BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
******

LƯƠNG SONG VÂN

HỌC MÁY, HỌC MÁY MÔ TẢ PHỨC: THUẬT TOÁN VÀ VẤN ĐỀ RÚT GỌN LỖI

LUẬN ÁN THẠC SỸ KHOA HỌC
CHUYÊN NGÀNH TIN HỌC

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PTS. HÀ QUANG THỤY

HÀ NỘI - 1999

MỤC LỤC

PHẦN MỞ ĐẦU

Học máy (học tự động) là một lĩnh vực quan trọng trong Tin học, đặc biệt đối với lĩnh vực công nghệ tri thức. Mục tiêu chính của học máy là tạo ra các phương pháp và chương trình làm cho máy tính có thể học được như người. Rất nhiều công trình nghiên cứu về lý thuyết và triển khai đã được công bố trong lĩnh vực học máy mà phần lớn được tập hợp trong tạp chí khá nổi tiếng "Machine Learning" do nhà xuất bản Kluwer ấn hành. Lĩnh vực học máy có quan hệ mật thiết với lĩnh vực phát hiện tri thức ([1, 3, 11]) và vì vậy hiện nay, số lượng các nghiên cứu về học máy vẫn đang ngày càng phát triển với tốc độ cao. Ở Việt nam, đã có nhiều nhà khoa học quan tâm đến lĩnh vực nói trên và nhiều công trình nghiên cứu có giá trị đã được công bố ([1]). Lĩnh vực học máy có liên quan mật thiết với nhiều lĩnh vực khác nhau của Toán học và Tin học. Nhiều mô hình, nhiều phương pháp trong học máy có quan hệ mật thiết với các mô hình Toán học như dàn Galois [2], lý thuyết Bayes [6, 7, 8, 13, 14] v.v.

Luận văn ***"Học máy, học máy mô tả phức: thuật toán và vấn đề rút gọn lỗi"*** có nội dung đề cập tới một số mô hình, thuật toán điển hình trong học máy. Hai nội dung cơ bản được trình bày trong luận văn là các thuật toán điển hình và vấn đề rút gọn lỗi trong học máy. Học máy mô tả phức là một mô hình học máy nhằm giảm thiểu lỗi trong học máy có giám sát đang được nghiên cứu rộng rãi trên thế giới hiện nay ([2, 6, 7, 8, 13, 14]) cũng được trình bày trong luận văn.

Nội dung của luận văn bao gồm bốn chương được trình bày như dưới đây.

Chương 1 với tiêu đề ***"Bài toán học máy và một số thuật toán"*** đề cập tới những vấn đề chung nhất của bài toán học máy: học máy không giám sát và học máy có giám sát, các thuật toán điển hình trong tách nhóm (học không giám sát) và phân lớp (học có giám sát). Các thuật toán Bayes, k-người láng giềng gần nhất, thuật toán cây quyết định v.v. được giới thiệu. Các nội dung nói trên được tổng hợp từ các tài liệu ([1, 2, 6, 7, 11, 14]).

Chương 2 với tiêu đề ***"Học máy mô tả phức"*** giới thiệu một số mô hình học máy mô tả phức được đề xướng và phát triển tại trường Đại học Tổng hợp California, Ivrin. Luận văn trình bày nội dung cơ bản về các mô hình học máy mô tả phức, các thuật toán phân lớp áp dụng trong các mô hình học máy mô tả phức từ FOIL đến HYDRA-MM. Các chiến lược "chia nhỏ để chế ngự", "leo đồi ngẫu nhiên" v.v., các thuật toán Bayes, k-người láng giềng gần nhất được mô tả trong mỗi mô hình học. Luận văn cũng giới thiệu sự tiến bộ của mô hình mới so với mô hình sẵn có. Các nội dung nói trên được tổng hợp từ các tài liệu ([6, 7, 8, 14]).

Chương 3 với tiêu đề ***"Rút gọn lỗi trong học máy"*** đề cập tới một số nội dung liên quan đến lỗi và rút gọn lỗi trong học máy và học máy mô tả phức. Các khái niệm về lỗi tuyệt đối, lỗi tương đối, lỗi tương quan được trình bày. Mô hình học máy mô tả phức là một giải pháp hiệu quả trong việc rút gọn lỗi. Một số giải pháp về thuộc tính không tương ứng, đa dạng hoá dữ liệu, tổ hợp chứng cứ v.v. được giới thiệu và phân tích về khả năng rút gọn lỗi của mỗi giải pháp. Một số đánh giá thực nghiệm của các tác giả mô hình cũng được nêu ra nhằm minh họa tính hiệu quả của các giải pháp. Các nội dung trong chương này được rút ra từ các tài liệu [5-11] và đặc biệt là từ công trình của Ali. K. & Pazzani M. [5].

