, theo Định nghĩa 2.5 ta phải có $2(|\mathfrak{S}_{(k)}|\text{-}2)+2 = |\boldsymbol{I}_{k+m}|$, hay $2(1+2^{k+1}\text{-}2)+2 = 2^{k+m} \Leftrightarrow 2^{k+2} = 2^{k+m} \Leftrightarrow m = 2$.

Từ (*i*) và (*ii*) $\Rightarrow$ *đpcm*. ■

Theo Định nghĩa 2.5, với $k \geq 1$, $\forall x \in \boldsymbol{X}_{(k)}$, $\mathfrak{I}^g(x) \in \mathfrak{S}_{(k)}$, ta xác định hai khoảng tương tự trong $\mathfrak{S}_{(k)}$ kề bên trái là $Neig_{left}(\mathfrak{I}^g(x)) = \mathfrak{I}^{g1}(y)$, kề bên phải là $Neig_{right}(\mathfrak{I}^g(x)) = \mathfrak{I}^{g2}(z)$ với $y,z \in \boldsymbol{X}_{(k)}$ sao cho $y < x < z$ và không tồn tại hạng từ $z \in \boldsymbol{X}_{(k)}$ nằm giữa chúng, tức là $lmp(\mathfrak{I}^g(x)) = rmp(\mathfrak{I}^{g1}(y))$ và $rmp(\mathfrak{I}^g(x)) = lmp(\mathfrak{I}^{g2}(z))$. Trường hợp $x = \mathbf{0}$, ta xác định $Neig_{left}(\mathfrak{I}^g(\mathbf{0})) = \mathfrak{I}(\mathbf{0})$ là khoảng tính mờ bé nhất trong $\boldsymbol{I}_{k+2}$, tương tự khi $x = \mathbf{1}$, $Neig_{right}(\mathfrak{I}^g(\mathbf{1})) = \mathfrak{I}(\mathbf{1})$ là khoảng tính mờ lớn nhất trong $\boldsymbol{I}_{k+2}$ (Hình 2.7).



Hình 2.7: Hệ phân hoạch các khoảng tương tự và láng giềng của chúng

Hệ quả 2.1. Cho $\mathcal{A}x^2$, một số $k \geq 1$ nguyên, $\forall v \in [0,1]$, luôn tồn tại duy nhất một hạng từ $x \in \boldsymbol{X}_{(k)}$ xác định khoảng tương tự bậc $g = 2+k\text{-}len(x)$, $\mathfrak{I}^g(x) \in \mathfrak{S}_{(k)}$ và $v \in \mathfrak{I}^g(x)$.

Chứng minh. Dễ dàng suy ra từ tính phân hoạch của $\mathfrak{S}_{(k)}$ trong đoạn [0,1] (Định lý 2.1). ■

Hệ quả này cho thấy tính ứng dụng của hệ khoảng tương tự $\mathfrak{S}_{(k)}$ của $\mathcal{AX}^2$ trong các quá trình thực.

Bổ đề 2.2. Cho $\mathcal{AX}^2$, với $u,v \in [0,1]$, $u \neq v$ và $0 < \varepsilon < |u\text{-}v|/6$, ta có $k = 1+\lceil log_{\lambda}(\varepsilon/\gamma)\rceil$, trong đó $\lambda = \max\{\mu(L), \mu(V)\}$, $\gamma = \max\{fm(\boldsymbol{c}^-), fm(\boldsymbol{c}^+)\}$, để bộ ba giá trị ngôn ngữ $x < y < z \in \boldsymbol{X}_k$ thỏa $\upsilon(z) - \upsilon(x) < |u\text{-}v|$.

Chứng minh. Theo Mệnh đề 2.4 trong [8], $\forall r \in [0,1]$ và $\varepsilon > 0$ bé tùy ý, đều tồn tại giá trị ngôn ngữ $x \in \boldsymbol{X}_k$ với $k = 1+\lceil log_{\lambda}(\varepsilon/\gamma)\rceil$ thỏa $|\upsilon(x) - r| \leq \varepsilon$.

Không mất tính tổng quát, giả sử $u < v$. Ta đặt $\eta = (v\text{-}u)/6$, $r_1 = u+\eta$, $r_2 = u+3\eta$ và $r_3 = u+5\eta$ (tức là r_1, r_2 và r_3 là 3 điểm chia trong đoạn $[u, v]$ thành 4 phần theo tỷ lệ 1:2:2:1), vậy có $x, y, z \in \boldsymbol{X}_k$ để $|\upsilon(x) - r_1| \leq \varepsilon$, $|\upsilon(y) - r_2| \leq \varepsilon$, $|\upsilon(z) - r_3| \leq \varepsilon$. Vì $r_1 < r_2 < r_3$ và $\varepsilon < \eta$ nên $\upsilon(x) < \upsilon(y) < \upsilon(z) \Rightarrow x < y < z$.

Hơn nữa $r_3 = r_1+4\eta$, suy ra $\upsilon(z) - \upsilon(x) \leq r_3 - r_1 + 2\varepsilon = 4\eta + 2\varepsilon \Rightarrow \upsilon(z) - \upsilon(x) < 4\eta + 2\eta = 6\eta$ (để ý $\varepsilon < \eta$). Mặt khác, $v\text{-}u = 6\eta$ nên ta có $\upsilon(z) - \upsilon(x) < v\text{-}u$, hay $\upsilon(z) - \upsilon(x) < |u\text{-}v| \Rightarrow đpcm$.■

Bổ đề này cho thấy chúng ta luôn xác định được bộ ba giá trị ngôn ngữ khác nhau $x, y, z \in \boldsymbol{X}_k$ với k được xác định như trên, sao cho giá trị định lượng ngữ nghĩa của chúng nằm giữa cặp $u, v \in [0,1]$ bất kỳ. Ở đây giá trị ε đóng vai trò độ chính xác khi chúng ta xấp xỉ các điểm chia r_1, r_2 và r_3 bằng các giá trị ngôn ngữ x, y và z tương ứng. Rõ ràng khi ε càng bé thì mức phân hoạch k ($\boldsymbol{X}_k$) càng lớn, và nó được giới hạn trên bởi 1/6 vì $|u\text{-}v| \leq 1$.

Trong ĐS2GT, ta có $0.5 \leq \lambda,\gamma < 1$ vì $\mu(V) = 1\text{-}\mu(L)$, $fm(\boldsymbol{c}^+) = 1\text{-}\ fm(\boldsymbol{c}^-)$, nên mức phân hoạch k xác định trong bổ đề trên được giới hạn dưới bằng $1+\lceil log_{0.5}((1/6)/0.5)\rceil = 3$, tức là $\lambda = \gamma = 0.5$ và $\varepsilon = 1/6$.

Định lý 2.2. Cho $\mathcal{AX}^2$, với $u,v \in [0,1]$, $u \neq v$, và $0 < \varepsilon < |u\text{-}v|/6$, ta có $k = 1+\lceil log_\lambda(\varepsilon/\gamma)\rceil$ để hệ khoảng tương tự mức $k^* = (k\text{-}2)$, tức là $\mathfrak{S}_{(k^*)}$, tương ứng với tập $\boldsymbol{X}_{(k^*)}$ sao cho $u \in \mathcal{I}(x)$, $v \in \mathcal{I}(y)$, $\mathcal{I}(x), \mathcal{I}(y) \in \mathfrak{S}_{(k^*)}$ (hay $x,y \in \boldsymbol{X}_{(k^*)}$) và $x \neq y$.

Chứng minh. Theo Định lý 2.1, hệ khoảng tương tự mức k ($\mathfrak{S}_{(k)}$) duy nhất được tạo bởi hệ khoảng tính mờ mức k+2 ($\boldsymbol{I}_{k+2}$). Ở đây hệ $\mathfrak{S}_{(k^*)}$ được tạo bởi hệ khoảng tính mờ $\boldsymbol{I}_k$ với $k = 1+\lceil log_\lambda(\varepsilon/\gamma)\rceil$ và $k^* = k\text{-}2$.

Theo Định nghĩa 2.5 về hệ khoảng tương tự, $\forall x \in \boldsymbol{X}_{(k^*)}$, $\mathcal{I}(x) \in \mathfrak{S}_{(k^*)}$, $\upsilon(x) \in \mathcal{I}(x) = \mathcal{I}(y) \oplus \mathcal{I}(z)$, với $\mathcal{I}(y), \mathcal{I}(z) \in \boldsymbol{I}_k$ và chúng kề nhau. Để ý $\mathcal{I}(\mathbf{0}) = \mathcal{I}(w')$ và $\mathcal{I}(\mathbf{1}) = \mathcal{I}(w'')$, với w' và w'' tương ứng là hai giá trị ngôn ngữ bé nhất và lớn nhất trong $\boldsymbol{X}_k$. Từ Bổ đề 2.2, chúng ta luôn xác định $y < z < w \in \boldsymbol{X}_k$ thỏa $\upsilon(w) - \upsilon(y) < |u\text{-}v|$, suy ra $u < \upsilon(y) < \upsilon(z) < \upsilon(w) < v$, giả sử $u < v$ không mất tính tổng quát. Để ý rằng $\upsilon(y) \in \mathcal{I}(y)$, $\upsilon(z) \in \mathcal{I}(z)$, $\upsilon(w) \in \mathcal{I}(w)$ và $\mathcal{I}(y) \leq \mathcal{I}(z) \leq \mathcal{I}(w)$, ta có các trường hợp sau:

(*i*) Trường hợp $\mathcal{I}(y) \subseteq \mathcal{I}(x_1)$, $\mathcal{I}(z) \subseteq \mathcal{I}(x_2)$, $\mathcal{I}(w) \subseteq \mathcal{I}(x_3)$, với $x_1 \neq x_2 \neq x_3 \in \boldsymbol{X}_{(k^*)}$. Suy ra $u \in \mathcal{I}(x_1)$ hoặc $u \in \mathcal{I}(x'_1)$ với $x'_1 < x_1$, và $v \in \mathcal{I}(x_3)$ hoặc $v \in \mathcal{I}(x'_3)$ với $x'_3 > x_3$.

(*ii*) Trường hợp $\mathcal{I}(y) \oplus \mathcal{I}(z) = \mathcal{I}(x_1)$, $\mathcal{I}(w) \subseteq \mathcal{I}(x_2)$, với $x_1 \neq x_2 \in \boldsymbol{X}_{(k^*)}$. Suy ra $u \in \mathcal{I}(x_1)$ hoặc $u \in \mathcal{I}(x'_1)$ với $x'_1 < x_1$, và $v \in \mathcal{I}(x_2)$ hoặc $v \in \mathcal{I}(x'_2)$ với $x'_2 > x_2$.

(*iii*) Trường hợp $\mathcal{I}(y) \subseteq \mathcal{I}(x_1)$, $\mathcal{I}(z) \oplus \mathcal{I}(w) = \mathcal{I}(x_2)$, với $x_1 \neq x_2 \in \boldsymbol{X}_{(k^*)}$. Suy ra $u \in \mathcal{I}(x_1)$ hoặc $u \in \mathcal{I}(x'_1)$ với $x'_1 < x_1$, và $v \in \mathcal{I}(x_2)$ hoặc $v \in \mathcal{I}(x'_2)$ với $x'_2 > x_2$.

Từ (*i*), (*ii*) và (*iii*) $\Rightarrow$ *đpcm*.■

Định lý này cho thấy khả năng phân hoạch mọi điểm trong [0,1] của hệ khoảng tương tự trong $\mathcal{AX}^2$.

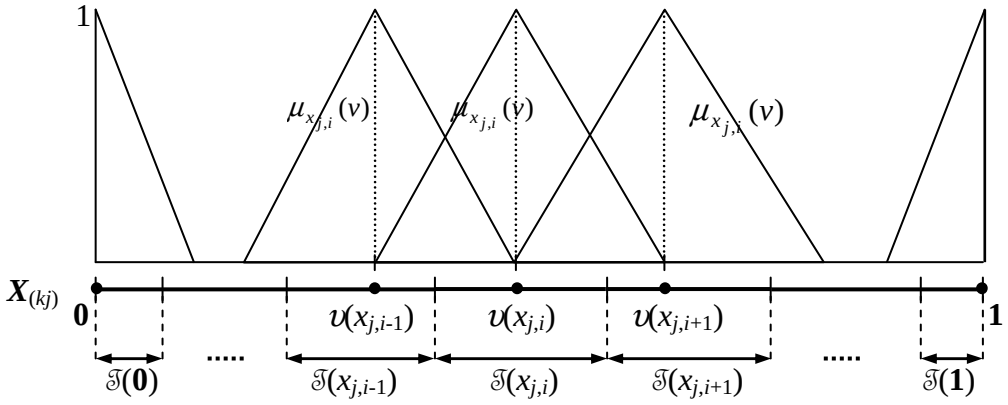
Hệ quả 2.2. Cho $\mathcal{AX}^2$, với một tập con $E \subset [0,1]^1$, luôn tồn tại hệ phân hoạch tương tự mức k - $\mathfrak{S}_{(k)}$ tương ứng tập giá trị ngôn ngữ $\boldsymbol{X}_{(k)}$ sao cho $\forall u,v \in E$, $u \neq v$, $\nexists x,y \in \boldsymbol{X}_{(k)}$, $x \neq y$ xác định $\mathcal{I}(x), \mathcal{I}(y) \in \mathfrak{S}_{(k)}$ và $u \in \mathcal{I}(x)$, $v \in \mathcal{I}(y)$.

Chứng minh. Dễ dàng suy ra từ Định lý 2.2 bằng cách đặt $k = max\{ k_i : k_i$ được xác định bằng k^* trong Định lý 2.2 cho mọi cặp $u,v \in E, u \neq v$, với ε_i được chọn sao cho $0 < \varepsilon_i < |u\text{-}v|/6\}$.■

Từ hệ quả trên chúng ta thấy rằng với bất kỳ miền thực nào, nếu lấy mẫu tạo thành một tập hữu hạn *E* tùy ý, luôn xây dựng được một hệ phân hoạch các khoảng tượng tự trong ĐS2GT để phân hoạch và xấp xỉ tập *E* đó. Tức là, mỗi mẫu trong *E* sẽ xác định một hạng từ duy nhất trong phân hoạch tương ứng với khoảng tương tự chứa nó. Điều này đem lại khả năng ứng dụng rất lớn của ĐS2GT trong việc xây dựng các mô hình giải bài toán phân lớp, sẽ được giới thiệu ở phần sau.

2.3.3 Thuật toán sinh luật mờ dựa trên hệ khoảng tương tự

Trong ĐS2GT, dựa trên hệ khoảng tương tự của tập $\boldsymbol{X}_{(kj)}$, chúng ta áp dụng lưới phân hoạch để sinh hệ luật mờ theo lược đồ đã trình bày trong Mục 2.1. Trước hết, mỗi hạng từ trong tập $\boldsymbol{X}_{(kj)} = \{ x_{j,0}, x_{j,1}, ..., x_{j,i-1}, x_{j,i}, x_{j,i+1}, ..., x_{j,1+2kj+1}\}$ được thiết kế hàm định lượng ngữ nghĩa theo dạng tam giác $\mu_{(x_{j,i})}(v)$ (công thức 2.6) sao cho giá trị hàm càng gần tâm $\upsilon(x_{j,i})$ thì càng cao và bằng 1 tại tâm, nó sẽ bằng 0 nếu vượt ra ngoài tâm của hai hạng từ láng giềng của $x_{j,i}$ trong tập $\boldsymbol{X}_{(kj)}$ (Hình 2.8):



Hình 2.8: Hàm định lượng dạng tam giác của các hạng từ trong ĐS2GT

$$\mu_{x_{j,i}}(v) = \min\left(\max\left(\frac{v - \upsilon(x_{j,i-1})}{\upsilon(x_{j,i}) - \upsilon(x_{j,i-1})}, 0 \right), \max\left(\frac{\upsilon(x_{j,i+1}) - v}{\upsilon(x_{j,i+1}) - \upsilon(x_{j,i})}, 0 \right) \right) \qquad (2.6)$$

Bây giờ chúng ta thiết kế thuật toán sinh hệ luật khởi đầu sử dụng phân hoạch dựa trên hệ khoảng tương tự trong ĐS2GT. Có hai điểm khác biệt so với thuật toán ***IFRG*1**. Thứ nhất, thay vì sử dụng phân hoạch các khoảng tính mờ tương ứng với ngữ nghĩa của các hạng từ độ dài k_j ở đây chúng ta sẽ xem xét ngữ nghĩa của tất cả các hạng từ độ dài 1 đến k_j ($\boldsymbol{X}_{(kj)}$) sử dụng hệ phân hoạch các khoảng tương tự $\mathfrak{S}_{(kj)}$ của chúng trong ĐS2GT. Do đó ngữ nghĩa của các hạng từ trong ĐSGT đều được áp dụng cho việc sinh luật.

Thứ hai, thuật toán này không chỉ sinh luật với độ dài cố định đúng bằng số thuộc tính mà xem xét hết các khả năng sinh luật theo độ dài của vế trái từ 1 đến giới hạn L bằng cách lấy tổ hợp các điều kiện trong vế trái của luật sinh từ các siêu hộp (đã trình bày ở phần trên). Thuật toán này sẽ chịu ảnh hưởng của sự bùng nổ tổ hợp khi số thuộc tính lớn, do đó hệ luật sinh ra có thể chứa một số lượng rất lớn các luật. Hơn nữa, hệ luật sinh ra gồm các luật có độ dài không giống nhau. Thuật toán sẽ khắc phục được vấn đề trích chọn đặc trưng từ tập dữ liệu mẫu và hạn chế dư thừa các điều kiện trong vế trái của luật.

Mức phân hoạch k_j có thể xác định bằng hệ quả 2.2 đối với một tập dữ liệu cho trước. Tuy nhiên, hệ luật mờ được xây dựng không chỉ đóng vai trò xấp xỉ tập dữ liệu, tức tính cá thể cao, mà còn phải có tính phổ quát nhằm giúp cho việc dự báo đối với các dữ liệu mới. Do vậy, chúng ta phải cân bằng giữa tính cá thể và tính phổ quát dựa trên mức phân hoạch k_j và chúng ta thiết lập tham số này cũng là đầu vào của thuật toán.

Thuật toán 2.4. Sinh các luật mờ từ tập dữ liệu mẫu dựa trên phân hoạch các khoảng tương tự trong ĐS2GT (***IFRG*2**).

Inputs:

- Tập dữ liệu mẫu $\boldsymbol{D} = \{ (p_i; c_i) \mid i=1, ..., N \}$, $p_i = (d_{i,1}, ..., d_{i,n}) \in \boldsymbol{P}$, $c_i \in \boldsymbol{C} = \{C_1, ..., C_m\}$, n là số thuộc tính, m là số lớp, N là số mẫu dữ liệu;

- Bộ các tham số mờ gia tử của ĐS2GT cho mỗi thuộc tính $\boldsymbol{PAR}_j = \{ fm_j(\boldsymbol{c}^-), fm_j(\boldsymbol{c}^+), \mu_j(h) \ \forall h \in \boldsymbol{H} \}$, $j = 1, ..., n$;

- Mức phân hoạch k_j hệ các khoảng tương tự trên miền của các thuộc tính;

- Giới hạn độ dài tối đa của luật L;

Outputs: Tập các luật mờ $\boldsymbol{S}_0 = \{R_1, ..., R_M\}$

Actions:

(*Step*1) Khởi tạo tập luật $\boldsymbol{S}_0 = \varnothing$,

(*Step*2) Tính phân hoạch hệ các khoảng tương tự,

$$\mathfrak{S}_{(kj)} = \{ \mathfrak{I}(x_{kj,1}), \mathfrak{I}(x_{kj,2}), ... \} \text{ và tương ứng}$$

$$\boldsymbol{X}_{(kj)} = \{x_{kj,1}, x_{kj,2}, ... \}, j = 1, 2, ..., n.$$

(*Step*3) Lặp trên mỗi mẫu dữ liệu $(p_i; c_i) \in \boldsymbol{D}$ và thực hiện:

(*Step*3.*a*) Xác định giá trị ngôn ngữ tương ứng với $d_{i,j} \in p_i$,

$$\{ A_{i,j} = x_{kj,i^*} \mid x_{kj,i^*} \in \boldsymbol{X}_{(kj)} \text{ và } d_{i,j} \in \mathfrak{I}(x_{kj,i^*}),\ j = 1, 2, ..., n \}$$

(*Step*3.*b*) Tạo một tuyển vế trái gồm n giá trị ngôn ngữ trên:

$$\boldsymbol{A}(i) = \{A_{i,1}, A_{i,2}, ..., A_{i,n}\},$$

(*Step*3.*c*) Sinh các tuyển vế trái thứ cấp từ $\boldsymbol{A}(i)$ bằng cách lấy tổ hợp các giá trị ngôn ngữ theo độ dài từ 1 đến giới hạn L,

$$\boldsymbol{C}(i) = \bigcup_{l=1}^{L} \boldsymbol{C}(l,i), \text{ với}$$

$$\boldsymbol{C}(l,i) = \{ (A_{i,j1}, ..., A_{i,jl}) \mid A_{i,jk} \in \boldsymbol{A}(i), j_k \in [1, n], k = 1,...,l \}$$

là các tập l giá trị ngôn ngữ chọn từ $\boldsymbol{A}(i)$, $l = 1, ..., L$.

(*Step*3.*d*) Tính độ hỗ trợ và độ tin cậy của tuyển các luật,

$$c(\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_h) \text{ và } s(\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_h), \text{ với } \forall \boldsymbol{A}_q \in \boldsymbol{C}(i), h = 1,...,m,$$

(*Step*3.*e*) Sinh tập luật mới,

$$\boldsymbol{S}_0 = \boldsymbol{S}_0 \cup \{\boldsymbol{A}_q \Rightarrow \boldsymbol{C}_q \mid \boldsymbol{A}_q \in \boldsymbol{C}(i)\},$$

$$\text{trong đó } \boldsymbol{C}_q = \arg\max_{C_h} \{ c(\boldsymbol{A}_q \Rightarrow C_h) \mid h = 1,...,m \}.$$

Return: Tập $\boldsymbol{S}_0$.

End.□

Mệnh đề 2.6. Độ phức tạp của thuật toán ***IFRG2*** là đa thức bậc 2 đối với kích thước (số mẫu, $|\boldsymbol{D}|$) của tập dữ liệu mẫu và đa thức bậc L đối với số thuộc tính n.

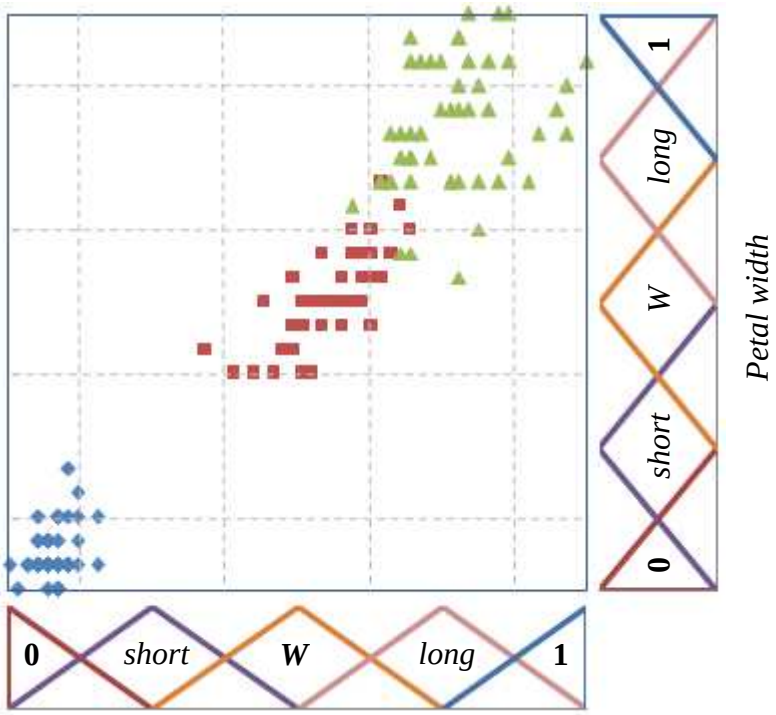
Chứng minh. Dễ dàng nhận thấy thời gian tính toán phân hoạch hệ các khoảng tương tự mức k_j cho các thuộc tính ở bước 2 là $O(n.|\boldsymbol{X}_{(k^*)}|)$, trong đó $k^* = \max\{ k_j : j = 1,...,n \}$. Thông thường chúng ta có $|\boldsymbol{X}_{(k^*)}| << |\boldsymbol{D}|$ (*).

Trong bước 3, thời gian tính toán để sinh các điều kiện vế trái là $O(n.|\boldsymbol{D}|)$. Trường hợp cực đoan kích thước của tập các tuyển điều kiện vế trái là $|\boldsymbol{D}|$, tức mỗi mẫu dữ liệu xác định một điều kiện vế trái. Mỗi điều kiện này được dùng để sinh tiếp các tuyển điều kiện vế trái có độ dài bé hơn, tức từ 1 đến L. Như vậy chúng ta có kích thước của tập tất cả các tuyển điều kiện luật là $|\boldsymbol{D}|.(C_1^n + C_2^n + ... + C_L^n)$ trong trường hợp cực đoan. Bước (3.*d*) cần phải tính độ tin cậy và độ hỗ trợ của các luật có vế trái là các tuyển điều kiện này và phần kết luận là các nhãn phân lớp. Thời gian tính toán của bước này là $O(m.|\boldsymbol{D}|^2.\sum_{l=1}^{L} C_l^n)$ (**). Khi $L << n$ thì biểu thức $\sum_{l=1}^{L} C_l^n$ được giới hạn thời gian bởi $O(n^L)$ (***).

Từ (*), (**) và (***) ta suy ra độ phức tạp của thuật toán ***IFRG2*** được giới hạn bởi $O(n^L.|\boldsymbol{D}|^2)$ (để ý rằng $m << |\boldsymbol{D}|$ và được xem như hằng số) $\Rightarrow$ *đpcm*. ■

Ví dụ 2.4. Minh họa phương pháp sinh luật của thuật toán ***IFRG2*** để sinh hệ luật cho tập dữ liệu mẫu của bài toán *IRIS*2 đã được xem xét trong Ví dụ 2.1.

Các tham số mờ gia tử đặt giống nhau $fm_j(\boldsymbol{c}^-) = fm_j(\boldsymbol{c}^+) = 0.5$, $\mu(L) = \mu(V) = 0.5$, và mức phân hoạch $k_j = 1$ cho cả hai thuộc tính *petal length* (*PL*) và *petal width* (*PW*). Trường hợp số thuộc tính nhỏ (n=2) nên giới hạn độ dài luật đúng bằng số thuộc tính L=2. Tính toán phân hoạch hệ các khoảng tương tự trong ĐS2GT trên miền của hai thuộc tính *PL* và *PW* tạo thành lưới phân hoạch mờ trong Hình 2.9 (hàm định lượng ngữ nghĩa gán cho mỗi giá trị ngôn ngữ được tính theo công thức (2.6)).



Hình 2.9: Lưới phân hoạch mờ dựa trên hệ các khoảng tương tự

Kết quả áp dụng thuật toán ***IFRG2*** gồm hệ f_n = 22 luật với độ dài trung bình f_a = 1.55 trong Bảng 2.4, mỗi luật được tính toán độ tin cậy ***c***, độ hỗ trợ ***s*** và các trọng số CF^1, CF^2, CF^3, CF^4. Ở đây chúng ta sử dụng giá trị ngôn ngữ *completely short* biểu diễn ngữ nghĩa cho hạng từ **0**, *completely long* biểu diễn cho hạng từ **1** và *medium* biểu diễn cho hạng từ ***W*** trong phân hoạch hệ các khoảng tương tự. Hệ luật ở đây có số luật nhiều hơn kết quả của thuật toán ***IFRG1*** trong Bảng 2.1 vì thuật toán này chịu tác động tổ hợp của độ dài luật trên số thuộc tính. Tỷ lệ phân lớp đúng trên tập mẫu theo cả hai phương pháp lập luận (*single winner rule, weighted vote*) với các đánh giá trọng số luật khác nhau thể hiện trong Bảng 2.5.

Bảng 2.4: Danh sách luật sinh bởi thuật toán ***IFRG2*** cho bài toán *IRIS*2

R_1	**if** *PL* is *completely short* **then** *Setosa* (***c*** = 1, ***s*** = 0.228, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_2	**if** *PL* is *short* **then** *Setosa* (***c*** = 0.851, ***s*** = 0.105, CF^1 = 0.851, CF^2 = 0.777, $CF^3 = CF^4$ = 0.703)

R_3	**if** *PW* is *completely short* **then** *Setosa* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.253, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_4	**if** *PW* is *short* **then** *Setosa* ($\boldsymbol{c}$ = 0.692, $\boldsymbol{s}$ = 0.080, $CF^1 = 0.692$, $CF^2 = 0.538$, $CF^3 = CF^4 = 0.385$)
R_5	**if** *PL* is *completely short* and *PW* is *completely short* **then** *Setosa* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.176, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_6	**if** *PL* is *completely short* and *PW* is *short* **then** *Setosa* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.053, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_7	**if** *PL* is *short* and *PW* is *completely short* **then** *Setosa* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.078, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_8	**if** *PW* is *medium* **then** *Versicolor* ($\boldsymbol{c}$ = 0.885, $\boldsymbol{s}$ = 0.248, $CF^1 = 0.885$, $CF^2 = 0.827$, $CF^3 = CF^4 = 0.770$)
R_9	**if** *PL* is *medium* **then** *Versicolor* ($\boldsymbol{c}$ = 0.862, $\boldsymbol{s}$ = 0.227, $CF^1 = 0.862$, $CF^2 = 0.793$, $CF^3 = CF^4 = 0.723$)
R_{10}	**if** *PL* is *long* and *PW* is *medium* **then** *Versicolor* ($\boldsymbol{c}$ = 0.727, $\boldsymbol{s}$ = 0.062, $CF^1 = 0.727$, $CF^2 = 0.590$, $CF^3 = CF^4 = 0.454$)
R_{11}	**if** *PL* is *medium* and *PW* is *medium* **then** *Versicolor* ($\boldsymbol{c}$ = 0.969, $\boldsymbol{s}$ = 0.174, $CF^1 = 0.969$, $CF^2 = 0.953$, $CF^3 = CF^4 = 0.937$)
R_{12}	**if** *PL* is *medium* and *PW* is *short* **then** *Versicolor* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.027, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_{13}	**if** *PL* is *short* and *PW* is *medium* **then** *Versicolor* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.011, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_{14}	**if** *PL* is *long* **then** *Virginica* ($\boldsymbol{c}$ = 0.724, $\boldsymbol{s}$ = 0.232, $CF^1 = 0.724$, $CF^2 = 0.586$, $CF^3 = CF^4 = 0.448$)
R_{15}	**if** *PL* is *completely long* **then** *Virginca* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.065, $CF^1 = CF^2 = CF^3 = CF^4 = 1$)
R_{16}	**if** *PW* is *long* **then** *Virginica*

	(***c*** = 0.799, ***s*** = 0.199, CF^1 = 0.799, CF^2 = 0.699, CF^3 = CF^4 = 0.598)
R_{17}	**if** *PW* is *completely long* **then** *Virginica* (***c*** = 1, ***s*** = 0.102, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_{18}	**if** *PL* is *long* and *PW* is *long* **then** *Virginica* (***c*** = 0.849, ***s*** = 0.137, CF^1 = 0.849, CF^2 = 0.773, CF^3 = CF^4 = 0.698)
R_{19}	**if** *PL* is *long* and *PW* is *completely long* **then** *Virginica* (***c*** = 1, ***s*** = 0.072, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_{20}	**if** *PL* is *completely long* and *PW* is *long* **then** *Virginica* (***c*** = 1, ***s*** = 0.036, CF^1 = CF^2 = CF^3 = CF^4 = 1)
R_{21}	**if** *PL* is *medium* and *PW* is *long* **then** *Virginica* (***c*** = 0.506, ***s*** = 0.026, CF^1 = 0.506, CF^2 = 0.259, CF^3 = CF^4 = 0.012)
R_{22}	**if** *PL* is *completely long* and *PW* is *completely long* **then** *Virginica* (***c*** = 1, ***s*** = 0.026, CF^1 = CF^2 = CF^3 = CF^4 = 1)

Bảng 2.5: Tỷ lệ (%) số mẫu phân lớp đúng của hệ luật trong bảng 2.4 theo các đánh giá trọng số luật với hai phương pháp lập luận

Trọng số luật được sử dụng	Phương pháp lập luận	
	single winner rule	*weighted vote*
Hệ luật sử dụng CF^0	97.33	95.33
Hệ luật sử dụng CF^1	95.33	96
Hệ luật sử dụng CF^2	95.33	95.33
Hệ luật sử dụng CF^3	95.33	95.33
Hệ luật sử dụng CF^4	95.33	95.33

Hệ luật sinh bởi thuật toán ***IFRG2*** chứa nhiều luật dư thừa, chẳng hạn loại hoa *Setosa* chỉ cần một luật R_1 là đủ để phân lớp (Hình 2.9) trong khi hệ này có đến 7 luật phân lớp cho *Setosa*. Thuật toán này sinh các luật đơn giản với ít thuộc tính tham gia trong phần điều kiện. So sánh với Ví dụ 2.1, loại hoa *Setosa* có thể phân

lớp bởi một luật R_1 chỉ có 1 thuộc tính trong khi thuật toán ***IFRG*1** sinh luật luôn có đủ 2 thuộc tính. Điều này minh họa thuật toán ***IFRG*2** có thể loại bỏ những thuộc tính ít liên quan đến việc phân lớp trong các luật.

Sự bùng nổ tổ hợp các điều kiện của vế trái khi sinh luật thứ cấp trong thuật toán ***IFRG*2** dẫn đến hệ luật khởi đầu chứa khá nhiều luật dư thừa. Chẳng hạn 4 luật R_1, R_2, R_3 và R_4 trong Bảng 2.4 là dư thừa trong khi chỉ cần một trong số chúng đều có thể phân lớp cho loại hoa *Setosa*. Trong phần tiếp theo chúng ta sẽ xem xét phương pháp rút gọn hệ luật mờ bằng phép sàng.

2.3.4 Phương pháp rút gọn hệ luật bằng phép sàng

Một phương pháp rút gọn tập luật được các tác giả trong [43] áp dụng đó là dựa trên đánh giá độ hỗ trợ và độ tin cậy của các luật (công thức (1.9) và (1.10)). Theo phương pháp này, luật nào được xác định từ siêu hộp trong phân hoạch chứa nhiều mẫu dữ liệu sẽ được ưu tiên lựa chọn, còn gọi phương pháp sàng theo tiêu chuẩn. Có ba tiêu chuẩn hay dùng đó là:

- Tiêu chuẩn sàng theo độ tin cậy của luật: $SR^1 = \boldsymbol{c}(A_q \Rightarrow C_q)$,

- Tiêu chuẩn sàng theo độ hỗ trợ luật: $SR^2 = \boldsymbol{s}(A_q \Rightarrow C_q)$,

- Tiêu chuẩn sàng dạng tích: $SR^3 = \boldsymbol{c}(A_q \Rightarrow C_q).\boldsymbol{s}(A_q \Rightarrow C_q)$.

Phương pháp sàng được đề xuất trong [43] là sàng cân bằng. Các luật trong $\boldsymbol{S_0}$ được chia nhóm theo nhãn phân lớp là phần kết luận của luật, như vậy chúng ta có m (m là số lớp) nhóm luật. Chọn ra trong mỗi nhóm một số lượng các luật như nhau (cân bằng) sao cho có giá trị của tiêu chuẩn sàng từ cao xuống thấp. Phương pháp này yêu cầu chọn ra một hệ luật $\boldsymbol{S}^*$ với số luật xác định trước, giả sử là M. Khi đó nếu M chia hết cho m thì tại mỗi nhóm lấy ra M /m luật, ngược lại sẽ lấy $\lfloor M/m \rfloor$ luật, trong đó ký hiệu $\lfloor \bullet \rfloor$ là số nguyên lớn nhất không lớn hơn "$\bullet$", còn lại $M - (m . \lfloor M/m \rfloor)$ luật sẽ được chọn theo giá trị của tiêu chuẩn sàng từ cao xuống thấp trên tất cả các nhóm.

Phương pháp sàng cân bằng trên sẽ không phù hợp đối với tập mẫu không cân bằng số mẫu giữa các lớp. Chẳng hạn trong tập mẫu có 100 mẫu thuộc lớp thứ nhất cần nhiều luật để phân lớp trong khi lớp thứ hai chỉ có 5 mẫu cần 1 luật, nếu chọn cân bằng mỗi lớp ra 2 luật thì lớp thứ hai thừa luật trong lớp thứ nhất lại thiếu. Trên cơ sở đó, luận án đề xuất phương pháp sàng không cân bằng. Mỗi nhóm luật sẽ được chọn ra số luật theo tỷ lệ của số mẫu trong lớp đó. Tức là lớp nào có nhiều mẫu sẽ được ưu tiên chọn nhiều luật và ngược lại. Giả sử t_C là tỷ lệ số mẫu dữ liệu thuộc lớp C trong tập mẫu, M là số luật cần chọn, khi đó lớp C sẽ được chọn ra $\lfloor t_C.M/m \rfloor$ luật.

Tuy nhiên phương pháp sàng luật theo các tiêu chuẩn sàng như trên chỉ mang tính *heuristic*, chưa thể khẳng định hệ luật được chọn thỏa mãn các mục tiêu trong (1.6). Dù sao, với một hệ luật quá lớn sinh ra bởi thuật toán ***IFRG2***, việc loại bỏ dưới dạng thô bằng phương pháp sàng phần nào giảm bớt dư thừa và sự chồng chéo giữa các luật, qua đó tăng hiệu quả phân lớp của hệ luật thu được. Điều này được thể hiện qua các ví dụ sau.

Ví dụ 2.5. Minh họa phương pháp sàng theo ba tiêu chuẩn ***c***, ***s*** và ***c.s*** cho hệ luật sinh bởi thuật toán ***IFRG2*** trong Ví dụ 2.4 (hệ luật trong Bảng 2.4).

Bài toán *IRIS*2 có 3 lớp, tỷ lệ các mẫu trong mỗi lớp cân bằng và đều là 50/150. Chúng ta chọn ra hệ có M = 6 luật, vậy mỗi lớp sẽ lấy hai luật. Kết quả hệ luật thu được có độ dài trung bình của hệ cùng với tỷ lệ phân lớp chính xác được thể hiện trong Bảng 2.6. Trong đó áp dụng cả hai phương pháp lập luận (*single winner rule* - *SWR* và *weighted vote* - *WV*) theo từng tiêu chuẩn sàng và các phương pháp đánh giá trọng số luật.

Qua ví dụ trên, tiêu chuẩn sàng SR^1 cho kết quả độ chính xác phân lớp thấp hơn nhiều trong khi độ dài trung bình của hệ luật lại cao hơn so với hai tiêu chuẩn còn lại (SR^2 và SR^3). So sánh với Bảng 2.5, số luật giảm đi rất nhiều (72.7%) và độ dài trung bình của hệ luật cũng giảm xuống (33.2%), nhưng hiệu quả phân lớp trong hai tiêu chuẩn SR^2 và SR^3 vẫn được đảm bảo. Điều này minh họa cho phương pháp sàng để rút gọn hệ luật và đảm bảo hiệu năng phân lớp.

Bảng 2.6: Kết quả áp dụng phương pháp sàng trên hệ luật trong Bảng 2.4

Tiêu chuẩn sàng	Hệ luật thu được	Độ dài trung bình	Phương pháp lập luận	Tỷ lệ (%) số mẫu phân lớp đúng (theo từng trọng số luật được sử dụng)				
				CF^0	CF^1	CF^2	CF^3	CF^4
SR^1	$\{R_1, R_3, R_{12}, R_{13}, R_{15}, R_{17}\}$	1.33	*SWR*	78.0	78.0	78.0	78.0	78.0
			WV	**78.0**	**78.0**	**78.0**	**78.0**	**78.0**
SR^2	$\{R_1, R_3, R_8, R_9, R_{14}, R_{16}\}$	1.0	*SWR*	97.33	95.33	95.33	95.33	95.33
			WV	**97.33**	**96.67**	**95.33**	**95.33**	**95.33**
SR^3	$\{R_1, R_3, R_8, R_9, R_{14}, R_{16}\}$	1.0	*SWR*	97.33	95.33	95.33	95.33	95.33
			WV	**97.33**	**96.67**	**95.33**	**95.33**	**95.33**

Kết quả phân lớp của hai tiêu chuẩn sàng SR^2 và SR^3 là như nhau, nhưng trong một số trường hợp tiêu chuẩn sàng SR^3 sẽ đạt hiệu quả cao hơn, điều này đã được phân tích trong [43]. Để thấy rõ hơn chúng ta thực hiện tiếp ví dụ sau.

Ví dụ 2.6. Áp dụng thuật toán sinh hệ luật khởi đầu ***IFRG2*** và phương pháp sàng luật đối với bài toán phân lớp các loại rượu (*WINE*).

Tập dữ liệu mẫu phân lớp cho bài toán *WINE* được thu thập bởi M. Forina và các cộng sự năm 1991 tại viện Công nghệ phân tích thực phẩm và dược phẩm, Italy. Hiện nay được công bố tại [76], đã có nhiều tác giả nghiên cứu sử dụng để thử nghiệm cho các mô hình phân lớp trong khai phá dữ liệu [10], [23], [40], [42]-[46], [50], [53], [59], [60]. Tập dữ liệu gồm N = 178 mẫu với m = 3 loại rượu. Mỗi mẫu có n = 13 thuộc tính gồm *Alcohol* (*AL*), *Malic acid* (*MA*), *Ash* (*AS*), *Alcalinity of ash* (*AA*), *Magnesium* (*MG*), *Total phenols* (*TP*), *Flavanoids* (*FL*), *Nonflavanoid phenols* (*NP*), *Proanthocyanins* (*PR*), *Color intensity* (*CI*), *Hue* (*HU*), *OD280/OD315 of diluted wines* (*OD*), *Proline* (*PL*). Tập dữ liệu mẫu này có số mẫu không cân bằng trên các lớp với tỷ lệ là 59/71/48.

Bộ tham số mờ gia tử được cho giống nhau trên các thuộc tính $fm_j(\boldsymbol{c}^-) = fm_j(\boldsymbol{c}^+) = 0.5$, $\mu_j(L) = \mu_j(V) = 0.5$, $k_j = 1$, j=1,2,...,n. Áp dụng thuật toán ***IFRG2*** với giới hạn

độ dài luật là $L \leq 3$, sinh hệ luật khởi đầu gồm 12533 luật giảm 54.3% so với phương pháp trong [43] (27423 luật). Thực hiện phương pháp sàng để rút gọn hệ luật theo 3 tiêu chuẩn với các tập luật kích thước khác nhau gồm 3, 6, 9, 30, 60, 90, 300, 600 và 900 luật. Kết quả tỷ lệ (%) số mẫu phân lớp đúng theo cả hai phương pháp lập luận *single winner rule* (*SWR*) và *weighted vote* (*WV*) thể hiện trong Bảng 2.7. Mỗi dòng tương ứng với một tiêu chuẩn sàng và kiểu đánh giá trọng số của luật theo các công thức (1.15)-(1.17), trong đó CF^0 là trường hợp không áp dụng trọng số, độ dài trung bình hệ luật là tổng số điều kiện của vế trái các luật chia cho số luật. Kết quả tốt nhất trong từng trường hợp được in đậm.

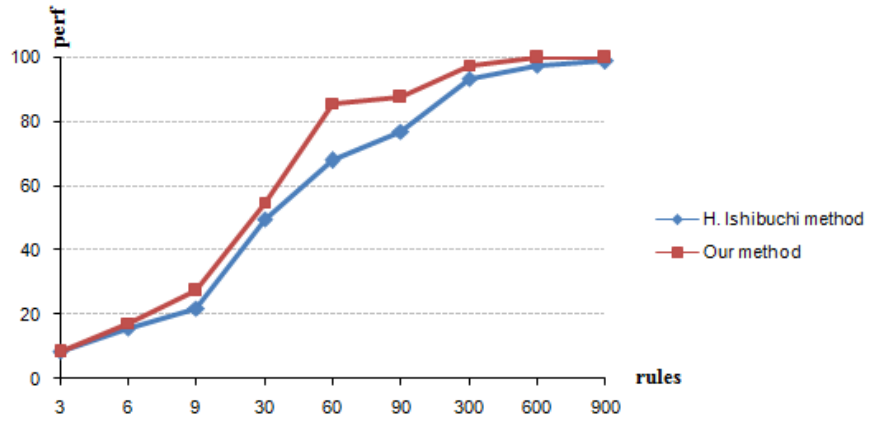
Bảng 2.7: Tỷ lệ (%) số mẫu phân lớp đúng theo mỗi phương pháp sàng

Sàng	Trọng số	Lập luận	Số luật của mỗi tập luật sàng								
			3	6	9	30	60	90	300	600	900
Độ dài trung bình hệ luật			1.67	2.00	2.22	2.50	2.57	2.57	2.76	2.85	2.88
SR^1	CF^0	*SWR*	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
		WV	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
SR^1	CF^1	*SWR*	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
		WV	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
SR^1	CF^2	*SWR*	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
		WV	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
SR^1	CF^3	*SWR*	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
		WV	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
SR^1	CF^4	*SWR*	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
		WV	8.43	12.36	27.53	58.99	83.15	86.52	**97.19**	**100**	**100**
Độ dài trung bình hệ luật			1.0	1.0	1.0	1.37	1.62	1.71	2.10	2.38	2.52
SR^2	CF^0	*SWR*	71.91	72.47	72.47	70.22	68.54	70.22	71.91	70.22	70.22
		WV	71.91	76.97	81.46	86.52	87.64	88.76	92.13	93.82	94.38
SR^2	CF^1	*SWR*	82.02	85.39	85.96	90.45	89.89	92.13	92.70	92.13	92.13
		WV	82.02	83.71	87.08	93.26	91.01	93.26	96.07	95.51	96.07
SR^2	CF^2	*SWR*	85.39	89.33	90.45	93.82	94.38	96.07	92.70	93.26	93.26
		WV	85.39	88.76	90.45	96.63	96.63	94.94	96.07	96.07	96.07

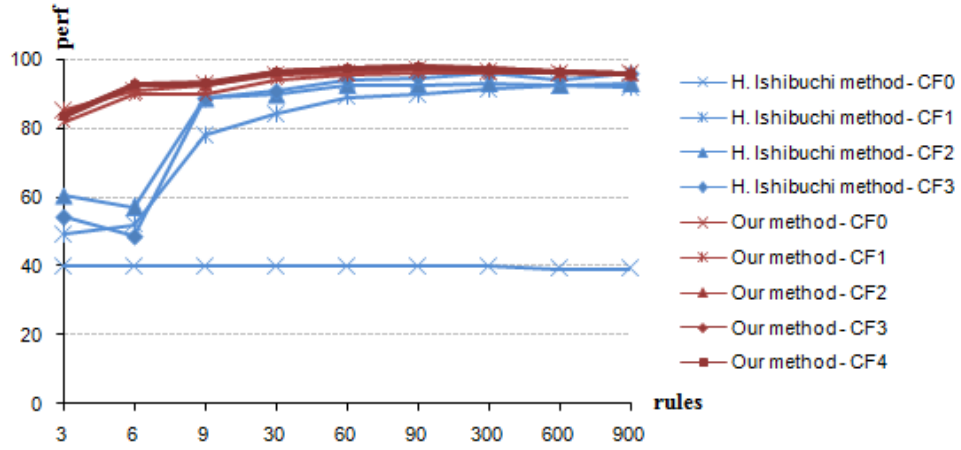
SR^2	CF^3	*SWR*	84.27	91.01	92.70	94.94	95.51	96.63	95.51	95.51	96.07
		WV	84.27	91.01	92.70	96.63	**97.75**	96.63	96.07	96.07	96.07
SR^2	CF^4	*SWR*	83.71	92.13	93.26	**97.19**	96.63	**97.19**	93.82	94.38	94.38
		WV	83.71	92.13	94.94	96.63	**97.75**	**97.19**	96.07	96.07	96.07
Độ dài trung bình hệ luật			1.0	1.17	1.11	1.53	1.72	1.83	2.17	2.44	2.56
SR^3	CF^0	*SWR*	81.46	77.53	81.46	73.03	70.79	70.79	71.35	71.91	71.91
		WV	81.46	89.33	91.57	93.26	92.13	94.38	95.51	96.07	96.07
SR^3	CF^1	*SWR*	87.64	88.76	92.13	92.70	91.57	91.57	92.70	92.70	92.13
		WV	87.64	93.26	94.38	94.94	94.38	94.94	96.07	96.07	96.07
SR^3	CF^2	*SWR*	89.33	92.70	94.94	94.94	93.26	93.26	93.26	93.26	93.82
		WV	89.33	94.94	96.63	95.51	94.94	94.94	96.07	96.07	96.07
SR^3	CF^3	*SWR*	**91.01**	93.82	96.07	96.07	95.51	95.51	94.94	96.07	96.07
		WV	**91.01**	**95.51**	97.19	96.07	96.63	96.07	96.07	96.07	96.07
SR^3	CF^4	*SWR*	**91.01**	93.26	95.51	96.07	94.94	95.51	93.82	94.38	94.38
		WV	**91.01**	**95.51**	**97.75**	96.07	97.19	96.07	96.63	96.07	96.07

Tiêu chuẩn sàng ***c*** có khuynh hướng chọn các luật với độ tin cậy cao nhưng độ hỗ trợ thấp. Một luật được chọn sẽ phân lớp đúng đối với một số nhỏ các mẫu dữ liệu mà nó bao trùm, hay tính phổ quát của nó không cao, do đó với số lượng luật ít khó có thể phân lớp cho một lượng lớn các mẫu dữ liệu. Như chúng ta thấy trong bảng trên, tiêu chuẩn sàng SR^1 = ***c*** đạt kết quả 100% với hệ 600 và 900 luật, trong khi hệ 3 luật chỉ đạt 8.43%. Tiêu chuẩn sàng này chọn hệ luật với kết quả phân lớp không bị tác động bởi trọng số của luật.

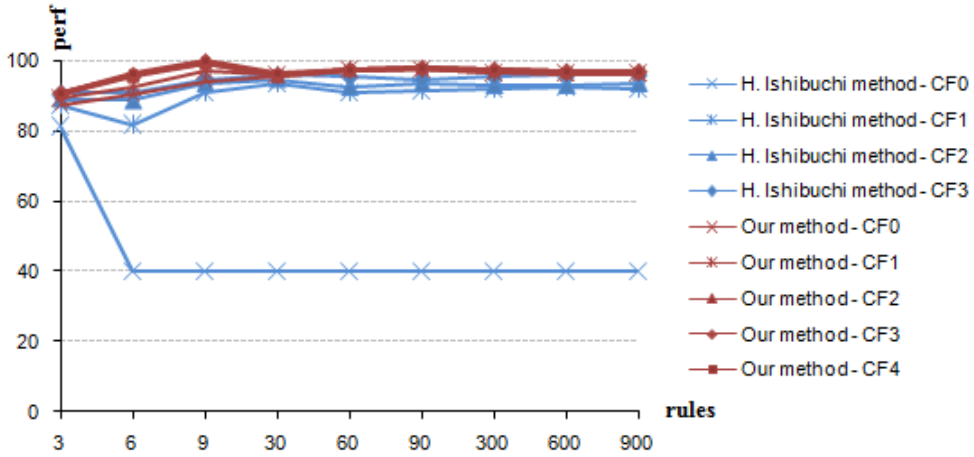
Mặt khác tiêu chuẩn sàng SR^2 = ***s*** có khuynh hướng chọn các luật với độ hỗ trợ cao nhưng độ tin cậy thấp. Luật được chọn có khả năng bao trùm nhiều mẫu dữ liệu hơn, tức tính phổ quát cao nhưng cũng sinh khá nhiều lỗi trong đó. Kết quả tốt nhất trong hai bảng trên theo tiêu chuẩn sàng SR^2 xuất hiện ở trường hợp từ 30 đến 90 luật. Điều này dẫn đến việc áp dụng tiêu chuẩn sàng tích SR^3 = ***c.s*** để đạt được thỏa hiệp giữa tính cá thể và tính phổ quát của luật, thể hiện rõ trong bảng trên. Kết quả phân lớp tốt nhất với trường hợp số luật nhỏ chủ yếu tập trung ở tiêu chuẩn sàng SR^3, và hầu hết xuất hiện tại trọng số luật CF^3, CF^4.