Chương 4 với tiêu đề "***Thuật toán tìm kiếm và phân lớp trong cơ sở dữ liệu full-text***" trình bày các nội dung liên quan đến hai bài toán điển hình trong cơ sở dữ liệu full-text, đó là tìm kiếm và phân lớp. Nội dung của chương này là sự phát triển một số nội dung đã được trình bày trong [4, 11]. Sử dụng mô hình vector trong thuật toán phân lớp là một thể hiện cụ thể các nội dung tương ứng trong [11] và cho phép thuật toán hoạt động với tốc độ nhanh. Luận văn đề xuất một số cải tiến trong mô hình vector trong vấn đề từ đồng nghĩa và số lượng xuất hiện từ khóa với hai mục đích: thể hiện tốt hơn nội dung văn bản và tăng tốc độ thực hiện các thuật toán. Do sự hạn chế về trình độ và thời gian nên luận văn mới

phác hoạ ý tưởng về một hệ quản trị cơ sở full-text có cài đặt các thuật toán trên đây.

Em xin chân thành bày tỏ lòng biết ơn sâu sắc tới thầy giáo - PTS. Hà Quang Thuỵ, người đã tận tình hướng dẫn, tạo điều kiện giúp đỡ và bổ sung cho em nhiều kiến thức quý báu trong suốt quá trình em làm luận văn. Em cũng xin cảm ơn thầy PGS. TS. Nguyễn Xuân Huy và thầy PTS. Nguyễn Tuệ đã đóng góp nhiều ý kiến giúp em hoàn chỉnh hơn luận văn của mình. Cuối cùng, em xin chân thành cảm ơn tất cả các thầy cô giáo trong khoa Công Nghệ Thông Tin (trước đây) và khoa Công Nghệ (hiện nay), cũng như phòng Khoa học và đào tạo sau đại học, trường Đại học Khoa học Tự nhiên đã tạo điều kiện giúp đỡ về các phương tiện nghiên cứu, giúp em hoàn thành mọi thủ tục để em được bảo vệ luận văn này.

Học viên

Lương Song Vân

CHƯƠNG 1. BÀI TOÁN HỌC MÁY VÀ MỘT SỐ THUẬT TOÁN

I.1. BÀI TOÁN HỌC MÁY

I.1.1. Bài toán học máy

Học máy (*machine learning*) được hiểu như một quá trình gồm hai giai đoạn: giai đoạn học và giai đoạn áp dụng nhằm tự động nhận rõ đặc trưng về đối tượng. Mỗi lĩnh vực được con người quan tâm luôn luôn liên quan đến tập hợp các khái niệm. Từ những kinh nghiệm đã học theo một số mẫu cho trước, cần phát hiện đặc trưng của một đối tượng mới. Học máy còn được quan niệm như là một quá trình thực hiện các kỹ xảo, mà nhờ đó, tri thức được thu nhận thông qua kinh nghiệm. Mục tiêu chính của học máy là tạo ra các phương pháp và chương trình làm cho máy tính "có thể học được" như người. Tuy nhiên, trong một số phạm vi nghiên cứu hẹp hơn, bài toán học máy được quan niệm một cách đơn giản dưới dạng bài toán "phân lớp": xếp một đối tượng nào đó vào một trong những lớp được coi là đã biết.

Bài toán học máy có thể được trình bày một cách hình thức như dưới đây.

Giả sử tồn tại một tập các khái niệm nền K_O (tập khái niệm nền K_O có thể chưa biết) tương ứng với một phân hoạch dữ liệu đối với một miền D nào đó. Tồn tại ánh xạ đa trị M từ K_O vào 2^D theo đó ứng với mỗi khái niệm nền x thuộc K_O tới một tập dữ liệu (được gọi là các ví dụ mẫu ứng với khái niệm x) thuộc miền D. Một khái niệm nền đặc trưng cho một lớp đối tượng.

Mở rộng tập khái niệm nền K_O tới tập khái niệm K ($K_O \subseteq K$) được gọi là tập các khái niệm. Cho biết tồn tại ánh xạ nào đó từ K_O tới $K \setminus K_O$ (ánh xạ nói trên có thể chưa biết) cho phép bằng cách nào đó nhận biết một khái niệm thông qua mối quan hệ với các khái niệm nền.

Quá trình học máy được phân chia thành hai giai đoạn và tương ứng với hai giai đoạn đó, kết quả của học máy có hai dạng như trình bày dưới đây.