Hình 2.10: Kết quả phân lớp theo tiêu chuẩn sàng ***c***



Hình 2.11: Kết quả phân lớp theo tiêu chuẩn sàng ***s***



Hình 2.12: Kết quả phân lớp theo tiêu chuẩn sàng ***c.s***

So sánh với phương pháp của H. Ishibuchi [43], mặc dù hệ luật khởi đầu trong luận án có số lượng giảm hơn một nửa nhưng khi áp dụng các tiêu chuẩn sàng vẫn

cho kết quả phân lớp với độ chính xác cao hơn trong hầu hết các trường hợp. Các Hình 2.10, 2.11 và 2.12 thể hiện so sánh kết quả phương pháp của luận án với [43] theo phương pháp lập luận *single-winner-rule*, đặc biệt rất tốt tại các trường hợp ít luật.

Thông thường áp dụng phương pháp sàng trên cho hệ luật sinh bởi thuật toán ***IFRG*2** vì số lượng các luật trong trường hợp này thường rất lớn và thời gian thực hiện phương pháp sàng nhanh (theo độ phức tạp của thuật toán sắp xếp, $log_2|\boldsymbol{S_0}|$).

2.4 Kết luận Chương 2

Trong chương này đã đề xuất một đại số gia tử hạn chế gồm hai gia tử - $\mathcal{A}x^2$, từ đó xây dựng hệ phân hoạch các khoảng tương tự cho tập các giá trị ngôn ngữ $\boldsymbol{X}_{(k)}$ thay vì áp dụng hệ phân hoạch các khoảng tính mờ đối với tập giá trị ngôn ngữ $\boldsymbol{X}_k$. Trên cơ sở những phân hoạch này luận án đã thiết kế hai thuật toán sinh hệ luật khởi đầu là ***IFRG*1** và ***IFRG*2** cho bài toán phân lớp theo tiếp cận hệ luật mờ. Khác với những thuật toán có độ phức tạp hàm mũ trong [42]-[47], cả hai thuật toán này đều được khẳng định là đa thức đối với tập dữ liệu mẫu.

Tuy vậy những hệ luật sinh bởi hai thuật toán trên có thể chứa những luật dư thừa. Với mục tiêu xây dựng hệ luật đơn giản, dễ hiểu và đạt hiệu quả cao cho bài toán phân lớp chúng tôi thiết kế hai phương pháp để loại bỏ các luật dư thừa bằng phép sàng dựa trên các tiêu chuẩn đánh giá của luật, hoặc bằng phép kết nhập dựa trên hàm đo mức độ gần nhau giữa các luật mờ.

Hai thuật toán sinh luật cùng với phương pháp rút gọn hệ luật đã được minh họa thử nghiệm trong các ví dụ và cho kết quả khả quan. Đặc biệt trong so sánh với các phương pháp khác đối với hai bài toán được sử dụng khá phổ biến là phân lớp các loại hoa (*iris*) và phân lớp các loại rượu (*wine*).

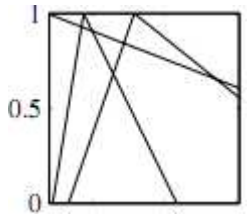
CHƯƠNG 3
PHƯƠNG PHÁP THIẾT KẾ NGÔN NGỮ VÀ TỐI ƯU HỆ LUẬT

3.1 Phương pháp thiết kế ngôn ngữ cho bài toán phân lớp

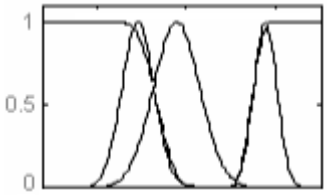
3.1.1 Đặt bài toán

Theo phương pháp sinh luật dựa trên lưới phân hoạch mờ đã trình bày trong Chương 2, có hai vấn đề cần được xem xét: Thứ nhất, mỗi thuộc tính của bài toán cần sử dụng những giá trị ngôn ngữ nào là phù hợp. Thứ hai, mỗi giá trị ngôn ngữ sử dụng phải được thiết kế hàm định lượng ngữ nghĩa sao cho thích hợp nhất đối với bài toán. Trong các phương pháp dựa trên tập mờ, các tác giả trong [17], [29], [40], [50], [74]... thiết kế trước một số các tập mờ cùng với các giá trị ngôn ngữ tương ứng, sau đó điều chỉnh các tham số của các tập mờ đã được thiết kế (hay gọi là “tunning”) bằng cách sử dụng các chiến lược tìm kiếm tối ưu xấp xỉ chủ yếu dựa trên giải thuật di truyền (GA).

Trong các phương pháp này vẫn tồn tại sự tách biệt giữa cú pháp của các giá trị ngôn ngữ và ngữ nghĩa của chúng biểu diễn bởi các tập mờ. Các tác giả chỉ tập trung vào mục đích thiết kế tập mờ biểu diễn ngữ nghĩa của ngôn ngữ, việc bỏ qua cú pháp của ngôn ngữ dẫn đến các giá trị ngôn ngữ chỉ đóng vai trò như là nhãn của các tập mờ, trong khi ngôn ngữ lại mang những thông tin quan trọng đối với con người. Trong [43], [60], giá trị ngôn ngữ được xem như là nhãn và ngữ nghĩa của chúng được giả định là cho trước. Một số tác giả [50], [40], [17] áp dụng phương pháp “tunning” các tham số tập mờ biểu diễn ngữ nghĩa, tuy nhiên kết quả các tập mờ không mang ngữ nghĩa của ngôn ngữ, dẫn đến hệ luật mờ trở nên khó hiểu và không trực quan đối với người dùng. Chẳng hạn, rất khó để chọn giá trị ngôn ngữ thích hợp và giải thích ngữ nghĩa của các tập mờ kết quả trong Hình 3.1(a) và 3.1(b) [10], [50]. Thậm chí các tập mờ hầu như không có sự phân biệt, tức tập mờ này gần như được chứa trọn bên trong một tập mờ khác (Hình 3.1 (b)).



(a)



(b)

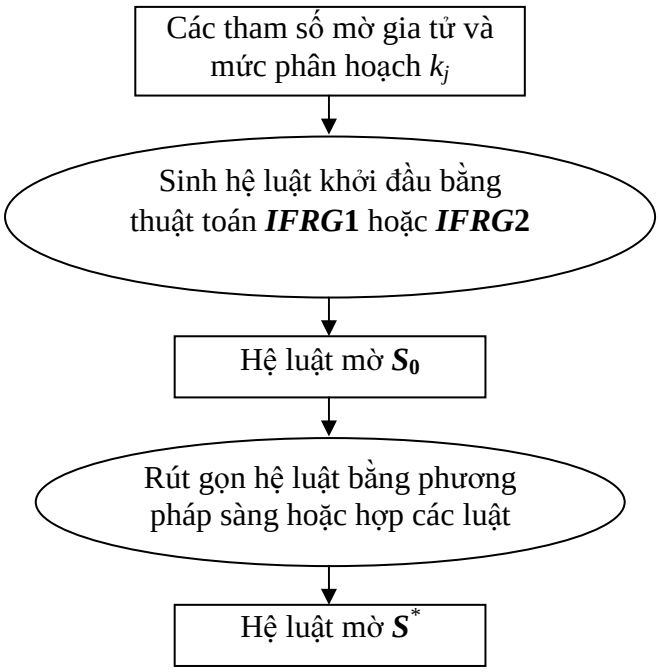
Hình 3.1: Tập mờ của ***Malic Acid*** [10] (a), ***Proline*** [50] (b)

Dựa trên các tính chất của ĐSGT, các tham số mờ gia tử có thể được điều chỉnh nhưng vẫn đảm bảo tính cấu trúc ngữ nghĩa của các hạng từ. Hơn nữa, với mức phân hoạch *k* được chọn để có tập các giá trị ngôn ngữ thích hợp cho bài toán. Rõ ràng, thực tế không có một quy trình hay công cụ để xác định các tham số này cho các bài toán ứng dụng. Phụ thuộc vào ngữ cảnh của bài toán ứng dụng và được kiểm chứng bởi các kết quả thực nghiệm.

Trong phần này của luận án sẽ đề cập phương pháp thiết kế các tập giá trị ngôn ngữ cho các thuộc tính của bài toán dựa trên phương pháp tối ưu tham số mờ gia tử và mức phân hoạch *k* của mô hình xây dựng hệ luật mờ phân lớp đã được trình bày trong Chương 2. Cả hai phương pháp sinh luật của thuật toán ***IFRG*1** và ***IFRG*2** đều dựa trên phân hoạch hệ các khoảng tính mờ hoặc các khoảng tương tự của các giá trị ngôn ngữ trong ĐSGT, tức với mỗi phân hoạch sẽ sinh ra một hệ luật mờ khác nhau và tất nhiên kết quả phân lớp sẽ khác nhau. Rõ ràng việc tính toán phân hoạch các khoảng tính mờ phụ thuộc hoàn toàn vào các tham số mờ gia tử. Như đã giới thiệu trong Chương 1, mục tiêu xây dựng hệ luật nhỏ gọn, đơn giản và dễ hiểu, hiệu quả phân lớp cao. Như vậy bài toán đặt ra là tìm kiếm bộ tham số gia tử và mức phân hoạch cho các ĐSGT của các thuộc tính sao cho hệ luật thu được sau quá trình sinh bởi thuật toán ***IFRG*1** hoặc ***IFRG*2** và rút gọn (Hình 3.2) thỏa mãn các mục tiêu trong (1.6). Để tiện về sau chúng ta gọi quá trình xây dựng hệ luật này là ***HAFRG*** (*Hedge Algebras based Fuzzy Rules Generation for Classification*).

Tuy nhiên ngoài các tham số mờ gia tử, tham số về mức phân hoạch các khoảng tính mờ hoặc các khoảng tương tự k_j cũng tác động lớn đến kết quả của mô hình. Mức phân hoạch càng lớn (k_j càng lớn) sẽ sinh ra càng nhiều luật và các luật có tính phân biệt càng cao làm tăng hiệu năng phân lớp, nhưng số lượng các luật sẽ

rất nhiều. Việc xác định k_j khó thực hiện bằng trực quan của người dùng đối với bất kỳ bài toán nào. Một cách tự nhiên chúng ta cũng đưa việc xác định mức phân hoạch k_j vào bài toán tìm kiếm tối ưu.



Hình 3.2: Quá trình ***HAFRG*** xây dựng hệ luật mờ phân lớp

Ký hiệu $\boldsymbol{PAR} = \{ fm_j(\boldsymbol{c}^-), fm_j(\boldsymbol{c}^+), \mu_j(h_i), k_j \mid \forall h_i \in \boldsymbol{H}, j = 1,...,n \}$ là tập các tham số mờ gia tử cùng với mức phân hoạch k_j của bài toán phân lớp, khi đó $\boldsymbol{S}^* = \boldsymbol{HAFRG}(\boldsymbol{PAR}, \boldsymbol{IFRG1}, \theta_{itg})$ là trường hợp sinh hệ luật sử dụng thuật toán ***IFRG*1** và hợp các luật theo ngưỡng θ_{itg} để rút gọn, hoặc $\boldsymbol{S}^* = \boldsymbol{HAFRG}(\boldsymbol{PAR}, \boldsymbol{IFRG2}, M)$ là quá trình sinh hệ luật sử dụng thuật toán ***IFRG*2** và phương pháp sàng để chọn ra M luật (theo quá trình trong Hình 3.2). Để thuận tiện có thể viết gọn $\boldsymbol{S}^* = \boldsymbol{HAFRG}(\boldsymbol{PAR})$ cho cả hai quá trình trên.

Để ý rằng thuật toán ***IFRG*1** có thể áp dụng đối với ĐSGT không hạn chế số gia tử, còn ***IFRG*2** chỉ áp dụng đối với ĐS2GT.

Chúng ta có bài toán tìm kiếm tối ưu tham số mờ gia tử được phát biểu dưới dạng sau.

Các mục tiêu:

$$f_p(\boldsymbol{S}^*) \to max, \ f_n(\boldsymbol{S}^*) \text{ và } f_a(\boldsymbol{S}^*) \to min, \tag{3.1}$$

Các ràng buộc:

i) Điều kiện cho các tham số trong ĐSGT:

$$0 < fm_j(\boldsymbol{c}^-), fm_j(\boldsymbol{c}^+), \mu_j(h_i) < 1, \quad (3.2)$$

$$fm_j(\boldsymbol{c}^-) + fm_j(\boldsymbol{c}^+) = 1, \quad (3.3)$$

$$\sum_i \mu_j(h_i) = 1, \text{ cho mọi } j = 1,2,...,n. \quad (3.4)$$

ii) Điều kiện cho mức phân hoạch:

$$0 < k_j \le \pi_{kj}, \text{ cho mọi } j = 1,2,...,n. \quad (3.5)$$

Trong đó $\boldsymbol{S}^*$ là hệ luật sinh bởi quá trình trên Hình 3.2, π_{kj} là ngưỡng tối đa cho mức phân hoạch tại thuộc tính X_j. Các hàm đánh giá $f_p(\boldsymbol{S}^*)$, $f_n(\boldsymbol{S}^*)$ và $f_a(\boldsymbol{S}^*)$ tương ứng là tỷ lệ % số mẫu phân lớp đúng trên tập dữ liệu (hay gọi là hiệu quả phân lớp), số lượng các luật và độ dài trung bình hệ luật.

Mặt khác, khi áp dụng thuật toán sinh luật ***IFRG2*** và phương pháp sàng để rút gọn hệ luật, nếu chỉ sử dụng một hệ $\boldsymbol{S}^*$ với M luật thì nhiều khả năng hàm đánh giá mục tiêu $f_p(\boldsymbol{S}^*)$ dẫn đến tối ưu tham số mang tính địa phương, tức là nó chỉ tối ưu đối với hệ đúng M luật. Một lý do khác là chúng ta khó xác định số luật M đối với một bài toán để có được bộ tham số tối ưu theo đúng nghĩa của nó (1.6). Ở đây chúng ta sẽ đánh giá mục tiêu hiệu quả phân lớp đối với một bộ tham số dựa trên nhiều hệ luật, giả sử với bộ tham số ***PAR***, áp dụng thuật toán sinh luật ***IFRG2*** để sinh hệ luật $\boldsymbol{S_0}$. Đặt N_s là số lượng các hệ luật cần dùng để đánh giá hiệu quả đối với ***PAR***, khi đó sử dụng phương pháp sàng để lấy ra N_s hệ luật $\{S^*_1, S^*_2,..., S^*_{Ns}\}$ có số lượng các luật lần lượt là $M_1, M_2,..., M_{Ns}$, gọi N_s hệ này là ***Set***(N_s) (ký hiệu $\boldsymbol{S}^{**}$).

$$\textbf{\textit{Set}}(N_s) = \{S^*_i \mid S^*_i = \textbf{\textit{HAFRG}}(\textbf{\textit{PAR}}, \textbf{\textit{IFRG2}}, M_i), i = 1,...,N_s\},$$

trong đó các M_i được cho trước. Theo cách này, chúng ta đánh giá các mục tiêu trong (3.1) dưới dạng trung bình, khi đó ký hiệu $f_p(\boldsymbol{S}^{**}) = \frac{1}{Ns}\sum_{i=1}^{Ns} f_p(S_i^*)$, $f_n(\boldsymbol{S}^{**}) = \frac{1}{Ns}\sum_{i=1}^{Ns} f_n(S_i^*)$ và $f_a(\boldsymbol{S}^{**}) = \frac{1}{Ns}\sum_{i=1}^{Ns} f_a(S_i^*)$ tương ứng là hiệu quả phân lớp, số luật và độ dài trung bình của luật đánh giá trên N_s hệ luật ***Set***(N_s).

Bài toán tối ưu ở trên có 3 mục tiêu. Để áp dụng giải thuật di truyền, theo cách thông thường (xem [45], [43]) chúng ta kết nhập theo trọng số các mục tiêu trong (3.1) thành một mục tiêu dưới dạng sau:

$$w_p.f_p(\boldsymbol{S}^{**}) + w_n.f_n(\boldsymbol{S}^{**})^{-1} + w_a.f_a(\boldsymbol{S}^{**})^{-1} \rightarrow max, \quad (3.6)$$

trong đó $0 < w_p,w_n,w_a < 1$ và $w_p+w_n+w_a = 1$, hiển nhiên $f_n(\boldsymbol{S}^{**}) > 0$ và $f_a(\boldsymbol{S}^{**}) > 0$ do đó $f_n(\boldsymbol{S}^{**})^{-1}$ và $f_a(\boldsymbol{S}^{**})^{-1}$ luôn tồn tại.

Theo điều kiện (3.3), ta giảm bớt tham số $fm(\boldsymbol{c}^+)$ trong tập $\boldsymbol{PAR}$, vì $fm(\boldsymbol{c}^+) = 1-fm(\boldsymbol{c}^-)$. Hơn nữa, trong ĐS2GT chỉ gồm hai gia tử $\{L,V\}$, từ điều kiện (3.4) cho phép giảm bớt tham số $\mu(V)$, vì $\mu(V) = 1-\mu(L)$. Như vậy không gian tìm kiếm sẽ được thu hẹp rất nhiều, cụ thể

$$\boldsymbol{PAR} = \{ fm_j(\boldsymbol{c}^-), \mu_j(L), k_j \mid j = 1,2,...,n \}, \quad (3.7)$$

khi đó $|\boldsymbol{PAR}| = 3.n$ thay vì $(|\boldsymbol{H}|+3).n$ như ban đầu, dẫn đến tốc độ tìm kiếm sẽ nhanh hơn. Các ràng buộc (3.3) và (3.4) đã được loại bỏ, bài toán chỉ còn hai ràng buộc (3.2) và (3.5).

Ngoài ra, để các giá trị ngôn ngữ mang ngữ nghĩa và định lượng của chúng thích hợp trong thực tế, chúng ta sẽ đặt ràng buộc (3.2) theo khoảng giới hạn nhỏ hơn. Nếu hai giá trị ngôn ngữ *short* và *long* có độ đo tính mờ $fm(short) = 0.95$ và $fm(long) = 0.05$ thì rõ ràng không thực tế. Tuy nhiên việc thu hẹp giới hạn của các tham số sẽ làm cho hiệu quả của mô hình có thể không cao, nhưng các giá trị ngôn ngữ biểu diễn trong các luật mờ có tính thực tế hơn. Đây là một sự thỏa hiệp giữa hiệu quả và tính dễ hiểu của hệ luật.

Bây giờ ràng buộc (3.2) trở thành:

$$La_j < fm_j(\boldsymbol{c}^-) < Lb_j,\ Lc_j < \mu_j(L) < Ld_j,\ j=1,2,...,n, \quad (3.8)$$

trong đó các hằng số thỏa mãn $0 \le La_j < Lb_j \le 1$ và $0 \le Lc_j < Ld_j \le 1$ được lựa chọn bởi người dùng để phù hợp với thực tế của bài toán.

Tiếp theo chúng ta sẽ thiết kế phương pháp giải bài toán sử dụng giải thuật di truyền (GA - *Genetic Algorithm*) có kết hợp thuật toán mô phỏng tôi luyện (SA - *Simulated Annealing*).

3.1.2 Phương pháp tối ưu tham số dựa trên giải thuật di truyền lai

Hai thuật toán GA [14], [51], [61] và SA [11], [13], [62] bản thân chúng là những phương pháp thích nghi để tìm kiếm tối ưu. Trong khi GA là một thuật toán tìm kiếm ngẫu nhiên chủ yếu dựa trên hai phép toán gen là lai ghép và đột biến thì SA được xem như một thuật toán áp dụng kỹ thuật "leo đồi" theo xác suất. Đã có nhiều tác giả nghiên cứu áp dụng GA và SA để giải bài toán tối ưu. Tuy nhiên, mỗi thuật toán đều có những thế mạnh riêng và một số tác giả đã kết hợp hai thuật toán trên làm tăng sức mạnh tìm kiếm [1], [11], [13].

Trong [1], các tác giả đã nhúng tham số nhiệt độ T mô phỏng nhiệt tôi luyện trong SA để điều khiển các phép toán gen của GA. Trong đó, các tham số xác suất để chọn lọc, lai ghép và đột biến được thay đổi qua từng thế hệ di truyền, chúng được tính tỷ lệ với tham số nhiệt độ hiện thời của thế hệ hiện tại. Tham số nhiệt ban đầu được tính dựa trên số thế hệ tiến hóa (thường khá lớn đảm bảo tính đa dạng của quần thể), sau mỗi thế hệ tham số nhiệt giảm dần để đảm bảo tính hội tụ và tính ổn định. Kết hợp này phù hợp với các chiến lược thay đổi tham số trong GA, làm tăng tốc độ hội tụ tìm kiếm và hiệu quả của thuật toán, ta gọi đây là thuật toán di truyền lai (SGA).

Tập tham số ***PAR*** trong (3.7) được xem như chuỗi gen mã hóa cho một cá thể trong SGA, giá trị hàm thích nghi của cá thể, ký hiệu *Fit*, tính theo công thức (3.6). Các phép toán di truyền được áp dụng như sau [1]:

(1) **Phép chọn lọc** (*SGA_Selection*): sử dụng sơ đồ chọn lọc xếp hạng không tuyến tính theo hàm số mũ, các cá thể được sắp xếp theo thứ tự giảm của giá trị hàm *Fit*, cá thể thứ i (xếp hạng i) sẽ được chọn vào quần thể bố mẹ (*parent*) theo xác suất sau:

$p_i = ((1\text{-}a).a^{-i})/(a^{-Npop}\text{-}1)$, với

$a = 1+\gamma(T_k)/N_{pop}$,

$\gamma(T_k) = 1+(\gamma_{max} - 1).(ln(T_0)\text{-}ln(T_k))/(ln(T_0)\text{-}ln(T_{\text{end}}))$,

trong đó N_{pop} là số cá thể trong quần thể, $T_k = T_0.\alpha^k$ là nhiệt độ tôi luyện ở thế hệ hiện tại k (k = 1,...,G_{max}), tham số T_k này giảm từ nhiệt độ ban đầu T_0 đến $T_{end} = T_0.\alpha^{Gmax}$, với $0<\alpha<1$ (thường chọn α=0.7), G_{max} là số thế hệ cần tiến hóa, hàm $\gamma(T_k)$ sẽ tăng tuyến tính theo số thế hệ đã tiến hóa từ 1 đến γ_{max} (thường chọn γ_{max}=9).

(2) **Phép lai ghép** (*SGA_Crossover*): sẽ chọn ngẫu nhiên theo phân bố đều một trong 3 phép lai ghép một điểm cắt, lai ghép tuyến tính và lai ghép tuyến tính mở rộng, cụ thể như sau:

(*i*) *Lai ghép một điểm cắt*: cho một cá thể X, ký hiệu $X|_i$ là nửa đầu của chuỗi gen trên X tính đến vị trí i và $_i|X$ là nửa phần còn lại của chuỗi gen tính từ i+1 đến hết. Với hai cá thể bố mẹ X, Y chọn để lai ghép, một vị trí ngẫu nhiên i được chọn theo phân bố đều trên chiều dài chuỗi gen của hai bố mẹ, hai cá thể con U, V sinh ra bằng cách ghép phần đầu của X với phần sau Y và ngược lại:

$U = X|_i \oplus {}_i|Y$,

$V = Y|_i \oplus {}_i|X$,

trong đó $\oplus$ ký hiệu của phép ghép nối hai chuỗi gen.

(*ii*) *Lai ghép tuyến tính*: chọn a ngẫu nhiên phân bố đều trong khoảng (0,1), sinh hai cá thể con U, V từ hai cá thể bố mẹ X, Y như sau:

$U[j] = a.X[j] + (1\text{-}a).Y[j]$,

$V[j] = (1\text{-}a).X[j] + a.Y[j]$, j = 1,...,L_{idv},

trong đó $X[j]$ là gen thứ j của cá thể X, L_{idv} là độ dài chuỗi gen của các cá thể. Để tiện về sau ta viết gọn lại dưới dạng $U = a.X + (1\text{-}a).Y$ và $V = (1\text{-}a).X + a.Y$.

(*iii*) *Lai ghép tuyến tính mở rộng*: chọn một điểm cắt *i* chia mỗi chuỗi gen của cá thể bố mẹ *X* và *Y* thành 2 phần, áp dụng phép lai ghép tuyến tính trên mỗi phần chia tạo thành cặp cá thể con như sau:

$$U|_i = a.X|_i + (1\text{-}a).Y|_i,\ \ {}_i|U = (1\text{-}a).{}_i|X + a.{}_i|Y,$$

$$V|_i = (1\text{-}a).X|_i + a.Y|_i,\ \ {}_i|V = a.{}_i|X + (1\text{-}a).{}_i|Y,$$

$$U = U|_i \oplus {}_i|U,\ V = V|_i \oplus {}_i|V,$$

trong đó *a* được chọn ngẫu nhiên phân phối đều trong khoảng (0,1).

(3) **Phép đột biến** (*SGA_Mutation*): giả sử một gen có giá trị $X[i]$ nằm trong khoảng giới hạn giá trị là $[Lc, Ld]$ được chọn để đột biến, giá trị gen sau đột biến là:

$$X[i]' = X[i] + z.(X[i] - Lc) \text{ nếu } u < 0.5,$$

$$= X[i] + z.(Ld - X[i]) \text{ nếu ngược lại,}$$

trong đó *u* chọn ngẫu nhiên trong đoạn [0,1], và

$$z = sign(u\text{-}0.5).T_k.[(1+1/T_k)^{|2u\text{-}1|} - 1],$$

với T_k là nhiệt độ tôi luyện tại thế hệ thứ *k*.

(4) **Phép thay thế** (*SGA_Replacement*): là phương pháp thay thế bố mẹ bằng các cá thể con, mỗi cá thể con sẽ cạnh tranh với cá thể tốt nhất trong hai cá thể bố mẹ. Gọi g_{p1}, g_{p2}, g_c tương ứng là giá trị hàm mục tiêu của hai cá thể bố mẹ và cá thể con, đặt $g^* = max\{g_{p1}, g_{p2}\}$, khi đó cá thể con được chấp nhận với xác suất $p = min\{1,\ e^{-(g_c - g^*)/T_k}\}$. Trong trường hợp cá thể con không được chấp nhận, cá thể bố mẹ tương ứng với g^* được chấp nhận để đưa vào thế hệ tiếp theo.

Bây giờ chúng ta sẽ thiết kế thuật toán tìm kiếm tối ưu bộ tham số gia tử (bài toán (3.6, 3.8, 3.5)) dựa trên các phép toán của SGA trên. Gọi thuật toán này là ***FPO-SGA*** (*fuzzy parameters optimization - SGA*).

Inputs:

- Tập dữ liệu mẫu $\boldsymbol{D} = \{ (p_i; c_i) \mid i=1, ..., N \}$, $p_i = (d_{i,1}, ..., d_{i,n}) \in \boldsymbol{P}$, $c_i \in \boldsymbol{C} = \{C_1, ..., C_m\}$, *n* là số thuộc tính, *m* là số lớp, *N* là số mẫu dữ liệu;

- Giới hạn ràng buộc các tham số theo (3.8) gồm: $0 \le La_j < Lb_j \le 1$, và $0 \le Lc_j < Ld_j \le 1$;

- Trọng số cho các mục tiêu của hàm thích nghi (3.6): $0 < w_p, w_n, w_a < 1$, $w_p + w_n + w_a = 1$;

Outputs:

- Bộ các tham số mờ gia tử và mức phân hoạch k_j của các thuộc tính ***PAR*** = { $fm_j(\boldsymbol{c}^-), fm_j(\boldsymbol{c}^+) = 1\text{-}\ fm_j(\boldsymbol{c}^-), \mu_j(h), k_j \mid \forall h \in \boldsymbol{H}, j = 1, ..., n$ };

Actions:

*Step*1) Khởi tạo một quần thể xuất phát gồm N_p cá thể Pop_0 = { $p_{0,1}, p_{0,2}, ..., p_{0,Np}$ } để tính tham số nhiệt ban đầu T_0 (ký hiệu $p_{k,i}$ là cá thể thứ i trong quần thể của thế hệ k, nó là mã hóa của bộ tham số ***PAR***, N_p là kích thước của quần thể tại mỗi thế hệ trong SGA);

*Step*2) Với mỗi $p_{0,i} \in Pop_0$, thực hiện quá trình sinh hệ luật phân lớp $\boldsymbol{S}(p_{0,i})$ = ***HAFRG***$(p_{0,i})$. Tính độ phù hợp của mỗi cá thể $Fit(\boldsymbol{S}(p_{0,i}))$ dựa trên hệ luật $\boldsymbol{S}(p_{0,i})$ theo công thức (3.6), và tính tham số nhiệt ban đầu:

$$T_0 = \sqrt{\frac{\sum_{i=1,...,N_p} \left(Fit(\boldsymbol{S}(p_{0,i})) - \frac{1}{N_p} \sum_{j=1}^{N_p} Fit(\boldsymbol{S}(p_{0,j})) \right)^2}{N_p}}$$

*Step*3) Đặt $k = 0$. Lặp theo mỗi k cho đến khi $k = G_{max}$, Pop_k = {$p_{k,1}, p_{k,2}, ..., p_{k,Np}$} và thực hiện

3.*a*) Tính tham số nhiệt cho thế hệ k+1, $T_{k+1} = \alpha^k.T_k$, trong đó $\alpha < 1$ là hệ số giảm nhiệt độ (thường chọn 0.7);

3.*b*) Tạo quần thể mới Pop_{k+1} cho thế hệ k+1 như sau:

Lặp theo i cho đến khi $|Pop_{k+1}| = N_p$,

Chọn hai cặp cá thể bố mẹ $p, q \in Pop_k$ sử dụng phép chọn lọc $SGA\_Selection(Pop_k, T_{k+1})$. Sau đó thực hiện các phép lai

ghép, độ biến và thay thế trên cặp bố mẹ này sử dụng các phép *SGA_Crossover*, *SGA_Mutation*, *SGA_Replacement* để tạo cặp cá thể mới p', q' và đưa vào Pop_{k+1}.

Kết quả $Pop_{k+1} = \{p_{k+1,1}, p_{k+1,2}, ..., p_{k+1,Np}\}$

3.*c*) Với mỗi $p_{k,i} \in Pop_{k+1}$, thực hiện quá trình sinh hệ luật phân lớp $\boldsymbol{S}(p_{k,i}) = \boldsymbol{HAFRG}(p_{k,i})$. Tính độ phù hợp của mỗi cá thể $Fit(\boldsymbol{S}(p_{k,i}))$ dựa trên hệ luật $\boldsymbol{S}(p_{k,i})$ theo công thức (3.6).

*Step*4) Trả về kết quả bộ tham số tương ứng với cá thể có độ phù hợp cao nhất trong thế hệ cuối cùng, $\boldsymbol{PAR}_{opt}$.

End.

Trong thuật này, các tham số để chạy quá trình sinh hệ luật ***HAFRG*** và để thực hiện các phép di truyền của SGA phải được cho trước. Tham số mức phân hoạch k_j là một giá trị nguyên 1,2,3… được mã hóa bởi gen là một số thực trong [0,1] và do đó $k_j = \lceil g_i.k_{max} \rceil$ ($\lceil \bullet \rceil$ là số nguyên bé nhất lớn hoặc bằng $\bullet$, k_{max} là mức phân hoạch tối đa được cho trước.

Kích thước quần thể mỗi thế hệ trong thuật toán ***FPO-SGA*** đều bằng N_p cho trước, với số thế hệ cần tiến hóa cũng cho trước là G_{max} nên số bước lặp để tiến hóa trong thuật toán là hữu hạn. Mặt khác, thuật toán này sử dụng phép chọn lọc xếp hạng theo hàm số mũ và phép thay thế dạng 3 nên có tính "*tinh hoa*" (xem *Định lý 3.1* trong [1]). Tức là, cá thể tốt nhất ở thế hệ sau có độ phù hợp không nhỏ hơn thế hệ trước, do đó kết quả của thuật toán này là cá thể tốt nhất trong toàn bộ quá trình tiến hóa. Tuy nhiên, đây là dạng thuật toán tìm kiếm thích nghi nên khó có thể chứng minh tính hội tụ của chúng đối với mục tiêu đặt ra (như đã phân tích trong [1]).

Các ví dụ dưới đây sẽ minh họa cho phương pháp thiết kế tập giá trị ngôn ngữ cho bài toán phân lớp dựa trên việc tìm kiếm tối ưu các ngữ nghĩa trong ĐSGT.

Ví dụ 3.1. Tối ưu tham số theo thuật toán sinh luật ***IFRG*1**. Áp dụng thuật toán ***FPO-SGA*** trên để tìm kiếm bộ tham số tối ưu cho bài toán phân lớp các loại hoa (*IRIS*2, đã đề cập trong Ví dụ 2.1) theo phương pháp sinh luật dựa trên hệ khoảng tính mờ - ***IFRG*1**.

Chúng ta sử dụng ĐS2GT với giá trị ngôn ngữ sinh là $\boldsymbol{c}^- = short$ và $\boldsymbol{c}^+ = long$. Quá trình sinh hệ luật ***HAFRG*** thực hiện bởi thuật toán ***IFRG*1** và phương pháp rút gọn hệ luật bằng phép hợp với ngưỡng $\theta_{itg} = 0.1$. Giá trị $\rho_L = \rho_R = 0.3$ áp dụng tính hàm thuộc của các giá trị ngôn ngữ (công thức (2.4)). Bài toán gồm 2 thuộc tính *petal length* (*PL*) và *petal width* (*PW*). Bộ tham số gia tử của 2 thuộc tính được mã hóa trong mỗi cá thể gồm $\boldsymbol{PAR} = \{fm_j(\boldsymbol{c}^-), \mu_j(L), k_j \mid j = 1,2\}$, giới hạn các tham số mờ gia tử là $0.1 \le fm_j(\boldsymbol{c}^-), \mu(L) \le 0.9$ và $1 \le k_j \le 3$. Các tham số chạy thuật toán tối ưu ***FPO-SGA*** $\alpha = 0.7$, $\gamma_{max} = 9$, kích thước quần thể tại mỗi thế hệ $N_p = 100$, số thế hệ tiến hóa $G_{max} = 150$. Trọng số cho các thành phần trong hàm mục tiêu (3.6) là $w_p = 0.99$, $w_n = 0.01$, $w_a = 0$ ($w_a = 0$ vì các luật sinh bởi thuật toán ***IFRG*1** đều có độ dài giống nhau và bằng số thuộc tính). Áp dụng trọng số luật CF^3, rút gọn hệ luật bằng phương pháp kết nhập nên không sử dụng tiêu chuẩn sàng, phương pháp lập luận là *single-winner-rule*.

Bảng 3.1: Các tham số gia tử tối ưu bằng thuật toán ***FPO-SGA*** cho bài toán *IRIS*2

	Petal length	*Petal width*
$fm(\boldsymbol{c}^-)$	0.67	0.59
$fm(\boldsymbol{c}^+)$	0.33	0.41
$\mu(L)$	0.79	0.47
$\mu(V)$	0.21	0.53
k_j	3	3

Kết quả bộ tham số mờ gia tử tối ưu và mức phân hoạch hệ khoảng tính mờ thu được sau khi chạy thuật toán ***FPO-SGA*** trong Bảng 3.1. Hệ luật sinh ra theo kết quả này gồm 5 luật trong Bảng 3.2 (mỗi luật có các tham số ngữ nghĩa của giá trị ngôn ngữ bên dưới gồm điểm mút trái, tâm và điểm mút phải của khoảng tính mờ tương ứng, các đánh giá gồm độ tin cậy, độ hỗ trợ và trọng số), với số lỗi phân lớp

là 1/150 mẫu dữ liệu (đạt hiệu quả 99.33%). So với kết quả trong Ví dụ 2.6 (tham số chưa tối ưu), số luật giảm 1 luật (16.67%) nhưng hiệu quả phân lớp tăng lên 2.66%.

Bảng 3.2: Danh sách các luật sinh bởi thuật toán ***IFRG*1** sau khi tối ưu tham số cho bài toán *IRIS*2 (mỗi giá trị ngôn ngữ trong điều kiện của luật được tính các tham số cho hàm định lượng ngữ nghĩa).

R_1	**if** *Petal length* is *short*, and *Petal width* is *V.short* **then** *Setosa* (CF^3 = 0.999) (0, 0.141, 0.67) (0, 0.166, 0.313) $\boldsymbol{c}$ = 0.999, $\boldsymbol{s}$ = 0.128
R_2	**if** *Petal length* is *L.short*, and *Petal width* is *L.short* **then** *Versicolor* (CF^3 = 0.989) (0.141, 0.559, 0.67) (0.313, 0.443, 0.59) $\boldsymbol{c}$ = 0.994, $\boldsymbol{s}$ = 0.155
R_3	**if** *Petal length* is *VL.long*, and *Petal width* is *VL.long* **then** *Versicolor* (CF^3=0.487) (0.67, 0.681, 0.725) (0.59, 0.644, 0.692) $\boldsymbol{c}$ = 0.744, $\boldsymbol{s}$ = 0.007
R_4	**if** *Petal length* is *long*, and *Petal width* is *long* **then** *Virginica* (CF^3 = 0.883) (0.67, 0.931, 1) (0.59, 0.783, 1) $\boldsymbol{c}$ = 0.942, $\boldsymbol{s}$ = 0.119
R_5	**if** *Petal length* is *L.long*, and *Petal width* is *VL.short* **then** *Virginica* (CF^3 = 0.757) (0.67, 0.725, 0.931) (0.443, 0.512, 0.59) $\boldsymbol{c}$ = 0.879, $\boldsymbol{s}$ = 0.006

Tiếp theo chúng ta sẽ tối ưu tham số mờ gia tử cho bài toán *IRIS*2 áp dụng phương pháp sinh luật dựa trên hệ khoảng tương tự của thuật toán ***IFRG*2** trong ĐS2GT bằng ví dụ sau.

Ví dụ 3.2. Tối ưu tham số theo thuật toán sinh luật ***IFRG*2**. Áp dụng thuật toán ***FPO-SGA*** trên để tìm kiếm bộ tham số tối ưu cho bài toán phân lớp các loại hoa (*IRIS2*, đã đề cập trong Ví dụ 2.1) theo phương pháp sinh luật dựa trên hệ khoảng tương tự - ***IFRG*2**.

Các tham số chạy thuật toán sinh luật ***IFRG*2** gồm tiêu chuẩn sàng luật SR^3 = $\boldsymbol{c}.\boldsymbol{s}$, trọng số luật CF^3 và phương pháp lập luận *single-winner-rule*. Mã hóa bộ tham số gia tử của 4 thuộc tính trong mỗi cá thể của thuật toán ***FPO-SGA*** gồm ***PAR*** = { $fm_j(\boldsymbol{c}^-)$, $\mu_j(L)$, $k_j \mid j$ = 1,2 }, giới hạn các tham số mờ gia tử là $0.2 \le fm_j(\boldsymbol{c}^-), \mu_j(L) \le 0.8$ và $1 \le k_j \le 2$. Số thuộc tính của bài toán nhỏ nên chúng ta đặt độ dài luật tối đa đúng bằng số thuộc tính ($L = n = 2$). Các tham số chạy thuật toán tối ưu ***FPO-SGA*** α = 0.7, γ_{max} = 9, kích thước quần thể tại mỗi thế hệ N_p = 100, số thế hệ tiến hóa G_{max} = 150. Trọng số cho các thành phần trong hàm *fitness* là w_p = 0.9, w_n = 0.01, w_a = 0.09. Với tỷ lệ số mẫu trong các lớp cân bằng, áp dụng phương pháp sàng cân bằng chọn ra một hệ có M = 3 luật (mỗi lớp lấy ra 1 luật) để đánh giá *fitness*.

Kết quả bộ tham số mờ gia tử tối ưu cùng mức phân hoạch hệ khoảng tương tự cho bài toán thu được sau khi chạy thuật toán ***FPO-SGA*** trong Bảng 3.3. Hệ luật kết quả thể hiện trong Bảng 3.4, với số luật là f_n = 3 luật và độ dài trung bình rất nhỏ (f_a = 1), số lỗi phân lớp là 3/150 mẫu dữ liệu (đạt hiệu quả 98%).

Bảng 3.3: Các tham số gia tử tối ưu bằng thuật toán ***FPO-SGA*** cho bài toán *IRIS*

	Petal length	*Petal width*
$fm(\boldsymbol{c}^-)$	0.410	0.710
$fm(\boldsymbol{c}^+)$	0.590	0.290
$\mu(L)$	0.227	0.479
$\mu(V)$	0.773	0.521
k_j	1	1

Bảng 3.4: Danh sách các luật sinh bởi thuật toán ***IFRG2*** theo bộ tham số tối ưu trong bảng 3.3 cho bài toán *IRIS* (mỗi giá trị ngôn ngữ trong điều kiện luật được tính các tham số của hàm định lượng ngữ nghĩa)

R_1	**if** *PW* is *completely short* **then** *Setosa* ($\boldsymbol{c}$ = 1, $\boldsymbol{s}$ = 0.279, CF^3 = 1) (0, 0, 0.3699)
R_2	**if** *PW* is *short* **then** *Versicolor* ($\boldsymbol{c}$=0.75, $\boldsymbol{s}$=0.195, CF^3 =0.542) (0, 0.3699, 0.71)
R_3	**if** *PL* is *completely long* **then** *Virginica* ($\boldsymbol{c}$ = 0.861, $\boldsymbol{s}$ = 0.166, CF^3=0.722) (0.544, 1, 1)

So sánh kết quả của các Ví dụ 3.1, Ví dụ 3.2 và Ví dụ 3.3 với các phương pháp khác. Xét kết quả tốt nhất trong [50], gồm 8 luật với độ dài của mỗi luật có đủ 4 thuộc tính và hiệu quả phân lớp đạt 100%, tương đương với kết quả trong Ví dụ 3.2 (chỉ thực hiện mục tiêu hiệu quả phân lớp và số luật). Nhưng Ví dụ 3.3 đạt được hệ luật đơn giản hơn rất nhiều, số luật giảm 62.5% (3/8), độ dài trung bình hệ luật giảm 75% (1/4), cho dù tỷ lệ phân lớp đúng là 98%, giảm 2% do phương pháp tối ưu trong luận án phải thỏa hiệp giữa ba mục tiêu (3.6) trong khi phương pháp của [50] chỉ một mục tiêu là hiệu quả phân lớp. Điều này minh họa cho việc các mục tiêu trong (3.6) không thể đạt được đồng thời. Kết quả trong Ví dụ 3.1 có 5 luật đạt hiệu quả 99.33% nhưng chỉ áp dụng hai thuộc tính.

Trong Bảng 3.5 sau thể hiện kết quả sau khi tối ưu tham số tăng lên nhiều so với trước khi tối ưu. Ví dụ 2.6 và Ví dụ 3.1 đều áp dụng cùng một phương pháp sinh luật (thuật toán ***IFRG*1**) trên 2 thuộc tính, hiệu quả phân lớp (tỷ lệ % số mẫu phân lớp đúng) sau khi tối ưu tham số tăng lên 99.33% so với trước khi tối ưu là 96.67% và số luật giảm từ 6 xuống còn 5 luật. Ví dụ 2.3 và Ví dụ 3.3 đều áp dụng cùng một phương pháp sinh luật (thuật toán ***IFRG*2**) trên 2 thuộc tính, kết quả tăng lên rõ rệt sau khi tối ưu, tỷ lệ phân lớp đúng tăng lên 98% so với 97.33% trong khi số luật giảm một nửa (từ 6 xuống còn 3 luật). Những so sánh trên cho thấy ý nghĩa và tính hiệu quả của phương pháp tối ưu tham số.

Bảng 3.5: So sánh kết quả trước và sau khi tối ưu tham số đối với bài toán *IRIS*2

Phương pháp sinh luật	Chưa tối ưu tham số			Đã tối ưu tham số		
	Số luật	Độ dài hệ luật	Hiệu quả phân lớp	Số luật	Độ dài hệ luật	Hiệu quả phân lớp
Thuật toán ***IFRG*1**	6	2	96.67%	5	2	99.33%
Thuật toán ***IFRG*2**	6	1	97.33%	3	1	98.0%

3.2 Bài toán thiết kế tối ưu hệ luật mờ

3.2.1 Đặt bài toán

Hệ luật khởi đầu $\boldsymbol{S_0}$ sinh bởi ***IFRG*2** có chứa những dư thừa về số luật với số lượng rất lớn do phương pháp lấy tổ hợp các thuộc tính ở vế trái của luật. Trong Chương 2 đã trình bày phương pháp sàng để rút gọn hệ luật mờ dựa trên tiêu chuẩn đánh giá độ quan trọng của luật. Tuy nhiên, phương pháp sàng dựa trên tiêu chuẩn là độ tin cậy, độ hỗ trợ hoặc tích của chúng chưa hoàn toàn đảm bảo việc loại bỏ các luật có giá trị của tiêu chuẩn thấp là ít quan trọng. Hay nói cách khác, việc chúng ta giữ lại các luật có giá trị tiêu chuẩn cao nhất theo độ hỗ trợ và độ tin cậy chưa khẳng định cho các mục tiêu (1.6). Đây là phương pháp mang tính *heuristic*.

Như vậy việc áp dụng quá trình ***HAFRG*** (Hình vẽ 3.1) để sinh hệ luật mờ phân lớp đủ nhỏ gọn và đơn giản với hiệu quả cao khó có thể đạt được đồng thời.

Chúng ta áp dụng quá trình này để sinh hệ luật $\boldsymbol{S}^*$ với một số lượng các luật đủ lớn nhằm đem lại hiệu quả phân lớp cao, ta gọi $\boldsymbol{S}^*$ là hệ luật thô. Trong phần này sẽ thiết kế một phương pháp để tối ưu hệ luật mờ, tức là chọn lựa các luật trong hệ luật thô $\boldsymbol{S}^*$ để đạt mục tiêu (1.6).

Bài toán đặt ra là với hệ luật thô $\boldsymbol{S}^*$ sinh bởi quá trình ***HAFRG*** khi áp dụng thuật toán ***IFRG2***, tìm kiếm một hệ con các luật trong $\boldsymbol{S}^*$ theo mục tiêu (1.6). Ký hiệu $\boldsymbol{S}_{opt}$ là hệ luật tối ưu, ta có bài toán như sau:

Các mục tiêu:

$$f_p(\boldsymbol{S}) \rightarrow max,\ f_n(\boldsymbol{S}) \text{ và } f_a(\boldsymbol{S}) \rightarrow min, \tag{3.9}$$

Các ràng buộc:

i) $\boldsymbol{S}$ là một hệ con của hệ luật thô $\boldsymbol{S}^*$:

$$\boldsymbol{S} \subseteq \boldsymbol{S}^* \tag{3.10}$$

ii) Số luật không vượt quá giới hạn cho trước:

$$|\boldsymbol{S}| \leq N_{max}, \tag{3.11}$$

trong đó N_{max} là số luật tối đa cần chọn tối ưu.

Bài toán này khác với bài toán tối ưu tham số mờ gia tử, chúng đều có mục tiêu như nhau nhưng ở đây giả sử đã có một bộ tham số mờ gia tử (có thể đã được tối ưu), áp dụng quá trình ***HAFRG*** để sinh hệ luật $\boldsymbol{S}^*$ với số lượng đủ lớn. Khi đó $\boldsymbol{S}^*$ là không gian tìm kiếm của bài toán này.

Trong Mục 3.2.2 sẽ thiết kế thuật toán dựa trên giải thuật di truyền để tìm kiếm hệ luật tối ưu cho bài toán này.

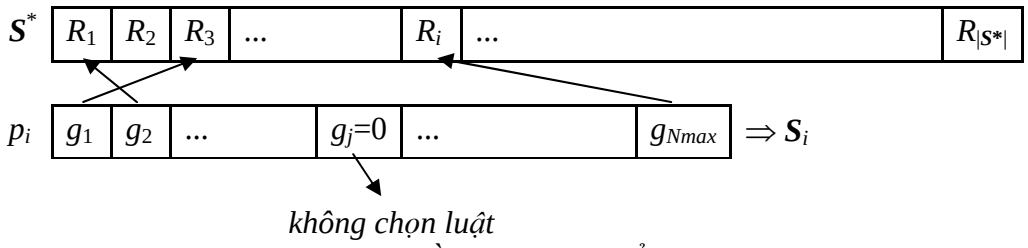
3.2.2 Tìm kiếm hệ luật tối ưu dựa trên giải thuật di truyền lai

Tương tự phương pháp giải bài toán tối ưu tham số mờ gia tử, các mục tiêu (3.9) sẽ được kết hợp theo trọng số trở thành hàm mục tiêu dạng (3.6) cho bài toán. Để áp dụng thuật toán di truyền lai SGA, chúng ta mã hóa mỗi cá thể tương ứng với một tập luật con của $\boldsymbol{S}^*$ dưới dạng số thực, ký hiệu p_i là cá thể thứ i. Mỗi gen của cá

thể p_i được mã hóa bởi một số thực trong đoạn [0,1] biểu diễn chỉ số của luật được chọn trong $\boldsymbol{S}^*$, ký hiệu g_j là gen thứ j, luật tương ứng được chọn có chỉ số trong tập $\boldsymbol{S}^*$ là $round(g_j.|\boldsymbol{S}^*|)$ (hàm $round(.)$ để làm tròn thành số nguyên). Trường hợp $round(g_j.|\boldsymbol{S}^*|) = 0$, tức không có luật được chọn (Hình vẽ 3.3). Phương pháp mã hóa này có thể hai gen g_{j1} và g_{j2} khác nhau ($j_1 \neq j_2$) nhưng cùng biểu diễn một chỉ số luật được chọn, khi đó chúng ta chỉ lấy một trong chúng. Hệ luật $\boldsymbol{S}_i$ tương ứng với cá thể p_i được xác định như sau:

$$\boldsymbol{S}_i = \{ R_k \mid g_j \in p_i, k = round(g_j.|\boldsymbol{S}^*|), k>0 \text{ và } R_k \in \boldsymbol{S}^* \}. \quad (3.12)$$

Rõ ràng hệ $\boldsymbol{S}_i \subseteq \boldsymbol{S}^*$, thỏa điều kiện (3.10) và (3.11).



Hình 3.3: Sơ đồ mã hóa cá thể chọn hệ luật

Các phép toán di truyền như chọn lọc, lai ghép, đột biến và thay thế sẽ áp dụng theo thuật toán tối ưu tham số mờ (Mục 3.1.2). Chúng ta có sơ đồ thuật toán như sau, ký hiệu là ***RBO-SGA*** (*rules base optimization - SGA*):

Inputs:

- Tập dữ liệu mẫu $\boldsymbol{D} = \{ (p_i; c_i) \mid i=1, ..., N \}$, $p_i = (d_{i,1}, ..., d_{i,n}) \in \boldsymbol{P}$, $c_i \in \boldsymbol{C} = \{C_1, ..., C_m\}$, n là số thuộc tính, m là số lớp, N là số mẫu dữ liệu;

- Hệ luật $\boldsymbol{S}^*$ sinh bởi quá trình ***HAFRG*** dựa trên bộ tham số mờ gia tử và mức phân hoạch được cho trước;

- Giới hạn ràng buộc số luật tối đa cần chọn tối ưu: N_{max};

- Trọng số cho các mục tiêu của hàm thích nghi (3.6): $0 < w_p, w_n, w_a < 1$, $w_p + w_n + w_a = 1$;

Outputs:

- Hệ luật tối ưu $\boldsymbol{S}_{opt}$;

Actions:

*Step*1) Khởi tạo một quần thể xuất phát gồm N_p cá thể Pop_0 = { $p_{0,1}$, $p_{0,2}$, ..., $p_{0,Np}$ } để tính tham số nhiệt ban đầu T_0 (ký hiệu $p_{k,i}$ là cá thể thứ i trong quần thể của thế hệ k, nó là mã hóa của tập chỉ số luật sẽ chọn từ $\boldsymbol{S}^*$, N_p là kích thước của quần thể tại mỗi thế hệ trong SGA);

*Step*2) Với mỗi cá thể $p_{0,i} \in Pop_0$, tính độ phù hợp $Fit(p_{0,i})$) theo công thức (3.6) dựa trên hệ luật $\boldsymbol{S}_{0,i}$ chọn từ $\boldsymbol{S}^*$ (3.12), và tính tham số nhiệt ban đầu:

$$T_0 = \sqrt{\frac{\sum_{i=1,...,N_p}\left(Fit(p_{0,i}) - \frac{1}{N_p}\sum_{j=1}^{N_p} Fit(p_{0,j})\right)^2}{N_p}}$$

*Step*3) Đặt k = 0. Lặp theo mỗi k cho đến khi $k = G_{max}$, Pop_k = {$p_{k,1}$, $p_{k,2}$, ..., $p_{k,Np}$} và thực hiện

3.*a*) Tính tham số nhiệt cho thế hệ k+1, $T_{k+1} = \alpha^k.T_k$, trong đó $\alpha < 1$ là hệ số giảm nhiệt độ (thường chọn 0.7);

3.*b*) Tạo quần thể mới Pop_{k+1} cho thế hệ k+1 như sau:

Lặp theo i cho đến khi $|Pop_{k+1}| = N_p$,

Chọn hai cặp cá thể bố mẹ $p, q \in Pop_k$ sử dụng phép chọn lọc *SGA_Selection*(Pop_k, T_{k+1}). Sau đó thực hiện các phép lai ghép, độ biến và thay thế trên cặp bố mẹ này bằng các phép *SGA_Crossover*, *SGA_Mutation*, *SGA_Replacement* để tạo cặp cá thể mới p', q' và đưa vào Pop_{k+1}.

Kết quả Pop_{k+1} = {$p_{k+1,1}$, $p_{k+1,2}$, ..., $p_{k+1,Np}$}

3.*c*) Với mỗi cá thể $p_{k,i} \in Pop_{k+1}$, tính độ phù hợp $Fit(p_{k,i})$ theo công thức (3.6) dựa trên hệ luật $\boldsymbol{S}_{0,i}$ chọn từ $\boldsymbol{S}^*$ theo công thức (3.12).

*Step*4) Trả về kết quả hệ luật ứng với cá thể tốt nhất ở thế hệ cuối, $\boldsymbol{S}_{opt}$.

End.

Các bước của thuật toán này về cơ bản giống với ***FPO-SGA***, ở đây chỉ khác tại *Step*2) và *Step*3.*c*), thay vì áp dụng quá trình ***HAFRG*** để sinh hệ luật thì chúng ta lại dựa vào mã hóa cá thể để xác định hệ luật con $\boldsymbol{S}$ từ hệ $\boldsymbol{S}^*$. Tính dừng và tính "*tinh hoa*" của thuật toán này tương tự thuật toán ***FPO-SGA*** (đã được xem xét trong Mục 3.1). Tập dữ liệu mẫu $\boldsymbol{D}$ được sử dụng để tính toán tỷ lệ số mẫu phân lớp đúng $f_p(\boldsymbol{S})$ đối với một hệ luật (công thức (3.6)).

Ví dụ 3.4. Áp dụng thuật toán ***RBO-SGA*** để tìm kiếm hệ luật mờ tối ưu cho bài toán phân lớp các loại rượu (*wine*, đã đề cập trong Ví dụ 2.6).

Bộ tham số mờ gia tử và mức phân hoạch k_j được tối ưu bởi thuật toán ***FPO-SGA*** trong Bảng 3.6. Quá trình sinh luật ***HAFRG*** áp dụng thuật toán ***IFRG2***, phương pháp sàng luật theo tiêu chuẩn $SR^3 = \boldsymbol{c.s}$, trọng số luật là $CF^3 = c_q - c_{q,2nd}$ và phương pháp lập luận là *single-winner-rule*. Hệ luật $\boldsymbol{S}^*$ sinh bởi quá trình ***HAFRG*** gồm 900 luật cho kết quả tỷ lệ phân lớp đúng 96.63%.

Bảng 3.6: Bảng tham số mờ gia tử cho bài toán *WINE*

Thuộc tính	$\boldsymbol{fm_j(c^-)}$	$\boldsymbol{fm_j(c^+)}$	$\boldsymbol{\mu_j(L)}$	$\boldsymbol{\mu_j(V)}$	$\boldsymbol{k_j}$
AL	0.512	0.488	0.538	0.462	2
MA	0.518	0.482	0.604	0.396	1
AS	0.456	0.544	0.56	0.44	2
AA	0.447	0.553	0.493	0.507	1
MG	0.544	0.456	0.466	0.534	1
TP	0.554	0.446	0.632	0.368	2
FL	0.555	0.445	0.511	0.489	1
NP	0.603	0.397	0.507	0.493	1
PR	0.399	0.601	0.435	0.565	1
CI	0.482	0.518	0.348	0.652	1
HU	0.46	0.54	0.361	0.639	2
OD	0.54	0.46	0.544	0.456	2
PL	0.316	0.684	0.387	0.613	1

Các tham số chạy thuật toán ***RBO-SGA*** gồm $\alpha = 0.7$, $\gamma_{max} = 9$, kích thước quần thể tại mỗi thế hệ $N_p = 500$, số thế hệ tiến hóa $G_{max} = 150$. Trọng số cho các

thành phần trong hàm mục tiêu (3.6) là w_p = 0.99, w_n = 0, w_a = 0.01. Với mong muốn hệ luật chọn tối ưu sẽ gồm đủ số lượng N_{max} luật, khi đó ta đặt w_n = 0. Chạy thuật toán này 5 lần với các kích thước tập luật cần tối ưu N_{max} khác nhau gồm 3, 4, 5, 6 và 7 luật. Kết quả được thể hiện trong Bảng 3.9 và so sánh với các phương pháp khác (ký hiệu "/" không thử nghiệm kết quả). Phương pháp [50] cho kết quả tốt nhất với hệ 3 luật và tỷ lệ đúng 100%, tiếp đến là phương pháp [10] với hệ 3 luật và tỷ lệ đúng 99.4%. Các phương pháp khác thấp hơn hoặc bằng kết quả của ***RBO-SGA***. Chẳng hạn so với phương pháp của Ishibuchi trong [43], hệ 3, 6, 7 luật cho kết quả tốt hơn và hệ 4, 5 luật cho kết quả bằng. Cụ thể trường hợp 3 luật cho kết quả 95.51% trong khi [43] chỉ đạt 93.3%, hay trường hợp 6 luật cho kết quả 100% nhưng [43] đạt 99.4%, trường hợp 7 luật có độ dài trung bình của hệ thấp hơn (1.71 so với 2 trong [43]). Hệ 6 luật của phương pháp ***RBO-SGA*** trong Bảng 3.7 được thể hiện trong Bảng 3.8 cùng các tham số của các giá trị ngôn ngữ trong mỗi luật. Điều này minh họa cho thuật toán ***RBO-SGA*** chọn được hệ luật để tăng hiệu năng phân lớp.

Bảng 3.7: Kết quả chạy ***RBO-SGA*** và so sánh với các phương pháp *FRBCS* khác dựa trên tập mờ

Phương pháp	***Số luật***	***Độ dài luật***	***Tỷ lệ phân lớp đúng***
E. Lughofer và cộng sự [59]	9	/	94.38
Yuan-Long Hou và cộng sự [40]	4	5	98.3
E. G. Mansoori và cộng sự [60]	124	/	98.31
J. Abonyi và cộng sự [10]	3	5	99.4
A. Khotanzad, E. Zhou [50]	**3**	**6**	**100**
H. Ishibuchi, T. Yamamoto [43]	3	1.33	93.3
	4	1.5	97.2
	5	1.6	98.3
	6	2	99.4
	7	**2**	**100**
Phương pháp *RBO-SGA*	3	1	95.51
	4	1.25	97.19
	5	2	98.31
	6	**2.33**	**100**
	7	**1.71**	**100**

Bảng 3.8: Hệ gồm 6 luật mờ đạt tỷ lệ số mẫu phân lớp đúng 100% trên *WINE*

R_1	**if** *MG* is *small* and *NP* is *small* and *PL* is *large* **then** *class*_1 (CF^3=0.964) (0,0.297,0.603) (0,0.29,0.544) (0.316,0.581,1)
R_2	**if** *FL* is *small* and *CI* is *completely small* **then** *class*_2 (CF^3=0.899) (0,0.271,0.555) (0,0,0.314)
R_3	**if** *MG* is *medium* and *FL* is *medium* and *PR* is *medium* **then** *class*_1 (CF^3=0.912) (0.29,0.544,0.756) (0.271,0.555,0.782) (0.225,0.399,0.66)
R_4	**if** *FL* is *completely small* and *PR* is *small* **then** *class*_3 (CF^3=0.902) (0,0,0.271) (0,0.225,0.399)
R_5	**if** *AS* is *L.large* and *FL* is *completely small* **then** *class*_3 (CF^3=0.898) (0.456,0.59,0.76) (0,0,0.271)
R_6	**if** *MG* is *small* and *PL* is *small* **then** *class*_2 (CF^3=0.121) (0,0.29,0.544) (0,0.194,0.316)

3.3 Kết luận Chương 3

Chương này đã giới thiệu phương pháp tối ưu tham số mờ gia tử cho mô hình sinh luật để phân lớp của hai thuật toán ***IFRG*1** và ***IFRG*2**. Bằng cách sử dụng giải thuật di truyền (GA) được nhúng thêm tham số nhiệt *T* lấy từ thuật toán mô phỏng tôi luyện, tác động vào các toán tử gen làm tăng tốc hội tụ của GA (SGA), chúng tôi đã thiết kế thuật toán ***FPO-SGA*** để tối tham số mờ. Kết quả thử nghiệm trong các Ví dụ 3.1 và 3.3 cho thấy hiệu quả phân lớp của hệ luật sau khi đã được tối ưu tham số tăng lên nhiều so với ban đầu (chưa tối ưu tham số trong Ví dụ 2.3 và 2.5).

Một thuật toán khác cũng được đề xuất trong chương này để tối ưu hệ luật mờ, gọi là ***RBO-SGA***. Với bộ tham số cho trước (có thể đã được tối ưu), sử dụng quá trình sinh hệ luật phân lớp bằng một trong hai thuật toán ***IFRG*1** và ***IFRG*2** ta thu được hệ luật. Tuy nhiên, hệ luật này cần được tối ưu một lần nữa với mong muốn đạt được hiệu quả cao, số luật ít và số điều kiện tham gia trong vế trái mỗi luật là nhỏ (mục tiêu (3.9)). Thuật toán cũng được thiết kế dựa trên thuật toán di truyền lai SGA tương tự ***FPO-SGA***. Kết quả thử nghiệm trong Ví dụ 3.4 cho thấy hiệu quả tăng lên rõ rệt của thuật toán tối ưu này (Bảng 3.8).

CHƯƠNG 4
MÔ PHỎNG BẰNG MÁY TÍNH TRÊN MỘT SỐ BÀI TOÁN PHÂN LỚP

4.1 Phương pháp mô phỏng cho bài toán phân lớp

Mô hình xây dựng hệ luật mờ phân lớp dựa trên ĐSGT được đề xuất với mục tiêu xây dựng hệ luật mờ để ứng dụng phân lớp cho các mẫu dữ liệu sao cho hệ luật phải có hiệu quả phân lớp cao, càng đơn giản, dễ hiểu và tường minh đối với người dùng càng tốt. Trong các chương trước chúng ta đã thực hiện một số ví dụ để minh họa cho phương pháp, trong chương này sẽ tập trung ứng dụng mô hình vào một số bài toán khá thông dụng. Các bài toán với tập dữ liệu mẫu được xây dựng bởi nhiều nhà khoa học và công bố công khai tại [76] của Đại học California tại Irvine. Các bài toán ở đây chủ yếu về lĩnh vực khai phá dữ liệu như phân lớp (*classification*), phân cụm (*clustering*),… và đã được rất nhiều tác giả nghiên cứu sử dụng để thử nghiệm [10], [17], [20], [23], [30]-[33], [40]-[47], [50], [53], [56], [59], [60], [74].

Các phương pháp ứng dụng thử nghiệm mô hình được nhiều tác giả áp dụng đó là *k-folds cross validation*. Chúng ta chia ngẫu nhiên tập dữ liệu mẫu của bài toán thành k phần bằng nhau, sử dụng một phần để thẩm định (*TEST*) mô hình còn lại (k-1) phần để sinh hệ luật (*TRAIN*). Phương pháp thử nghiệm này nhằm khắc phục nhược điểm của các mô hình đó là hiện tượng quá khớp (*overfit*), tức là mô hình sẽ làm việc tốt đối với tập dữ liệu mẫu dùng để xây dựng trong khi cho kết quả rất tồi đối với các mẫu dữ liệu mới. Phương pháp thử nghiệm này sẽ được lặp lại k lần, mỗi lần lấy ra lần lượt một phần trong số k phần để kiểm tra.

Hầu hết các tác giả áp dụng phương pháp này với $k = 10$, $k = 5$ và $k = 2$, để tiện về sau chúng ta ký hiệu các trường hợp này là *CV*10 (10% số mẫu kiểm tra), *CV*20 (20% số mẫu kiểm tra) và *CV*50 (50% số mẫu kiểm tra). Trong luận án, với mong muốn kiểm chứng sự ổn định của mô hình đối với các bài toán ứng dụng, mỗi trường hợp thử nghiệm sẽ thực hiện nhiều lần với các k-phần chia ngẫu nhiên.

Chẳng hạn trường hợp *CV*10 với số lần thử nghiệm là 10, do đó số lần chạy để thử nghiệm mô hình là 10×10 = 100 lần. Ký hiệu *n*×*CV*10 cho *n* lần lặp thử nghiệm trong trường hợp *CV*10, tương tự với *n*×*CV*20 và *n*×*CV*50.

Ngoài ra, phương pháp thử nghiệm lấy một mẫu dữ liệu ra để kiểm tra, còn lại các mẫu dùng để xây dựng mô hình, được gọi là *Leave-One-Out*, ký hiệu *LV*1. Phương pháp này sẽ được lặp lại theo lần lượt mỗi mẫu được lấy ra để kiểm tra, như vậy số lần lặp để thử nghiệm đúng bằng số mẫu. Tuy nhiên phương pháp *LV*1 sẽ phải lặp lại rất nhiều lần nếu tập dữ liệu mẫu có kích thước lớn, do đó chúng ta sẽ không áp dụng cho những bài toán có tập dữ liệu mẫu lớn.

Bây giờ chúng ta sẽ xây dựng quy trình ứng dụng thử nghiệm mô hình trong các bài toán. Mỗi bài toán với tập dữ liệu được cho, trước hết chúng ta sẽ áp dụng thuật toán ***FPO-SGA*** để tìm kiếm tối ưu bộ tham số mờ gia tử cũng như mức phân hoạch k_j dựa trên ĐSGT hoặc ĐS2GT. Sử dụng bộ tham số tối ưu này ($\boldsymbol{PAR_{opt}}$), chúng ta thiết kế hai sơ đồ ứng dụng thử nghiệm. Thứ nhất, áp dụng quá trình ***HAFRG*** để sinh hệ luật đủ nhỏ bằng phép sàng hoặc hợp các luật mờ và đánh giá kết quả. Sơ đồ này áp dụng cho cả hai phương pháp sinh luật bằng thuật toán ***IFRG*1** và ***IFRG*2**, ký hiệu sơ đồ này là ***No-RBO***. Thứ hai, quá trình ***HAFRG*** cũng được dùng để sinh tập luật với số lượng đủ lớn và thực hiện tìm kiếm tối ưu hệ luật trên tập luật này bằng thuật toán ***RBO-SGA***, ký hiệu sơ đồ này là ***RBO-SGA*** và chỉ áp dụng đối với thuật toán sinh luật ***IFRG*2**.

Đánh giá kết quả gồm các yếu tố của hệ luật như sau:

+ $\boldsymbol{P_{Nr}}$: là số luật của hệ thu được,

+ $\boldsymbol{P_{Rl}}$: là độ dài trung bình của luật trong hệ luật, tức tổng số điều kiện trong vế trái các luật chia cho số luật,

+ $\boldsymbol{P_{Tr}}$: là tỷ lệ số mẫu phân lớp đúng trên tập huấn luyện (*TRAIN*). Thông thường các kết quả nghiên cứu được công bố ít quan tâm đến đánh giá này, tuy nhiên trong luận án có đưa ra kết quả này nhằm tham khảo trong quá trình thử nghiệm.

+ $\boldsymbol{P_{Te}}$: là tỷ lệ số mẫu phân lớp đúng trên tập kiểm tra (*TEST*).

Đối với mỗi bài toán, chạy nhiều lần thử nghiệm nên các kết quả này sẽ được tính trung bình trên các lần chạy đó.

Các kết quả thử nghiệm trong luận án được so sánh với các kết quả nghiên cứu theo mô hình hệ mờ dạng luật dựa trên tập mờ. Sự so sánh này nhằm đảm bảo sự tương ứng về các yếu tố hiệu quả phân lớp, độ phức tạp của hệ được thể hiện ở số luật và độ dài luật.

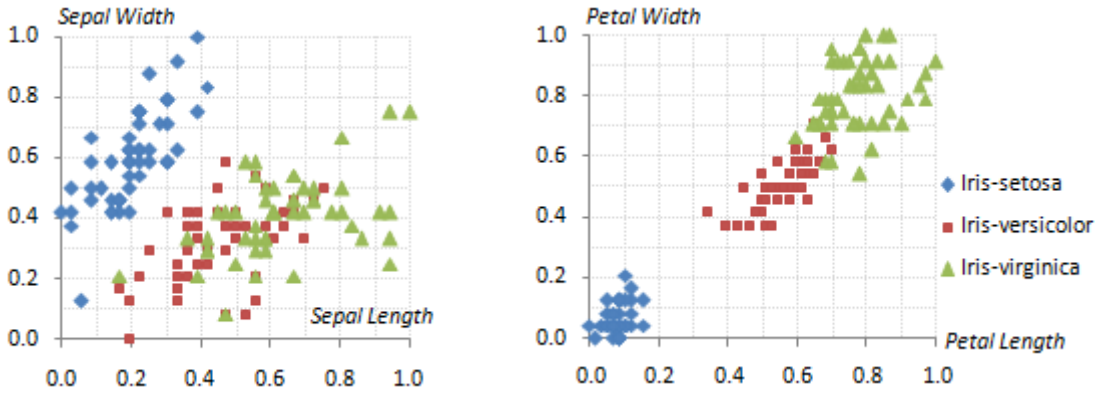
Tiếp theo, chúng ta sẽ ứng dụng thử nghiệm mô hình vào 4 bài toán gồm phân lớp các loại hoa (*IRIS*), phân lớp các loại rượu (*WINE*), phân lớp các loại kính (*GLASS*) và phân lớp các loại men sinh học (*YEAST*).

4.2 Bài toán phân lớp các loại hoa - IRIS

Bài toán này đã được đề cập trong Ví dụ 2.1 của Chương 2. Trong các ví dụ trước chúng ta sử dụng hai thuộc tính là độ dài cánh hoa (*Petal length*) và độ rộng cánh hoa (*Petal width*) để minh họa phương pháp sinh luật cũng như phương pháp tối ưu tham số. Trong phần này, bằng các phương pháp thử nghiệm khác nhau để khẳng định hiệu quả của phương pháp sinh luật, phương pháp tối ưu tham số và phương pháp tìm kiếm hệ luật mờ tối ưu đối với bài toán.

Hình vẽ 4.1 thể hiện sự phân bố dữ liệu của tập mẫu giữa các lớp theo từng cặp thuộc tính. Cặp thuộc tính *Petal length* và *Petal width* (Hình 4.1*b*) có tính quyết định đến phân lớp rõ rệt hơn so với hai thuộc tính còn lại là độ dài đài hoa (*Sepal length*) và độ rộng đài hoa (*Sepal width*) (Hình 4.1*a*). Như vậy trong một luật có thể chỉ cần một vài thuộc tính đủ để quyết định phân lớp, chẳng hạn chỉ cần một thuộc tính *Petal length* hoặc *Petal width* đã chắc chắn quyết định được việc phân lớp cho *Setosa*.

Chúng ta sẽ ứng dụng cả hai phương pháp sinh luật bằng thuật toán ***IFRG*1** và ***IFRG*2** cho bài toán này.



(a) (b)

Hình 4.1: Sơ đồ phân bố dữ liệu giữa các lớp của bài toán *IRIS*

4.2.1 Áp dụng thuật toán sinh luật *IFRG*1

Với thuật toán ***IFRG*1** không hạn chế về số gia tử, chúng ta sử dụng ĐSGT gồm 4 gia tử $\boldsymbol{H}^- = \{L, P\}$ và $\boldsymbol{H}^+ = \{M, V\}$. Bài toán *IRIS* có 4 thuộc tính nên bộ tham số cần tối ưu sẽ là $\boldsymbol{PAR} = \{fm_j(L), \mu_j(P), \mu_j(L), \mu_j(M), \mu_j(V), k_j : j = 1,...,4\}$. Áp dụng thuật toán ***FPO-SGA*** để tìm kiếm tối ưu bộ tham số này, đặt trọng số hàm *fitness* là $w_p = 0.7$, $w_n = 0.3$ và $w_a = 0$ (chọn $w_a = 0$ vì thuật toán ***IFRG*1** sinh hệ luật có độ dài như nhau và bằng số thuộc tính), ngưỡng hợp các luật là $\theta_{itg} = 0.1$. Giới hạn các tham số gồm $0.1 < fm_j(L), \mu_j(P), \mu_j(L), \mu_j(M), \mu_j(V) < 0.9$ và $1 \le k_j \le 5$. Kích thước quần thể trong mỗi thế hệ là $N_p = 500$, số thế hệ tiến hóa $G_{max} = 150$. Kết quả bộ tham số thu được trong Bảng 4.1, hệ luật sinh gồm 7 luật (Bảng 4.2) với tỷ lệ phân lớp đúng đạt 100%. Ở đây không áp dụng trọng số luật (tức là $CF^0 = 1$) và phương pháp lập luận là *single-winner-rule*.

Bảng 4.1: Các tham số gia tử tối ưu của thuật toán ***FPO-SGA*** cho bài toán *IRIS*

	Sepal length	*Sepal width*	*Petal length*	*Petal width*
$fm(\boldsymbol{c}^-)$	0.825	0.869	0.671	0.279
$fm(\boldsymbol{c}^+)$	0.175	0.131	0.329	0.721
$\mu(L)$	0.295	0.141	0.172	0.106
$\mu(P)$	0.188	0.278	0.252	0.138
$\mu(M)$	0.281	0.369	0.259	0.325
$\mu(V)$	0.236	0.212	0.317	0.431
k_j	3	3	4	3

Bảng 4.2: Danh sách các luật kết quả của thuật toán ***FPO-SGA*** cho bài toán *IRIS*

R_1	**if** *SL* is *short*, *SW* is *short*, *PL* is *V.short*, and *PW* is *short* **then** *Setosa* (0, 0.427, 0.825) (0, 0.505, 0.869) (0, 0.123, 0.213) (0, 0.211, 0.279) ***c*** = 1, ***s*** = 0.056
R_2	**if** *SL* is *M.short*, *SW* is *long*, *PL* is *VMV.short*, and *PW* is *short* **then** *Setosa* (0.195, 0.315, 0.427) (0.869, 0.924, 1) (0.067, 0.077, 0.085) (0, 0.211, 0.279) ***c*** = 1, ***s*** = 0.001
R_3	**if** *SL* is *short*, *SW* is *short*, *PL* is *short*, and *PW* is *long* **then** *Versicolor* (0,0.427,0.825) (0,0.505,0.869) (0,0.386,0.671) (0.279,0.455,1) ***c***=0.867, ***s***=0.097
R_4	**if** *SL* is *short*, *SW* is *M.short*, *PL* is *PP.long*, and *PW* is *PL.long* **then** *Versicolor* (0,0.427,0.825) (0.184,0.371,0.505) (0.671,0.685,0.704) (0.59,0.614,0.691) ***c***=0.819, ***s***=0.003
R_5	**if** *SL* is *short*, *SW* is *short*, *PL* is *long*, and *PW* is *long* **then** *Virginica* (0,0.427,0.825) (0,0.505,0.869) (0.671,0.81,1) (0.279,0.455,1) ***c***=0.895, ***s***=0.052
R_6	**if** *SL* is *long*, *SW* is *short*, *PL* is *long*, and *PW* is *long* **then** *Virginica* (0.825,0.91,1) (0,0.505,0.869) (0.671,0.81,1) (0.279,0.455,1) ***c***=1, ***s***=0.009
R_7	**if** *SL* is *short*, *SW* is *M.short*, *PL* is *P.short*, and *PW* is *L.long* **then** *Virginica* (0,0.427,0.825)(0.184,0.371,0.505)(0.556,0.622,0.671)(0.59,0.647,0.824) ***c***=0.615, ***s***=0.01

Tiếp theo sử dụng bộ tham số đã tối ưu ở trên để ứng dụng thử nghiệm bài toán trong ba trường hợp *LV*1, *CV*10 và *CV*50. Đối với *LV*1, kết quả hệ luật tại mỗi lần chạy đều có 7 luật, số lỗi phân lớp trên tập sinh luật là 0 và trên tập kiểm tra là 2 tại hai mẫu kiểm tra (mẫu 77 và 83). Chạy 10 lần *CV*10, hệ luật trong các lần chạy từ 6 đến 7 luật, số lỗi trên tập sinh luật là 0 và trên tập kiểm tra từ 0 đến 2. Trong *CV*50 chạy 50 lần, các lần chạy với hệ luật từ 5 đến 7 luật, số lỗi trên tập sinh luật từ 0 đến 3 và số lỗi trên tập kiểm tra từ 0 đến 8. Đánh giá trung bình các kết quả thể hiện trong Bảng 4.3 (dấu "/" không có kết quả) và so sánh với các phương pháp khác thì thuật toán ***IFRG*1** đạt hiệu quả phân lớp khá tốt trên tập kiểm tra. Chẳng hạn, trong *LV*1 cao hơn các phương pháp và bằng [50], đối với *CV*50 bằng [43], thấp hơn [17], [56] và cao hơn [60], [50].

Bảng 4.3: Kết quả của thuật toán ***IFRG*1** và so sánh với các phương pháp *FRBCS* khác trên bài toán *IRIS*

Phương pháp	$\boldsymbol{P_{Nr}}$	$\boldsymbol{P_{Tr}}$ (%)	$\boldsymbol{P_{Te}}$ (%)
Leave-one-out (*LV*1)			
E. G. Mansoori và cộng sự [60]	9	/	76.0

M. Grabisch, F. Dispot [26]	/	/	94.33
Bayes Classifier	/	/	97.33
X.G. Chang, J.H. Lilly [16]	4.75	/	98
A. Khotanzad, E. Zhou [50]	5.4	/	98.67
Thuật toán *IFRG*1	**7**	**100**	**98.67**
10 *folds cross-validation* (*CV*10)			
Thuật toán *IFRG*1	**6.96**	**100**	**98.67**
2 *folds cross-validation* (*CV*50)			
E. G. Mansoori và cộng sự [60]	9	/	77.87
A. Khotanzad, E. Zhou [50]	3.5	/	95.5
C.C. Chen [17]	4.73	/	96.8
H. Ishibuchi, T. Yamamoto [43]	3	/	96.4
C.Y. Lee và cộng sự [56]	/	/	98.0
Thuật toán *IFRG*1	**6.59**	**99.65**	**96.39**

4.2.2 Áp dụng thuật toán sinh luật *IFRG*2

Phương pháp sinh luật dựa trên phân hoạch hệ khoảng tương tự trong ĐS2GT, được thiết kế bởi thuật toán ***IFRG*2**. Áp dụng tiêu chuẩn là $SR^3 = \boldsymbol{c.s}$ để sàng rút gọn hệ luật, sử dụng trọng số luật CF^3 và phương pháp lập luận *single-winner-rule*. Trước hết, tối ưu tham số mờ gia tử của mô hình áp dụng cho bài toán bằng thuật toán ***FPO-SGA***, các tham số thực hiện thuật toán này gồm $N_p = 300$, $G_{max} = 150$, $0.2 \le fm_j(\boldsymbol{c}^-)$, $\mu_j(L) \le 0.8$, $1 \le k_j \le 2$ (j=1,...,4). Số thuộc tính của bài toán nhỏ nên chúng ta đặt độ dài luật tối đa đúng bằng số thuộc tính $L = n = 4$. Trọng số cho các thành phần trong hàm *fitness* là $w_p = 0.99$, $w_n = 0$, $w_a = 0.01$ (trong thuật toán tối ưu tham số sử dụng phương pháp sinh luật bằng thuật toán ***IFRG*2** và sàng luật để rút gọn nên số luật trong mỗi kết quả chạy là như nhau, vì vậy chọn $w_n = 0$). Với tỷ lệ số mẫu trong các lớp cân bằng, áp dụng phương pháp sàng cân bằng để chọn ra $N_s = 5$ hệ luật $\boldsymbol{Set}(5) = \{S_1, S_2, S_3, S_4, S_5\}$, hệ S_i gồm $M_i = i.3$ luật (mỗi lớp lấy ra i luật), để đánh giá các tham số hàm mục tiêu (3.6). Kết quả bộ tham số gia tử và mức phân hoạch mờ tối ưu $\boldsymbol{PAR}_{iris}$ thu được trong Bảng 4.4 (để ý rằng $fm(\boldsymbol{c}^+) = 1 - fm(\boldsymbol{c}^-)$, $\mu(V) = 1 - \mu(L)$).

Bảng 4.4: Kết quả tham số tối ưu (***PAR***$_{iris}$) theo thuật toán ***IFRG2*** cho bài toán *IRIS*

	Sepal length	***Sepal width***	***Petal length***	***Petal width***
$fm_j(\bar{\boldsymbol{c}})$	0.243845	0.584775	0.662068	0.401447
$\mu_j(L)$	0.648815	0.498686	0.547278	0.314544
k_j	1	1	1	1

Với bộ tham số đã tối ưu trong Bảng 4.5, chúng ta thực hiện 2 sơ đồ thử nghiệm ***No-RBO*** và ***RBO-SGA***. Trong sơ đồ ***No-RBO***, áp dụng quá trình sinh luật ***HARG*** gồm thuật toán ***IFRG2*** và phương pháp sàng để xây dựng hệ luật với số luật là 6 (mỗi lớp sàng lấy ra 2 luật), $\boldsymbol{S_6}$ = ***HAFRG***(***PAR***$_{iris}$, ***IFRG2***, 6). Đánh giá các yếu tố của hệ luật $\boldsymbol{S_6}$ này gồm $\boldsymbol{P_{Nr}}$, $\boldsymbol{P_{Rl}}$, $\boldsymbol{P_{Tr}}$ và $\boldsymbol{P_{Te}}$.

Sơ đồ ***RBO-SGA*** áp dụng quá trình sinh luật trên để sinh tập luật $\boldsymbol{S_{300}}$ = ***HAFRG***(***PAR***$_{iris}$, ***IFRG2***, 300) (300 luật). Tiếp theo sử dụng thuật toán ***RBO-SGA*** để tìm kiếm tối ưu hệ luật trong tập luật $\boldsymbol{S_{300}}$ này. Cả hai sơ đồ này được thực hiện với các trường hợp thử nghiệm *LV*1, 10.*CV*10, 20.*CV*20 và 50.*CV*50. Các tham số thực hiện tối ưu hệ luật gồm N_p = 200, G_{max} = 150, trọng số các mục tiêu hàm *fitness* w_p = 0.99, w_n = 0.009, w_a = 0.001. Số luật tối đa cần tối ưu là N_{max} = 7. Kết quả của 2 sơ đồ trong 4 trường hợp thử nghiệm thể hiện Bảng 4.5 và so sánh với các phương pháp khác (ký hiệu "/" không có kết quả thử nghiệm). Rõ ràng kết quả tối ưu hệ luật tốt hơn cả về số luật và hiệu quả phân lớp, trong khi độ dài hệ luật tăng không nhiều.

So sánh kết quả có áp dụng tối ưu hệ luật (***RBO-SGA***) với các phương pháp khác (Bảng 4.5), tỷ lệ phân lớp đúng trên tập kiểm tra (P_{Te}) của luận án đều tốt hơn trong cả 4 trường hợp thử nghiệm. Độ dài trung bình của các luật nhỏ hơn và số lượng trung bình các luật trong các lần thử nghiệm cũng nhỏ hơn, do đó thể hiện hệ luật kết quả ***RBO-SGA*** đơn giản hơn, dễ hiểu và tường minh hơn đối với người dùng. Hơn nữa, số lần thử nghiệm trong mỗi trường hợp của luận án là khá lớn (100 lần chạy), cho thấy sự ổn định của phương pháp trong ứng dụng. Kết quả trong [60] thấp do tác giả không sử dụng phương pháp tối ưu hệ luật, nhưng nếu so sánh với kết quả không tối ưu hệ luật (***No-RBO***) của luận án thì cũng thấp hơn nhiều, chẳng

hạn trường hợp *CV*50 [60] có $\boldsymbol{P_{Nr}}$ = 9 và $\boldsymbol{P_{Te}}$ = 77.87%, trong khi của luận án đạt $\boldsymbol{P_{Nr}}$ = 6 và $\boldsymbol{P_{Te}}$ = 96.5%.

Bảng 4.5: Kết quả thử nghiệm của bài toán *IRIS* trên hai sơ đồ không tối ưu và có tối ưu hệ luật, và so sánh với các phương pháp *FRBCS* khác

Phương pháp	$\boldsymbol{P_{Nr}}$	$\boldsymbol{P_{Rl}}$	$\boldsymbol{P_{Tr}}$ (%)	$\boldsymbol{P_{Te}}$ (%)
Leave-one-out (*LV*1)				
E. G. Mansoori và cộng sự [60]	9	/	/	76.0
A. Khotanzad, E. Zhou [50]	5.4	4	/	98.67
Sơ đồ ***No-RBO***	**6**	**1**	**97.79**	**96.67**
10-*folds cross validation* (*CV*10)				
S.M. Fakhrahmad và cộng sự [23]	/	/	/	98.3
Sơ đồ ***No-RBO***	**6**	**1**	**97.51**	**97.07**
Sơ đồ ***RBO-SGA***	**5.71**	**1.68**	**99.26**	**98.0**
5-*folds cross validation* (*CV*20)				
Li-Hui Wang và cộng sự [77]	8.85	/	/	96.7
I.E. El-Semman và cộng sự [74]	/	/	/	98.0
Sơ đồ ***No-RBO***	**6**	**1**	**97.09**	**97.7**
Sơ đồ ***RBO-SGA***	**5.78**	**1.67**	**99.31**	**98.90**
2-*folds cross validation* (*CV*50)				
E. G. Mansoori và cộng sự [60]	9	/	/	77.87
A. Khotanzad, E. Zhou [50]	3.5	/	/	95.5
H. Ishibuchi, T. Yamamoto [43]	3	2	/	96.4
C.C. Chen [17]	4.72	/	98.87	96.8
C.Y. Lee và cộng sự [56]	/	2	/	98.0
Sơ đồ ***No-RBO***	**6**	**1**	**96.68**	**96.5**
Sơ đồ ***RBO-SGA***	**5.78**	**1.7**	**99.67**	**98.75**

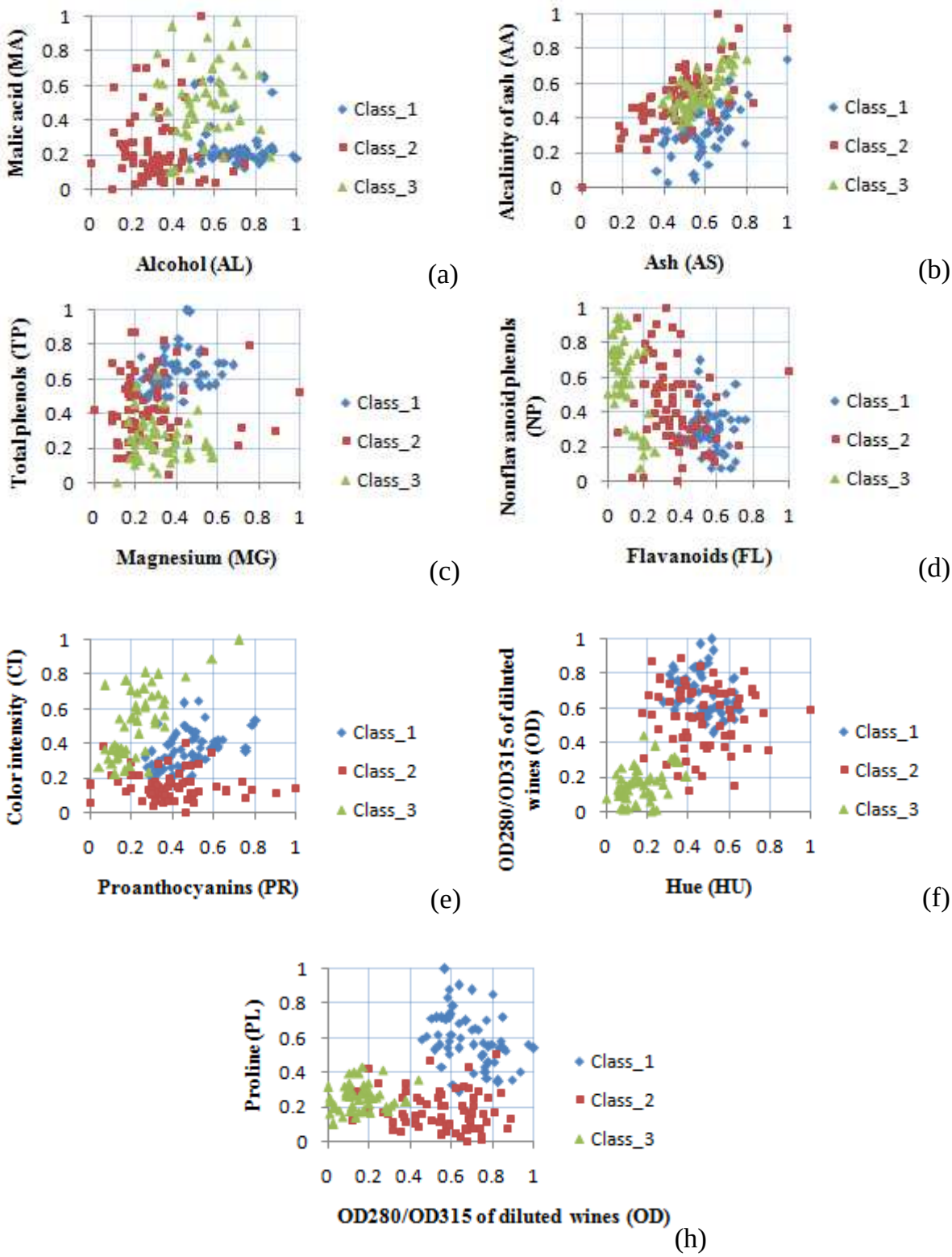
Mặt khác, hầu hết các tác giả chỉ đưa ra kết quả đánh giá trên tập kiểm tra mà không có kết quả trên tập huấn luyện. Tuy nhiên luận án có đưa ra kết quả này và cho thấy hiệu quả phân lớp trên tập huấn luyện ngày càng tăng khi kích thước của tập huấn luyện giảm, hiệu quả trên tập kiểm tra giảm khi kích thước tập kiểm tra tăng theo các phương pháp thử nghiệm khác nhau. Điều này khá tự nhiên, vì khi giới hạn kích thước tập luật để tối ưu là như nhau thì tập dữ liệu mẫu để sinh luật nhỏ dẫn đến tỷ lệ phân lớp đúng trên tập đó sẽ cao và tương tự đối với tập kiểm tra. Thật vậy, nó cũng đúng cho kết quả của các phương pháp khác.

4.3 Bài toán phân lớp các loại rượu - WINE

Bài toán phân lớp các loại rượu (*WINE*) đã được đề cập trong Ví dụ 2.6 của Chương 2, tập dữ liệu gồm n=13 thuộc tính với số lượng mẫu 178, có 3 loại rượu ký hiệu là *Class*_1, *Class*_2 và *Class*_3. Sơ đồ phân bố các dữ liệu mẫu theo từng cặp thuộc tính trên 3 lớp được thể hiện trong các hình vẽ của Hình 4.2, thuộc tính *OD*280/*OD*315 *of diluted wines* (*OD*) được thể hiện cả trong Hình 4.2*f* và 4.2*h* vì thuộc tính lẻ ra *Proline* cần được kết hợp để thể hiện dưới dạng sơ đồ hai chiều. Dữ liệu trên các lớp chồng chéo lên nhau khá nhiều, trực quan ta thấy thuộc tính *Flavanoids* (*FL*) có sự tách biệt dữ liệu lớn nhất giữa các lớp, trong khi cặp thuộc tính *Ash* (*AS*) và *Alcalinity of ash* (*AA*) hoặc thuộc tính *Magnesium* (*MG*) có dữ liệu ở các lớp chồng lên nhau khá dày đặc. Điều này cho thấy thế mạnh quyết định đến việc phân lớp các loại rượu của mỗi thuộc tính là khác nhau, và phương pháp của luận án cho phép loại bỏ các thuộc tính ít quyết định đến phân lớp trong một luật. Hơn nữa, với số thuộc tính quá nhiều và nếu không được rút gọn vế trái luật thì hệ luật sinh ra sẽ rất phức tạp, chứa nhiều các điều kiện của thuộc tính dư thừa trong các luật. Luận án sẽ áp dụng phương pháp sinh luật bằng thuật toán ***IFRG2*** để khắc phục điều này.

Theo quy trình thử nghiệm, trước hết chúng ta chạy thuật toán ***FPO-SGA*** để tối ưu tham số mờ gia tử cho bài toán. Sử dụng phương pháp sinh luật dựa trên hệ phân hoạch các khoảng tương tự trong ĐS2GT của miền các thuộc tính (thuật toán ***IFRG2***) và phương pháp sàng với tiêu chuẩn $SR^3 = \boldsymbol{c.s}$ để rút gọn hệ luật, ở đây tập dữ liệu mẫu có số mẫu trong các lớp không cân bằng nhưng tỷ lệ chênh lệch không quá lớn (59/71/48) nên chúng tôi vẫn sử dụng phương pháp sàng cân bằng. Các tham số chạy thuật toán tối ưu ***FPO-SGA*** gồm kích thước quần thể N_p = 300 cá thể, số thế hệ tiến hóa G_{max} = 150, ràng buộc các tham số là $0.2 \le fm(\boldsymbol{c}^-), \mu(L) \le 0.8$, $1 \le k_j \le 2$ (j=1,...,13), trọng số các mục tiêu hàm *fitness* là w_p = 0.99, w_n = 0 và w_a = 0.01. Luận án áp dụng N_s = 5 hệ luật $\{S_i : |S_i| = i.3, i=1,...,5\}$ sinh bởi quá trình ***HAFRG*** để đánh giá các mục tiêu và tính giá trị hàm *fitness* (công thức (3.6)), trong

đó áp dụng phương pháp lập luận *single-winner-rule*, trọng số luật CF^3. Kết quả tham số mờ gia tử và mức phân hoạch k_j của các thuộc tính thể hiện trong Bảng 4.6.



Hình 4.2: Sơ đồ phân bố dữ liệu giữa các lớp của bài toán *WINE*

Bảng 4.6: Kết quả tối ưu tham số mờ gia tử (***PAR***$_{wine}$) theo thuật toán ***IFRG2*** của bài toán *WINE*

Thuộc tính	$fm_j(c^-)$	$fm_j(c^+)$	$\mu_j(L)$	$\mu_j(V)$	k_j
AL	0.652451	0.347549	0.688971	0.311029	2
MA	0.316883	0.683117	0.582869	0.417131	2
AS	0.465903	0.534097	0.363529	0.636471	1
AA	0.431044	0.568956	0.510630	0.48937	1
MG	0.669737	0.330263	0.297940	0.702060	2
TP	0.215561	0.784439	0.632396	0.367604	2
FL	0.583797	0.416203	0.272576	0.727424	2
NP	0.541593	0.458407	0.724026	0.275974	2
PR	0.599239	0.400761	0.436461	0.563539	1
CI	0.459081	0.540919	0.238348	0.761652	1
HU	0.686288	0.313712	0.352165	0.647835	2
OD	0.626838	0.373162	0.741012	0.258988	2
PL	0.230629	0.769371	0.439029	0.560971	1

Sử dụng bộ tham số mờ gia tử đã được tối ưu (***PAR***$_{wine}$) ở trên, chúng ta sẽ ứng dụng thử nghiệm cho các trường hợp đối với bài toán, bao gồm cả hai sơ đồ ***No-RBO*** và ***RBO-SGA***. Quá trình sinh luật ***HAFRG*** gồm thuật toán ***IFRG2*** và phương pháp sàng cân bằng theo tiêu chuẩn SR^3 = ***c.s***.

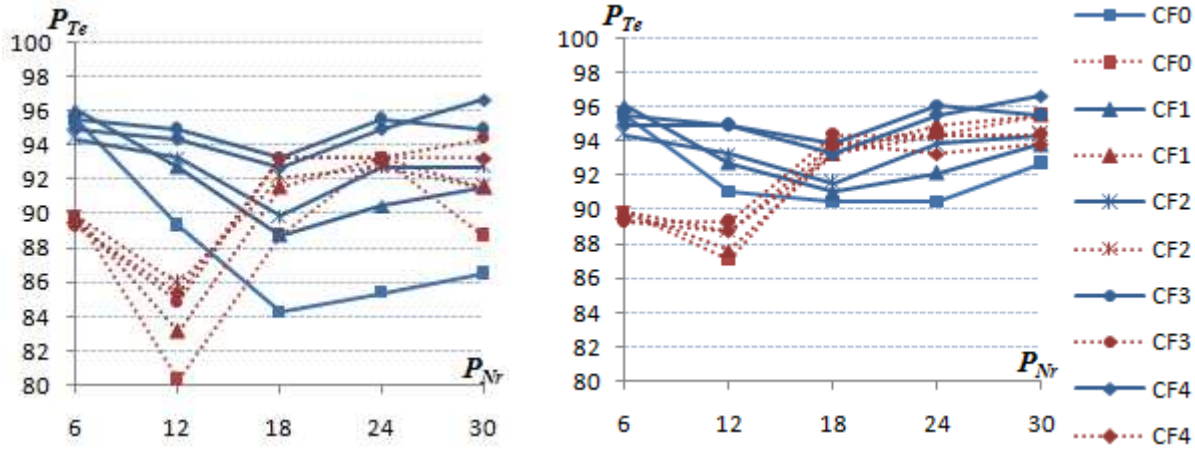
Sơ đồ ***No-RBO*** sử dụng cho trường hợp thử nghiệm *LV*1, các hệ luật sinh bởi quá trình ***HAFRG*** có kích thước tương ứng là 3, 6, 9, 12 và 15 luật. Đánh giá kết quả trên mỗi hệ luật này đối với tập dữ liệu kiểm tra (***P***$_{Te}$) theo cả hai phương pháp lập luận cùng với 4 phương pháp đánh giá trọng số luật, so sánh với kết quả của H. Ishibuchi [44] thể hiện trong Bảng 4.7 và Hình 4.3 (4.3*a* là phương pháp lập luận *single-winner-rule*, 4.3*b* là phương pháp lập luận *weigted-vote*). Kết quả ***No-RBO*** (chữ đậm) nhìn chung tốt hơn so với [44] (chữ nghiêng) trong các hệ luật có 3, 6, 12 và 15 luật, ký hiệu * là tốt nhất. Chẳng hạn với hệ 3 luật tỷ lệ phân lớp đúng đạt 96.07% lớn hơn của [44] là 89.89%, hệ 15 luật có kết quả 96.96% lớn hơn của [44] là 95.51%. Với hệ có 9 luật thì kết quả ***No-RBO*** thấp hơn không nhiều so với [44], theo phương pháp lập luận *weighted-vote*, tỷ số kết quả là 93.82% / 94.38%.

Bảng 4.7: Kết quả phân lớp ($\boldsymbol{P_{Te}}$(%)) sơ đồ ***No-RBO*** theo thuật toán ***IFRG2*** trong trường hợp *LV*1 của bài toán *WINE*, so sánh với phương pháp *FRBCS* của Ishibuchi [44] (chữ nghiêng)

Phương pháp đánh giá trọng số luật	Số luật ($\boldsymbol{P_{Nr}}$)				
	3	6	9	12	15
Phương pháp lập luận *single-winner-rule*					
Độ dài ($\boldsymbol{P_{Rl}}$)	**1.0**	**1.0**	**1.11**	**1.25**	**1.33**
CF^0	**95.51**	**89.33**	**84.27**	**85.39**	**86.52**
	*89.89**	*80.34*	*88.76*	*93.26**	*88.76*
CF^1	**96.07***	**92.70**	**88.76**	**90.45**	**91.57**
	*89.89**	*83.15*	*91.57*	*93.26*	*91.57*
CF^2	**94.38**	**93.26**	**89.89**	**92.70**	**92.70**
	*89.89**	*85.96**	*92.13*	*92.7*	*91.57*
CF^3	**95.51**	**94.94***	**93.26***	**95.51***	**94.94**
	89.33	*84.83*	*93.26**	*93.26**	*94.38**
CF^4	**94.94**	**94.38**	**92.70**	**94.94**	**96.63***
	89.33	*85.39*	*93.26**	*93.26**	*93.26*
Phương pháp lập luận *weighted-vote*					
Độ dài ($\boldsymbol{P_{Rl}}$)	**1.0**	**1.0**	**1.11**	**1.25**	**1.33**
CF^0	**95.51**	**91.01**	**90.45**	**90.45**	**92.70**
	*89.89**	*87.08*	*93.82*	*94.38*	*95.51**
CF^1	**96.07***	**92.70**	**91.01**	**92.13**	**93.82**
	*89.89**	*87.64*	*93.26*	*94.94**	*95.51**
CF^2	**94.38**	**93.26**	**91.57**	**93.82**	**94.38**
	*89.89**	*88.76*	*93.26*	*94.38*	*94.38*
CF^3	**95.51**	**94.94***	**93.82***	**96.07***	**95.51**
	89.33	*89.33**	*94.38**	*94.38*	*94.38*
CF^4	**94.94**	**94.94***	**93.26**	**95.51**	**96.63***
	89.33	*88.76*	*93.82*	*93.26*	*93.82*

Với sơ đồ thử nghiệm ***RBO-SGA***, chúng ta sinh tập luật $\boldsymbol{S_{900}}$ = ***HAFRG***(***PAR***$_{wine}$, ***IFRG2***, 900) (900 luật). Sử dụng thuật toán tìm kiếm tối ưu hệ luật mờ ***RBO-SGA*** trên tập luật $\boldsymbol{S_{900}}$ này và đánh giá kết quả đối với hệ luật tìm được theo phương pháp lập luận *single-winner-rule*, trọng số luật là CF^3. Các tham số chạy thuật toán ***RBO-SGA*** gồm kích thước quần thể N_p = 500 cá thể, số thế hệ

tiến hóa G_{max} = 150, trọng số các mục tiêu hàm *fitness* w_p = 0.99, w_n = 0.009 và w_a = 0.001. Sơ đồ này áp dụng cho 3 trường hợp thử nghiệm là *CV*10, *CV*20 và *CV*50. Kết quả thể hiện trong Bảng 4.8 cho thấy phương pháp trong luận án đạt hiệu quả khá cao trong tất cả các trường hợp thử nghiệm. Kết quả thử nghiệm các trường hợp *CV*10 là 99.51%, *CV*20 là 98.12% và *CV*50 là 97.39%. Điều này cho thấy mô hình sinh luật và tìm kiếm hệ luật tối ưu có khả năng dự báo tốt đối với các mẫu dữ liệu không sử dụng để sinh luật. Đối với tập huấn luyện (dùng để sinh luật), tỷ lệ phân lớp đúng được đánh giá trong các trường hợp đạt từ 99.17% đến 99.76%, cao hơn của F. Herrera [33] (95.71%). Trong đó phương pháp của H. Ishibuchi [47] đạt tỷ lệ cao nhất $\boldsymbol{P_{Tr}}$ = 100%. Ở đây phương pháp trong [60] không áp dụng tìm kiếm hệ luật tối ưu cũng như phương pháp rút gọn hệ luật nên kết quả có số luật khá lớn (124 luật).



(a) (b)

Hình 4.3: Đồ thị hiệu quả phân lớp ($\boldsymbol{P_{Te}}$) theo sơ đồ ***No-RBO*** trong trường hợp *LV*1 của bài toán *WINE*

Quá trình tìm kiếm hệ luật tối ưu của phương pháp trong luận án đặt giới hạn số luật tối đa là N_{max} = 7, do đó kết quả các hệ luật thu được với số lượng trung bình chỉ từ 6.78 đến 6.95. Tương tự, độ dài mỗi luật cũng được giới hạn tối đa là 3 điều kiện trong vế trái luật nên trung bình của các lần chạy thử nghiệm từ 1.72 đến 1.84. Rõ ràng kết quả này cho thấy hệ luật thu được khá đơn giản với số luật ít, dễ hiểu và tường minh đối với người dùng với số điều kiện trong vế trái của mỗi luật nhỏ.

Bảng 4.8: Kết quả thử nghiệm sơ đồ ***RBO-SGA*** theo thuật toán ***IFRG2*** của bài toán *WINE*, so sánh với các phương pháp *FRBCS* khác

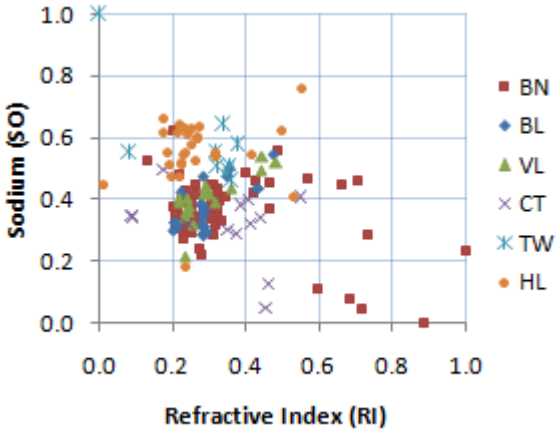
Phương pháp	P_{Nr}	P_{Rl}	P_{Tr} (%)	P_{Te} (%)
10-*folds cross validation* (*CV*10)				
S.M. Fakhrahmad và cộng sự [23]	/	/	/	95.3
H. Ishibuchi và cộng sự [47]	5.55	/	100.0	94.33
Sơ đồ ***RBO-SGA***	**6.78**	**1.72**	**99.17**	**99.51**
5-*folds cross validation* (*CV*20)				
F. Herrera và cộng sự [33]	/	/	95.71	54.24
Sơ đồ ***RBO-SGA***	**6.80**	**1.72**	**99.50**	**98.12**
2-*folds cross validation* (*CV*50)				
E. G. Mansoori và cộng sự [60]	124	/	/	93.93
Sơ đồ ***RBO-SGA***	**6.95**	**1.84**	**99.76**	**97.39**

4.4 Bài toán phân lớp các loại kính - GLASS

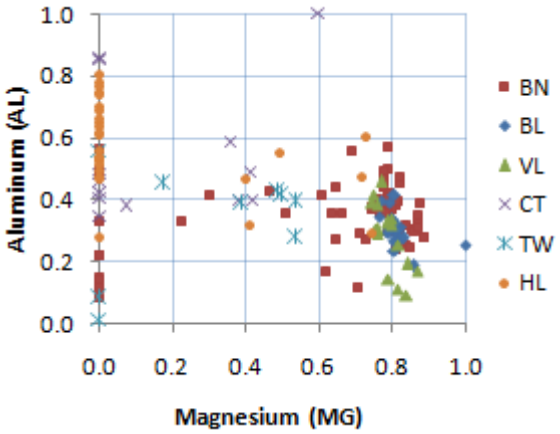
Bài toán phân lớp các loại kính với các mẫu dữ liệu được thu thập bởi B. German tại *Central Research Establishment Home Office Forensic Science Service*, và do Tiến sĩ Vina Spiehler giới thiệu, công bố tại [76]. Bài toán này có 9 thuộc tính gồm *Refractive index* (*RI*), *Sodium* (*SO*), *Magnesium* (*MG*), *Aluminum* (*AL*), *Silicon* (*SI*), *Potassium* (*PO*), *Calcium* (*CA*), *Barium* (*BA*) và *Iron* (*IR*). Các thuộc tính này (trừ thuộc tính *RI*) đều đo tỷ lệ phần trăm (%) trong mỗi đơn vị ô-xít. Tập dữ liệu gồm 214 mẫu với 6 lớp gồm *Building windows float processed* (*BF*), *Building windows non float processed* (*BN*), *Vehicle windows float processed* (*VF*), *Containers* (*CT*), *Tableware* (*TW*) và *Headlamps* (*HL*). Tỷ lệ số mẫu trong mỗi lớp tương ứng như sau: 70/*BF*, 76/*BN*, 17/*VF*, 13/*CT*, 9/*TW*, 29/*HL*. Tỷ lệ này chênh lệch khá lớn và là một trở ngại đối với việc xây dựng các mô hình phân lớp, do đó trong ứng dụng thử nghiệm chúng tôi chọn phương pháp sàng không cân bằng để rút gọn hệ luật.

Sơ đồ phân bố các dữ liệu trong các lớp theo từng cặp thuộc tính được thể hiện trong Hình 4.4. Quan sát trực quan thấy tập dữ liệu mẫu không có sự phân chia các lớp bởi các thuộc tính, các mẫu dữ liệu hầu như chồng chéo lên nhau giữa các lớp.

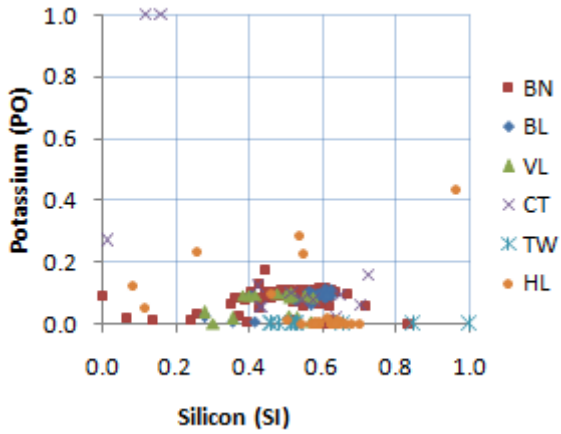
Đặc biệt ở các Hình 4.4*c* của cặp thuộc tính *SI* và *PO*, Hình 4.4*d* của cặp thuộc tính *CA* và *BA*, Hình 4.4*e* của cặp thuộc tính *IR* và *BA*. Rõ ràng đây là bài toán khá phức tạp trong vấn đề xây dựng các mô hình phân lớp. Ở đây thuộc tính *BA* được thể hiện cả trong Hình 4.4*d* và 4.4*e* vì sơ đồ cuối chỉ còn một thuộc tính *IR* nên thể hiện cùng với thuộc tính *BA* dưới dạng hai chiều.



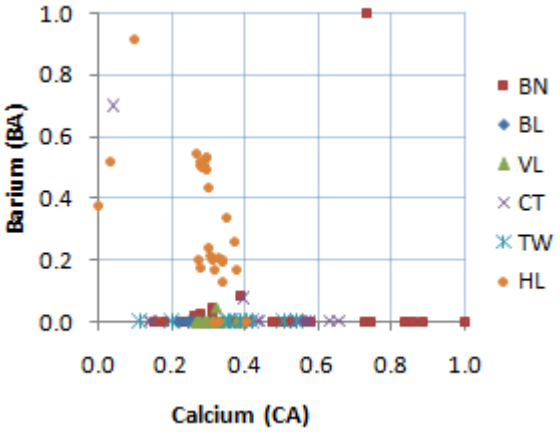
(a)



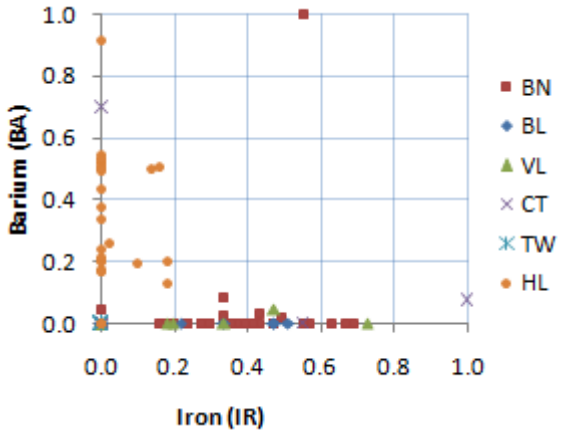
(b)



(c)



(d)



(e)

Hình 4.4: Sơ đồ phân bố các dữ liệu giữa các lớp của bài toán *GLASS*

Theo quy trình ứng dụng thử nghiệm mô hình, trước hết chúng ta áp dụng thuật toán ***FPO-SGA*** để tìm kiếm tối ưu bộ tham số mờ gia tử và mức phân hoạch mờ k_j trong ĐSGT. Trong bài toán này sẽ áp dụng ĐS2GT với phương pháp sinh luật dựa trên phân hoạch hệ các khoảng tương tự (thuật toán ***IFRG2***) và tiêu chuẩn sàng $SR^3 = \boldsymbol{c.s}$ để rút gọn hệ luật, áp dụng phương pháp lập luận *single-winner-rule* với trọng số luật CF^3. Kết quả bộ tham số tối ưu thể hiện trong Bảng 4.9.

Bảng 4.9: Tham số mờ gia tử tối ưu (***PAR***$_{glass}$) theo thuật toán ***IFRG2*** của bài toán *GLASS*

Thuộc tính	$fm_j(c^-)$	$fm_j(c^+)$	$\mu_j(L)$	$\mu_j(V)$	k_j
RI	0.391	0.609	0.343	0.657	3
SO	0.464	0.536	0.506	0.494	3
MG	0.307	0.693	0.321	0.679	2
AL	0.467	0.533	0.378	0.622	3
SI	0.571	0.429	0.466	0.534	1
PO	0.602	0.398	0.445	0.555	2
CA	0.325	0.675	0.350	0.650	2
BA	0.658	0.342	0.322	0.678	1
IR	0.392	0.608	0.426	0.574	1

Bây giờ chúng ta sẽ ứng dụng xây dựng hệ luật phân lớp cho bài toán với sơ đồ ***No-RBO*** trong trường hợp *LV*1. Các hệ luật gồm $\boldsymbol{S_6}$, $\boldsymbol{S_{12}}$, $\boldsymbol{S_{18}}$, $\boldsymbol{S_{24}}$, $\boldsymbol{S_{30}}$ được sinh

bởi thuật toán ***IFRG2*** và phương pháp sàng theo tiêu chuẩn SR^3. Đánh giá các kết quả thể hiện trong Bảng 4.10 chữ đậm, còn chữ nghiêng của phương pháp [44]. So sánh ta thấy kết quả phân lớp của phương pháp trong luận án ổn định và có nhiều trường hợp tốt hơn [44] trong các đánh giá trọng số luật từ CF^0 đến CF^3 trên cả hai phương pháp lập luận. Kết quả của [44] chỉ tốt với trọng số luật là CF^4, các trường hợp còn lại khá thấp (đều dưới 50%), trong khi kết quả ***No-RBO*** của luận án hầu hết đạt xấp xỉ và trên 50%. Chẳng hạn tại CF^3 với trường hợp 6 luật, kết quả của luận án đạt 52.34% trong khi của [44] chỉ đạt 39.25%.

Bảng 4.10: Kết quả phân lớp ($\boldsymbol{P_{Te}}$(%)) sơ đồ ***No-RBO*** theo thuật toán ***IFRG2*** trong trường hợp *LV*1 của bài toán *GLASS*, so sánh với phương pháp *FRBCS* của Ishibuchi [44] (chữ nghiêng)

Phương pháp đánh giá trọng số luật	Số luật ($\boldsymbol{P_{Nr}}$)				
	6	12	18	24	30
Phương pháp lập luận *single-winner-rule*					
Độ dài ($\boldsymbol{P_{Rl}}$)	**2.17**	**2.08**	**2.33**	**2.38**	**2.37**
CF^0	**48.60**	**49.07**	**50.93**	**50.93**	**49.53**
	45.79	*45.33*	*45.33*	*45.33*	*39.72*
CF^1	**51.40**	**51.87**	**53.74**	**54.67**	**51.87**
	49.53	*48.6*	*48.6*	*48.6*	*48.13*
CF^2	**51.87**	**52.34***	**54.21***	**55.61**	**52.34***
	45.79	*45.79*	*45.79*	*45.33*	*45.33*
CF^3	**52.34***	**52.34***	**54.21***	**56.07***	**50.93**
	39.25	*39.72*	*39.72*	*40.19*	*40.19*
CF^4	**50.00**	**50.00**	**52.34**	**53.74**	**48.60**
	58.88	*67.76*	*66.82*	*65.89*	*54.21*
Phương pháp lập luận *weighted-vote*					
Độ dài ($\boldsymbol{P_{Rl}}$)	**2.17**	**2.08**	**2.33**	**2.38**	**2.37**
CF^0	**48.60**	**49.07**	**50.47**	**51.40**	**51.40**
	45.79	*45.33*	*45.33*	*45.33*	*45.79*
CF^1	**51.40**	**52.34**	**54.67***	**55.61**	**54.67**
	49.53	*48.6*	*47.2*	*47.2*	*46.73*
CF^2	**51.87**	**52.80***	**54.67***	**56.54***	**55.61***
	45.79	*46.26*	*47.2*	*48.6*	*47.2*
CF^3	**52.34***	**51.87**	**54.21**	**56.07**	**55.14**
	39.25	*39.25*	*40.19*	*40.19*	*42.06*

CF^4	**50.00**	**50.47**	**52.34**	**55.14**	**54.67**
	58.88	*67.76*	*68.22*	*68.22*	*66.36*

Tiếp theo chúng ta sẽ ứng dụng với sơ đồ ***RBO-SGA***, bộ tham số mờ gia tử tối ưu (***PAR*$_{glass}$**) được dùng để sinh một tập luật đủ lớn bằng thuật toán ***IFRG*2** và phương pháp sàng theo tiêu chuẩn SR^3, ***S*$_{1000}$** = ***HAFRG***(***PAR*$_{glass}$**, ***IFRG*2**, 1000), trong đó giới hạn độ dài luật L = 4. Tìm kiếm tối ưu hệ luật mờ trên tập ***S*$_{1000}$** này bằng thuật toán ***RBO-SGA***, giới hạn số luật tối đa cho hệ tối ưu là N_{max} = 30. Kết quả thể hiện trong Bảng 4.11, cao hơn hẳn so với các phương pháp khác. Trong trường hợp thử nghiệm *CV*10, tỷ lệ phân lớp đúng trên tập kiểm tra (***P*$_{Te}$**) đạt 84.84% với số luật trung bình là 28.2, trong khi của [46] chỉ đạt 62.97% tại 28.32 luật và 61.64% tại 9.06 luật, của [23] đạt 70.1% nhưng không đưa ra số luật. Trường hợp *CV*50, kết quả 74.80% cũng cao hơn so với của [60] (53.32%).

Kết quả của sơ đồ ***RBO-SGA*** cao hơn hẳn ***No-RBO*** cho thấy rằng việc chọn một hệ luật đủ tốt cho bài toán về trực quan sử dụng các tiêu chuẩn để sàng là rất khó khăn. Một thuật toán tìm kiếm tối ưu được thiết kế thích hợp sẽ cho kết quả hệ luật mờ đạt tỷ lệ phân lớp khá cao, phương pháp dựa trên ***GA*** được hầu hết các tác giả quan tâm nghiên cứu và áp dụng.

Bảng 4.11: Kết quả thử nghiệm sơ đồ ***RBO-SGA*** theo thuật toán ***IFRG*2** của bài toán *GLASS*, so sánh với các phương pháp *FRBCS* khác

Phương pháp	***P*$_{Nr}$**	***P*$_{Rl}$**	***P*$_{Tr}$** (%)	***P*$_{Te}$** (%)
10-*folds cross validation* (*CV*10)				
S.M. Fakhrahmad và cộng sự [23]	/	/	/	70.1
H. Ishibuchi và cộng sự [46]	9.06	/	77.64	61.64
	28.32	/	82.09	62.97
Sơ đồ ***RBO-SGA***	**28.2**	**2.71**	**88.23**	**84.84**
2-*folds cross validation* (*CV*50)				
E. G. Mansoori và cộng sự [60]	33	/	/	53.32
L. Sanchez và cộng sự [73]	/	/	/	65.14
Sơ đồ ***RBO-SGA***	**28.87**	**2.83**	**93.78**	**74.80**

4.5 Bài toán phân lớp các loại men sinh học - YEAST

Tập dữ liệu mẫu cho bài toán phân lớp các loại men sinh học (***Yeast***) do giáo sư K. Nakai thu thập tại Viện phân tử và tế bào sinh học, Đại học Osaka, Nhật Bản, và được công bố trong [76]. Nhiều tác giả nghiên cứu đã sử dụng tập dữ liệu này để thử nghiệm các mô hình cho bài toán phân lớp [64], [58], [47]. Tập dữ liệu gồm 1484 mẫu chia thành 10 lớp và có 8 thuộc tính đó là:

1. (*MCG*) - McGeoch's method for signal sequence recognition.

2. (*GVH*) - Heijne's method for signal sequence recognition.

3. (*ALM*) - Score of the ALOM membrane spanning region prediction program.

4. (*MIT*) - Score of discriminant analysis of the amino acid content of the N-terminal region (20 residues long) of mitochondrial and non-mitochondrial proteins.

5. (*ERL*) - Presence of "HDEL" substring (thought to act as a signal for retention in the endoplasmic reticulum lumen). Binary attribute.

6. (*POX*) - Peroxisomal targeting signal in the C-terminus.

7. (*VAC*) - Score of discriminant analysis of the amino acid content of vacuolar and extracellular proteins.

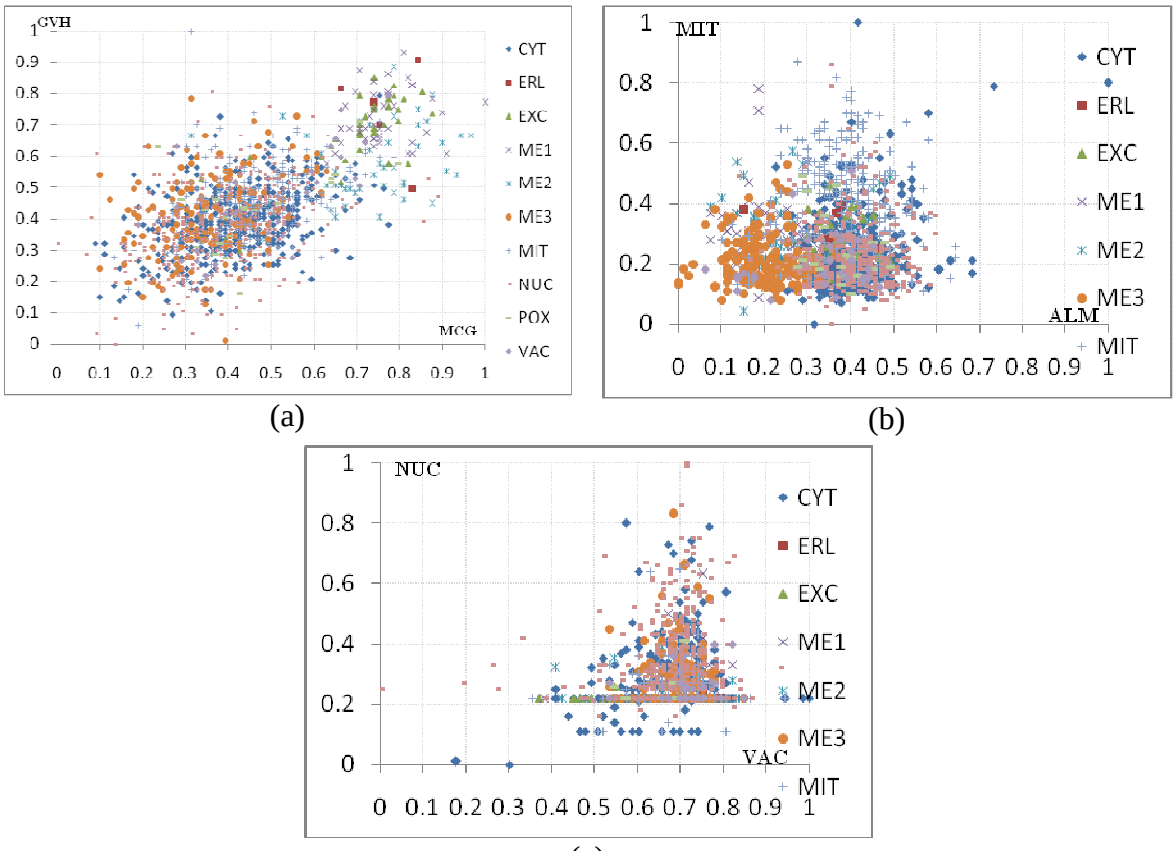
8. (*NUC*) - Score of discriminant analysis of nuclear localization signals of nuclear and non-nuclear proteins.

Bảng 4.12 thể hiện phân bố số lượng các mẫu dữ liệu theo từng lớp, Hình 4.8 thể hiện sự phân bố dữ liệu trên các lớp theo từng cặp thuộc tính: 4.8*a* cho cặp thuộc tính *MCG* và *GVH*, 4.8*b* cho cặp thuộc tính *ALM* và *MIT*, 4.8*c* cho cặp thuộc tính *VAC* và *NUC*. Đối với cặp thuộc tính *ERL* và *POX* có hầu hết các mẫu dữ liệu bằng 0 hoặc 1. Trực quan trên biểu đồ phân bố dữ liệu cho thấy bài toán rất phức tạp, các mẫu dữ liệu ở các lớp chồng chéo lên nhau, hầu như không có thuộc tính nào thể hiện tính trội hơn hẳn để phân lớp. Hơn nữa, số lượng mẫu trong tập dữ liệu khá lớn cùng với sự phân bố các mẫu dữ liệu không cân bằng nhau, tỷ số chênh lệch

phân bố này rất lớn, lên đến 463/5. Đây cũng là bài toán có số lớp khá lớn (10 lớp). Những thách thức không nhỏ đối với bất kỳ mô hình phân lớp nào.

Bảng 4.12: Số lượng các mẫu dữ liệu trong mỗi lớp của bài toán *YEAST*

Lớp	Mô tả	Số mẫu
CYT	Cytosolic or cytoskeletal	463
ERL	Endoplasmic reticulum lumen	5
EXC	Extracellular	37
ME1	Membrane protein, cleaved signal	44
ME2	Membrane protein, uncleaved signal	51
ME3	Membrane protein, no N-terminal signal	163
MIT	Mitochondrial	244
NUC	Nuclear	429
POX	Peroxisomal	20
VAC	Vacuolar	30



(a) (b)

(c)

Hình 4.5: Sơ đồ phân bố dữ liệu giữa các lớp của bài toán *YEAST*

Áp dụng thuật toán ***FPO-SGA*** để tối ưu bộ tham số gia tử theo phương pháp sinh luật dựa trên hệ phân hoạch các khoảng tính mờ trong ĐS2GT (thuật toán ***IFRG2***) và phương pháp sàng luật không cân bằng (vì tỷ lệ chênh lệch số mẫu giữa các lớp quá lớn) theo tiêu chuẩn SR^3. Hệ luật được sinh để đánh giá bộ tham số tối ưu là $\boldsymbol{S_{20}}$ (20 luật), sử dụng trọng số luật CF^3 và phương pháp lập luận *single-winner-rule*. Các tham số chạy ***FPO-SGA*** gồm kích thước quần thể tại mỗi thế hệ N_p = 100, số thế hệ tiến hóa G_{max} = 150, ràng buộc tham số là $0.2 \le fm(\boldsymbol{c}^-), \mu(L) \le 0.8$ và $1 \le k_j \le 2$. Kết quả tham số tối ưu thể hiện trong Bảng 4.13 sau.

Bảng 4.13: Tham số mờ gia tử tối ưu ($\boldsymbol{PAR_{yeast}}$) theo thuật toán ***IFRG2*** của bài toán *YEAST*

Thuộc tính	$\boldsymbol{fm_j(c^-)}$	$\boldsymbol{fm_j(c^+)}$	$\boldsymbol{\mu_j(L)}$	$\boldsymbol{\mu_j(V)}$	$\boldsymbol{k_j}$
MCG	0.528529	0.471471	0.396943	0.603057	1
GVH	0.441752	0.558248	0.338875	0.661125	1
ALM	0.436463	0.563537	0.341487	0.658513	2
MIT	0.764595	0.235405	0.572735	0.427265	1
ERL	0.519181	0.480819	0.529925	0.470075	1
POX	0.427810	0.572190	0.458303	0.541697	1
VAC	0.500003	0.499997	0.585536	0.414464	2
NUC	0.224894	0.775106	0.628089	0.371911	1

Tiếp theo chúng ta sẽ ứng dụng thử nghiệm trong hai trường hợp *CV*10 và *CV*20 theo sơ đồ ***RBO-SGA***. Sinh một tập luật đủ lớn $\boldsymbol{S_{900}}$ = ***HAFRG***($\boldsymbol{PAR_{yeast}}$, ***IFRG2***, 900) và áp dụng thuật toán ***RBO-SGA*** để tìm hệ luật tối ưu. Số luật tối đa để tìm kiếm tối ưu là N_{max} = 30, trọng số hàm *fitness* là w_p = 0.99, w_n = 0.009 và w_a = 0.001. Kích thước quần thể tại mỗi thế hệ N_p = 500, số thế hệ tiến hóa G_{max} = 150. Đánh giá kết quả trên hệ luật tìm được trong các trường hợp thử nghiệm thể hiện trong Bảng 4.14. Trong trường hợp *CV*10, hiệu quả trên tập kiểm tra ($\boldsymbol{P_{Te}}$) của [64] cao hơn [47] nhưng không đáng kể (58.26/57.42), trong khi đó kết quả của ***RBO-SGA*** tốt hơn đáng kể (60.09%), cả hiệu quả trên tập dữ liệu để sinh luật ($\boldsymbol{P_{Tr}}$). Số luật của ***RBO-SGA*** lớn hơn [47] (30/22.45). Nhìn chung các kết quả của ***RBO-SGA***

tốt hơn so với các phương pháp được so sánh. Điều này chứng tỏ hiệu quả của phương pháp ***RBO-SGA***, hệ luật đạt được khá đơn giản nhưng hiệu quả phân lớp tăng lên khá rõ rệt.

Bảng 4.14: Kết quả thử nghiệm sơ đồ ***RBO-SGA*** theo thuật toán ***IFRG2*** của bài toán *YEAST*, so sánh với các phương pháp *FRBCS* khác

Phương pháp	P_{Nr}	P_{Rl}	P_{Tr} (%)	P_{Te} (%)
10-*folds cross validation* (*CV*10)				
N.G. Pavlidis và cộng sự [64]	/	/	/	58.26
H. Ishibuchi và cộng sự [47]	22.45	2.92	63.23	57.42
Sơ đồ ***RBO-SGA***	**30.0**	**2.86**	**64.39**	**60.09**
5-*folds cross validation* (*CV*20)				
Sơ đồ ***RBO-SGA***	**30.0**	**2.93**	**64.94**	**59.96**
2-*folds cross validation* (*CV*50)				
L. Sanchez và cộng sự [73]	/	/	/	56.66
Sơ đồ ***RBO-SGA***	**30.0**	**2.92**	**66.04**	**58.56**

4.6 Kết luận Chương 4

Trong chương này luận án đã ứng dụng mô hình xây dựng hệ luật mờ theo tiếp cận ĐSGT để giải 4 bài toán phân lớp khá thông dụng, được nhiều tác giả nghiên cứu sử dụng để thử nghiệm các mô hình phân lớp. Tập dữ liệu mẫu của các bài toán được công bố rộng rãi trong [76] tại Đại học California, Irvin. Các bài toán này với những đặc trưng riêng biệt, từ đơn giản đến phức tạp cả về số thuộc tính, số lượng mẫu dữ liệu, mức độ chênh lệch số lượng mẫu dữ liệu giữa các lớp cũng như sự phân bố dữ liệu giữa các lớp.

Bài toán *IRIS* là đơn giản nhất trong số 4 bài trên, với số thuộc tính nhỏ và tập dữ liệu mẫu khá phân biệt giữa các lớp, số lượng mẫu cân bằng. Kết quả ứng dụng cả hai phương pháp sinh luật là thuật toán ***IFRG1*** và ***IFRG2*** đều cho thấy hiệu quả cao hơn hẳn so với các phương pháp trong các trường hợp thử nghiệm. Đặc biệt phương pháp ***IFRG1*** đạt hiệu quả phân lớp tối đa với 3 luật, trong khi của [50] đạt được với 5 luật. Hơn nữa, phương pháp ***IFRG2*** không những cho hiệu quả phân lớp

cao, hệ luật nhỏ mà còn đơn giản, tức số điều kiện tham gia trong mỗi luật ít. Như vậy, đã giảm thiểu được các thuộc tính dư thừa trong mỗi luật quyết định đến việc phân lớp tương ứng.

Sự phức tạp của bài toán *WINE* lớn hơn *IRIS*, do có nhiều thuộc tính nhất. Nếu không có sự rút gọn vế trái của luật thì hệ luật sinh ra sẽ rất phức tạp, chứa nhiều điều kiện của các thuộc tính dư thừa. Hơn nữa sự phân bố dữ liệu khá chồng chéo giữa các lớp. Do đó việc đã áp dụng phương pháp sinh luật bằng thuật toán ***IFRG2*** để giải quyết bài toán này là thích hợp. Kết quả đạt được cho thấy tính hiệu quả cao của phương pháp, sự đơn giản của hệ luật sinh ra. Trong hầu hết các trường hợp thử nghiệm, kết quả của phương pháp này tốt hơn nhiều trong sự so sánh với các phương pháp khác.

Hai bài toán còn lại *GLASS* và *YEAST* rất phức tạp, mặc dù số thuộc tính ít hơn *WINE* nhưng các dữ liệu chồng chéo dày đặc lên nhau, không phân biệt giữa các lớp. Đặc biệt bài toán *YEAST* có số mẫu dữ liệu lớn và phân bố số lượng mẫu trong các lớp chênh lệch nhau quá cao. Thật vậy, các phương pháp của các tác giả chỉ đạt hiệu quả phân lớp trên tập kiểm tra ($\boldsymbol{P_{Te}}$) trong khoảng từ 50% đến 70% đối với bài toán *GLASS*, còn bài toán *YEAST* rất thấp hầu hết dưới 60% trong các trường hợp thử nghiệm. Luận án đã ứng dụng ĐS2GT vào 2 bài toán này với thuật toán sinh luật ***IFRG2***, kết quả phân lớp ($\boldsymbol{P_{Te}}$) đạt khoảng 80% trong *GLASS* và xấp xỉ 60% trong *YEAST*, cao hơn so với các phương pháp khác. Chẳng hạn trong *GLASS* với trường hợp *CV*10, $\boldsymbol{P_{Te}}$ = 84.84% trong khi đó kết quả các phương pháp khác cao nhất là 70.1%.

Tuy nhiên, đánh giá kết quả của phương pháp chưa tính toán đến yếu tố thời gian. Các thuật toán di truyền để tìm kiếm bộ tham số mờ gia tử tối ưu chiếm thời gian khá lớn, mặc dù phương pháp dựa trên ĐSGT và đặc biệt là ĐS2GT, đã giảm bớt không gian các tham số cần tìm kiếm. Điều này cũng chưa được phân tích và đánh giá bởi các tác giả nghiên cứu, có thể do sự phức tạp và đa dạng của các bài toán ứng dụng.

KẾT LUẬN CHUNG

Luận án đạt được một số kết quả chính như sau:

1) Đề xuất sử dụng đại số 2 gia tử (ĐS2GT), tức là ĐSGT chỉ gồm 2 gia tử (một gia tử dương và một gia tử âm) và khảo sát các tính chất của nó. Khảo sát tính chất kế thừa ngữ nghĩa và quan hệ ngữ nghĩa của các giá trị ngôn ngữ. Giới thiệu khái niệm khoảng tương tự của các giá trị ngôn ngữ và xây dựng hệ khoảng tương tự cho một tập các giá trị ngôn ngữ. Trên cơ sở ĐS2GT, trong luận án đã khẳng định hệ khoảng tương tự luôn tồn tại và có thể ứng dụng xấp xỉ cho mọi quá trình thực.

2) Thiết kế hai thuật toán sinh luật mờ trực tiếp từ tập dữ liệu mẫu cho bài toán phân lớp. Thứ nhất, thuật toán ***IFRG*1** dựa trên hệ khoảng tính mờ của tập các giá trị ngôn ngữ tại mức *k* trong ĐSGT để sinh các luật mờ, thứ hai là thuật toán ***IFRG*2** dựa trên hệ khoảng tương tự của tập tất cả các giá trị ngôn ngữ từ mức 1 đến mức *k* trong ĐS2GT để sinh các luật mờ. Cả hai phương pháp này đều thực hiện theo "*vết*" dữ liệu mang ngữ nghĩa của các giá trị ngôn ngữ dẫn đến kết quả các luật được sinh ra. Khác với một số phương pháp FRBCS có độ phức tạp sinh luật là hàm mũ, hai thuật toán này được khẳng định là độ phức tạp đa thức đối với kích thước tập mẫu.

3) Trên cơ sở quan hệ ngữ nghĩa của các giá trị ngôn ngữ, luận án đã xây dựng phép kết nhập các giá trị ngôn ngữ khi chúng có kế thừa ngữ nghĩa và phục vụ cho việc kết nhập các luật mờ, nhằm rút gọn hệ luật. Bên cạnh đó, phương pháp sàng dựa trên các tiêu chuẩn đánh giá như độ tin cậy, độ hỗ trợ của luật cũng được áp dụng để rút gọn hệ luật.

4) Thiết kế hai thuật toán tìm kiếm tối ưu gồm thuật toán ***FPO-SGA*** để tìm bộ tham số mờ gia tử tối ưu cho mô hình đối với một bài toán ứng dụng, thuật toán ***RBO-SGA*** để tìm kiếm hệ luật mờ tối ưu cho bài toán đó. Hai thuật toán này được thiết kế dựa trên giải thuật di truyền (*Genetic Algorithm - GA*) kết hợp thuật toán mô phỏng tôi luyện (*Simulated Annealing - SA*) nhằm tăng tốc độ hội tụ cũng như tính ổn định của phương pháp tìm kiếm.

5) Ứng dụng mô phỏng mô hình vào 4 bài toán phân lớp rất đặc trưng với tập dữ liệu cung cấp bởi Đại học California - Irvin, được nhiều tác giả dùng để thử nghiệm cho các mô hình phân lớp. Đánh giá và so sánh kết quả với các phương pháp khác cho thấy tính hiệu quả của mô hình trong luận án.

Những kết quả trên đã mở rộng khả năng ứng dụng của ĐSGT, minh chứng cho ưu thế của ĐSGT trong việc tiếp cận đến phương pháp lập luận xấp xỉ và đóng góp vào giải quyết các bài toán phân lớp trong lĩnh vực khai phá dữ liệu. Song, một số nội dung trong luận án cần được tiếp tục nghiên cứu hoàn chỉnh và làm sâu sắc hơn:

- Phương pháp kết nhập các giá trị ngôn ngữ mới chỉ dừng lại ở mức độ ngữ nghĩa của chúng, nên chăng gia cố thêm các đánh giá về mặt thông tin để phép kết nhập đảm bảo có tính ứng dụng cao. Trên cơ sở đó, phương pháp kết nhập các luật cần được tinh chỉnh để đạt được hiệu quả cao về mặt thời gian.

- Mở rộng phương pháp xây dựng hệ luật mờ phân lớp dựa trên hệ khoảng tương tự trong ĐSGT tuyến tính thông thường, thay vì áp dụng trong ĐS2GT. Điều này cần một phương pháp xây dựng hệ khoảng tương tự trong ĐSGT mà không hạn chế số gia tử. Chắc chắn rằng phương pháp này mang tính tổng quát hơn cho việc ứng dụng về sau.

- Mỗi thuộc tính trong bài toán có tính chất quyết định đến việc phân lớp khác nhau, ở đây muốn nói đến mức độ. Do đó, việc rút gọn vế trái của luật bằng phương pháp loại bỏ một cách cơ học có thể làm mất mát thông tin. Có thể thay thế bằng cách bổ sung cho mỗi thuộc tính một trọng số thể hiện mức độ quyết định đến phân lớp.

- 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,...

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

1. Duong Thang Long, Nguyen Cat Ho, Tran Thai Son (2008), Hedge algebras for fuzzy reasoning using neural networks and genetic algorithms, *Proceedings of International Conference on Scientific Research in Open and Distance Education, Melia - Hanoi, VietNam*, pp. 138-153.

2. Nguyễn Cát Hồ, Dương Thăng Long, Trần Thái Sơn (2009), "Tiếp cận đại số gia tử cho phân lớp mờ", *Tạp chí Tin học và Điều khiển học*, Tập 25(1), tr. 53–68.

3. Nguyễn Cát Hồ, Dương Thăng Long, Trần Thái Sơn (2010), "Đại số gia tử hạn chế $\boldsymbol{AX}^2$ và ứng dụng cho bài toán phân lớp", *Tạp chí Khoa học và Công nghệ*, Tập 48(5), tr. 23-36.

4. Dương Thăng Long (2010), "Một phương pháp xây dựng hệ mờ có trọng số để phân lớp dựa trên đại số gia tử", *Tạp chí Tin học và Điều khiển học*, Tập 26(1), tr. 55-71.

5. Nguyễn Cát Hồ, Trần Duy Hùng, Dương Thăng Long, Trần Thái Sơn (2010), "Phương pháp tối ưu *Pareto* hệ luật mờ dựa trên đại số gia tử sử dụng giải thuật di truyền và ứng dụng vào bài toán phân lớp", *Tạp chí Tin học và Điều khiển học*, Tập 26(2), tr. 103-117.

6. Duong Thang Long, Nguyen Cat Ho, Tran Thai Son, Witold Pedrycz (2010), "Fuzzy Rule Extraction for Classification Problems Using Hedge Algebra-Based Semantics of Vague Terms", submitted to *International Journal of Approximate Reasoning*.

7. Dương Thăng Long, Lương Cao Đông, Trương Công Đoàn (2010), "Ảnh hưởng của tham số các gia tử trong hệ luật mờ phân lớp dựa trên đại số gia tử", báo cáo *Hội thảo Quốc gia về một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông*, Hưng Yên 19-20/8/2010.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Trần Ngọc Hà (2002), *Các hệ thống thông minh lai ứng dụng trong xử lý dữ liệu*, Luận án Tiến sĩ Toán học, Trường Đại học Bách khoa Hà Nội.

[2] Nguyễn Cát Hồ (2006), "Lý thuyết tập mờ và Công nghệ tính toán mềm", *Tuyển tập các bài giảng về Trường thu hệ mờ và ứng dụng*, in lần thứ 2, tr. 51-92.

[3] Nguyễn Cát Hồ (2008), "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.

[4] Nguyễn Cát Hồ, Phạm Thanh Hà (2007), "Giải pháp kết hợp sử dụng đại số gia tử và mạng nơron RBF trong việc giải quyết bài toán điều khiển mờ", *Tạp chí Tin học và Điều khiển học*, Tập 25(1), tr. 17-32.

[5] Nguyễn Cát Hồ, Nguyễn Văn Long (2003), "Làm đầy đại số gia tử trên cơ sở bổ sung các phần tử giới hạn", *Tạp chí Tin học và Điều khiển học*, Tập 19(1), tr. 62–71.

[6] Nguyễn Cát Hồ, Trần Thái Sơn (1995), "Về khoảng cách giữa các giá trị của biến ngôn ngữ trong đại số gia tử", *Tạp chí Tin học và Điều khiển học*, Tập 11(1), tr. 10-20.

[7] Trần Thái Sơn, Nguyễn Thế Dũng (2005), "Một phương pháp nội suy giải bài toán mô hình mờ trên cơ sở đại số gia tử", *Tạp chí Tin học và Điều khiển học*, Tập 21(3), tr. 248-260.

[8] Lê Xuân Việt (2008), *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 - Viện Khoa học và Công nghệ Việt Nam.

[9] Lê Xuân Vinh (2006), *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.

Tiếng Anh

[10] Abonyi J., Roubos J.A. and Setnes M. (2003), "Learning fuzzy classification rules from labeled data", *Information Sciences*, vol.150, pp. 77-93.

[11] Adler D. (1993), "Genetic Algorithms and Simulated Annealing: A Marriage Proposal", *Proc of the International Conf. On Neural Networks*, vol. 2, pp. 1104-1109.

[12] Akay D., Akcayol M.A., Kurt M. (2008), "NEFCLASS based extraction of fuzzy rules and classification of risks of low back disorders", *Expert Systems with Applications*, vol. 35, pp. 2107-2112.

[13] Bisht S. (2004), "Hybrid Genetic-simulated Annealing Algorithm for Optimal Weapon Allocation in Multilayer Defence Scenario", *Defence Science Journal*, vol. 54, no. 3, pp. 395-405.

[14] Bodenhofer U. (2004), *Genetic Algorithms: Theory and Applications*, lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2003/2004.

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

[16] Chang X.G. and Lilly J.H. (2004), "Evolutionary design of a fuzzy classifier from data", *IEEE Trans. Systems, Man., and Cybernetics*, part B 34 (4), pp. 1894-1906.

[17] Chen C.C. (2006), "Design of PSO-based Fuzzy Classification Systems", *Tamkang Journal of Science and Engineering*, vol. 9, no 1, pp. 63-70.

[18] Chen G. and Pham T.T. (2001), *Fuzzy Sets, Fuzzy Logic and Fuzzy Control Systems*, CRC Press, USA.

[19] Cheung K.C. and Wu J.N. (1998), "An Efficient Algorithm for Inducing Fuzzy Rules from Numerical Data", *Proceedings of the Eleventh International FLAIRS Conference*, American, 1998.

[20] Chow M.Y., Xu L., and Taylor L.S. (2006), “Data Mining Based Fuzzy Classification Algorithm for Imbalanced Data”, *IEEE International Conference on Fuzzy Systems*, Canada, 2006.

[21] Deb K., Agrawal S., Pratap A., and Meyarivan T. (2000), “A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II”, *Proc. of the Parallel Problem Solving from Nature VI Conference*, France, pp. 849-858.

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

[23] Fakhrahmad S.M. and Jahromi M. Zolghadri (2009), “A New Rule-weight Learning Method based on Gradient Descent”, *Proceedings of World Congress on Engineering 2009*, vol.1, WCE-2009.

[24] Fernandez A., Calderon M., Barrenechea E., Bustince H. and Herrera F. (2009), “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, 9/2009.

[25] Fuller R. (1995), *Neural Fuzzy Systems*, Physica-Verlag, Germany.

[26] Grabisch M. and Dispot F. (1992), “A comparison of some methods of fuzzy classification on real data”, *Proc. of IIZUKA ’92*, Iizuka, Japan, pp. 659-662.

[27] Guo Y., Robert G. (2002), *High Performance Data Mining*: *Scaling Algorithms, Applications and Systems*, Kluwer Academic Publishers, USA.

[28] Herrera F., Aguilera J.J., Chica M. and Jesus M.J. del (2007), “Niching genetic feature selection algorithms applied to the design of fuzzy rule-based classification systems”, *Proceedings of the IEEE International Conference on Fuzzy Systems*, London (UK), pp. 1794-1799.

[29] Herrera F., Fernandez A. and Jesus M.J. del (2008), “A Short Study on the Use of Genetic 2-Tuples Tuning for Fuzzy Rule Based Classification Systems in Imbalanced Data-Sets”, *8th International Conference on Hybrid Intelligent Systems*, Spain, pp. 483-488.

[30] Herrera F., Fernandez A., Garcıa1 S. and Jesus M.J. del (2007), “A Study on the Use of the Fuzzy Reasoning Method Based on the Winning Rule vs. Voting Procedure for Classification with Imbalanced Data Sets”, *Proceedings of the 9th International Work-Conference on Artificial Neural Networks*, Spain, pp. 375-382.

[31] Herrera F., Fernandez A., Garcıa1 S. and Jesus M.J. del (2008), “A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets”, *Fuzzy Sets and Systems*, vol.159, pp. 2378 – 2398.

[32] Herrera F., Sanz J., Fernandez A. and Bustince H. (2009), “A First Study on the Use of Interval-Valued Fuzzy Sets with Genetic Tuning for Classification with Imbalanced Data-Sets”, *Proceedings of the Fourth International Conference on Hybrid Artificial Intelligence Systems*, Salamanca (Spain), pp. 581-588.

[33] Herrera F., Villar P. and Fernandez A. (2009), “A Genetic Learning of the Fuzzy Rule-Based Classification System Granularity for highly Imbalanced Data-Sets”, *IEEE International Conference on Fuzzy Systems*, Jeju Island (Korea), pp. 1689-1694.

[34] Ho N. C. (2007), “A topological completion of refined hedge algebras and a model of fuzziness of linguistic terms and hedges”, *Fuzzy Sets and Systems*, vol.158, pp.436-451.

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

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

[37] Ho N. C. and Wechler W. (1990), “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.

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

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

[40] Hou Yuan-long, Chen Ji-lin, Xing Zong-yi, Jia Li-min, and Tong Zhong-zhi (2006), “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.

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

[42] Ishibuchi H. and Nakashima T. (2001), “Effect of Rule Weights in Fuzzy Rule-Based Classification Systems”, *IEEE Trans. on Fuzzy Systems*, vol.9, no.4, pp.506-515.

[43] Ishibuchi H. and Yamamoto T. (2004), “Fuzzy Rule Selection by Multi-Objective Genetic Local Search Algorithms and Rule Evaluation Measures in Data Mining”, *Fuzzy Sets and Systems*, vol.141, no.1, pp. 59-88.

[44] Ishibuchi H. and Yamamoto T. (2005), “Rule weight specification in fuzzy rule-based classification systems”, *IEEE Trans. on Fuzzy Systems*, vol. 13, no. 4, pp. 428-435.

[45] Ishibuchi H., Nakashima T. and Murata T. (2001), “Three-Objective Genetics-Based Machine Learning for Linguistic Rule Extraction”, *Information Science*, vol.136, no.1-4, pp.109-133.

[46] Ishibuchi H., Nojima Y. (2007), “Analysis of interpretability-accuracy trade-off fuzzy systems by multiobjective fuzzy genetics-based machine learning”, *International Journal of Approximate Reasoning*, vol.44, no.1, pp.4–31.

[47] Ishibuchi H., Nojima Y. and Kuwajima I. (2009), “Parallel distributed genetic fuzzy rule selection”, *Soft Computing - A Fusion of Foundations*, Methodologies and Applications, SpringerLink, vol. 13, no. 5, pp. 511-519.

[48] Kasabov N.K. (1998), *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, The MIT Press, USA.

[49] Kevin L. and Olivier S. (2006), “Fuzzy Histograms and Density Estimation”, *Advances in Soft Computing*, Springer Berlin, ISSN 1615-3871, pp. 45-52.

[50] Khotanzad A. and Zhou E. (2007), “Fuzzy Classifier Design Using Genetic Algorithms”, *Pattern Recognition*, vol. 40, no.12, pp. 3401-3414.

[51] Koza R.J. (1998), *Genetic Programming*: *On the Programming of Computers by Means of Natural Selection*, The MIT Press, UK.

[52] Kruse R., Klawonn F. and Nauck D. (1992), “Fuzzy Sets, Fuzzy Controllers and Neural Networks”, *Scientific Journal of the Humboldt-University of Berlin*, Series Medicine 41, no.4, pp.99-120.

[53] Kubalika J., Rothkrantz L. and Lazanskya J. (2001), “Genetic Programming Fuzzy Rule Extractor Using Class Preserving Representation”, *The 13th Belgian-Dutch Conference on Artificial Intelligence*, University of Amsterdam, pp.167-174.

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

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

[56] Lee C.Y., Lin C.J, and Hong S.J. (2006), "An Efficient Fuzzy Classifier Based on Hierarchical Fuzzy Entropy", *International Journal of Information Technology*, vol.12, no.6.

[57] Leondes C.T. (1998), *Fuzzy Logic and Expert Systems Applications*, Academic Press, USA.

[58] Liu Huan, Jin Rong (2005), "A Novel Approach to Model Generation for Heterogeneous Data Classification", *Proceedings of the 19th International Joint Conference on Artificial Intelligence* - Scotland, pp. 746-751.

[59] Lughofer E., Angelov P., Zhou X. and Filev D. (2007), "Architectures for Evolving Fuzzy Rule-based Classifiers", *Proc. Systems, Man and Cybernetics conference (SMC) 2007*, Montreal, Canada, pp. 2050-2055.

[60] Mansoori E.G., Mansoori J.Z. and Katebi Seraj D. (2007), "A weighting function for improving fuzzy classification systems performance", *Fuzzy Sets and Systems*, vol. 158, pp.583 – 591.

[61] Menon A. (2004), *Frontiers of Evolutionary Computation*, Kluwer Academic Publishers, USA.

[62] Mukhopadhyay A. and Saha I. (2008), "Genetic Algorithm and Simulated Annealing based Approaches to Categorical Data Clustering", *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong - 2008, vol.1, pp. 1-6.

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

[64] Pavlidis N.G., Georgiou V.L., Parsopoulos K.E., Alevizos, Vrahatis M.N. (2004), "Optimizing the Performance of Probabilistic Neural Networks in a

Bionformatics Task", *Proceedings of the EUNITE 2004 Conference*, pp. 34-40.

[65] Pedrycz W. and Kwak K.C. (2006), "Linguistic models as a framework of user-centric system modeling", *IEEE Transactions on Systems, Man, and Cybernetics*, Part A 36(4), pp. 727-745.

[66] Pedrycz W. and Pizzi N.J. (2009), "Discriminatory Components for Pattern Classification", *IFSA/EUSFLAT Conf. 2009*, pp. 748-753.

[67] Pedrycz W. and Weber R. (2008), "Special issue on soft computing for dynamic data mining", *Appl. Soft Comput.* 8(4), pp. 1281-1282.

[68] Pedrycz W. and Yu F. (2009), "The design of fuzzy information granules: Tradeoffs between specificity and experimental evidence", *Appl. Soft Comput.* 9(1), pp. 264-273.

[69] Pedrycz W., Oliveira de J.V. (2007), *Advances in Fuzzy Clustering and Its Applications*, John Wiley & Sons Ltd, UK.

[70] Prade H., Djouadi Y., Alouane B. (2009), "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.

[71] Rao C.R., Jadaan O.A., Rajamani L. (2008), "Non-Dominated Ranked Genetic Algorithm for Solving Multi-Objective Optimization Problems: NRGA", *Journal of Theoretical and Applied Information Technology*, Pakistan.

[72] Ross T.J. (2004), *Fuzzy Logic with Engineering Applications*, John Wiley & Sons Ltd, UK.

[73] Sanchez L., Cordon O., Quirin A., and Trawinski K. (2010), "Introducing a Genetic Fuzzy Linguistic Combination Method for Bagging Fuzzy Rule-Based

Multiclassification Systems", *Fourth International Workshop on Genetic and Evolutionary Fuzzy Systems*, March 2010, Mieres, Spain.

[74] Semman I.E. and Marghny M.H. (2005), "Extracting fuzzy classification rules with gene expression programming", *In Proceedings of the International Conference on Artificial Intelligence and Machine Learning, AIML 2005*, Cairo, Egypt.

[75] Shen Q. and Huang Z.H. (2003), "A new fuzzy interpolative reasoning method based on center of gravity", *Proceedings of the International Conference on Fuzzy Systems*, vol.1, pp.25–30.

[76] The Machine Learning Repository of University of California - Irvine, at address of *http://archive.ics.uci.edu/ml/datasets.html*.

[77] Wang Li-Hui, Chen Yung-Chou and Chen Shyi-Ming (2006), "Generating Weighted Fuzzy Rules from Training Data for Dealing with the Iris Data Classification Problem", *International Journal of Applied Science and Engineering*, vol. 4, no.1, pp.41-52.

[78] Yahmada K. and Phuong N.H. (editors) (2001), *Proceedings of the Second Vietnam-Japan Symposium on Fuzzy Systems and Applications*, VJFUZZY'2001.

[79] Ying H. (1998), "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.

[80] Zadeh L.A. (1965), "Fuzzy sets", *Information and Control* 8, pp.338-358.

[81] Zadeh L.A. (2000), *Fuzzy sets and fuzzy information granulation theory – key selected papers*, Beijing Normal University Press, China.

[82] Zimmermann H.J. (1991), *Fuzzy sets theory and its applications*, 2nd Ed., Kluwer Acad. Pub., USA.