- Kết quả của việc học máy cho ra *tập khái niệm K*, *tập khái niệm nền* K_O và *ánh xạ L từ* K_O *tới một tập các luật suy diễn liên quan tới mỗi khái niệm nền* (Trường hợp đặc biệt, tập khái niệm K và tập khái niệm nền K_O là đã biết). Theo ánh xạ này, mỗi khái niệm nền được tương ứng với một số luật suy diễn dạng Horn - cấp 1. Kiểu học này được gọi là "*học không giám sát*" theo nghĩa không có một áp đặt từ trước đối với quá trình học do thông tin về mô hình là rất ít. Một dạng đặc biệt của học máy không giám sát là tách (phân hoạch) một tập đối tượng thành một số nhóm (đoạn) đối tượng với một số đặc trưng nào đó. Bài toán học dạng này được gọi là *bài toán tách nhóm* (tách đoạn).

- Giả sử đã có ánh xạ L nói trên (từ mỗi khái niệm nền thuộc K_O tới các mô tả tương ứng) và phép biểu diễn một khái niệm thông qua các khái niệm nền. Bài toán đặt ra là cần tìm ra khái niệm tương ứng với ví dụ được hệ thống tiếp nhận. Học máy kiểu này còn được gọi là "*học có giám sát*" theo nghĩa đã hướng đích tới tập khái niệm K. Có thể sử dụng một số cách thức đoán nhận trước đối với các khái niệm để nhanh chóng phát hiện khái niệm tương ứng với ví dụ. Một dạng đặc biệt của học có giám sát là phân một đối tượng vào lớp thích hợp trong một tập các lớp cho trước. Bài toán học kiểu này được gọi là "*bài toán phân lớp*".

I.1.2. Một số đặc trưng trong học máy

Các phương pháp học máy thường được phân loại theo bản chất của dữ liệu được sử dụng cho quá trình học. Tương ứng với phương pháp học không giám sát là quá trình máy cần phát hiện ra các khái niệm dựa trên một tập thể hiện chưa biết thuộc về khái niệm nào. Tương ứng với phương pháp học có giám sát là quá trình máy tính cần tìm ra đặc trưng của các khái niệm dựa trên tập các thể hiện (instances) đã biết về khái niệm này.

Học máy không giám sát (bài toán tách nhóm) cần đạt được một số mục tiêu như sau [2]:

- Phân rã tập đối tượng thành các tập con, mỗi tập con đó tương ứng với một khái niệm (tách nhóm). Chính bản thân khái niệm cũng được phát hiện trong quá trình học máy. Trong một số trường hợp riêng, quá trình tách nhóm còn được thể hiện dưới dạng cây nên quá trình học máy dạng này được gọi là phân loại phân cấp (hierarchical clustering).

- Tìm ra đặc trưng của các tập con đã được phân hoạch trong quá trình phân rã. Những đặc trưng này được dùng cho việc phân lớp một đối tượng vào một tập con. Quá trình này còn được gọi là đặc trưng hoá các khái niệm. Luật suy diễn dạng Horn-cấp 1 là một trong những dạng biểu diễn điển hình về đặc trưng hoá các khái niệm ([6, 7, 8]). Tuy nhiên, trong nhiều trường hợp mô hình sử dụng một tập mẫu thay cho một khái niệm do chưa thể tìm ra được biểu diễn đối với các khái niệm tương ứng.

Như đã được trình bày, do bài toán học máy không giám sát tiếp nhận rất ít thông tin đầu vào và vì vậy, chưa có được nhiều kết quả nghiên cứu và công nghệ giải quyết bài toán ([2]). Phần sau của luận văn sẽ trình bày một số giải pháp chung nhất đối với bài toán học máy không giám sát. Một dạng đơn giản của thuật toán học máy không giám sát được trình bày trong [2], trong đó nghiên cứu sự thay đổi của hệ thống khái niệm cùng các đặc trưng của chúng khi dữ liệu được thay đổi. Nhiều dạng khác nhau của học máy không giám sát đã được khảo sát mà việc nghiên cứu về sự phụ thuộc thô là một trong những dạng điển hình ([03]).

Khác với học máy không giám sát, ***học máy có giám sát*** thu nhận được nhiều thành tựu cả về lý luận lẫn triển khai ứng dụng. Dưới đây là một số nội dung đặc trưng của học máy có giám sát:

- Trong một số mô hình học máy có giám sát, việc đặc trưng hoá mỗi khái niệm (mỗi nhóm dữ liệu) được thể hiện thông qua việc mô tả một tập ví dụ điển

hình tương ứng với khái niệm đó. Thông qua một khoảng cách giữa các đối tượng được xác định một cách thích hợp, nhiều thuật toán đã được sử dụng để kiểm nghiệm sự tương ứng một đối tượng đối với một khái niệm.

- Trong nhiều mô hình học máy khác, mỗi khái niệm được biểu diễn nhờ một dãy các luật Horn-cấp 1 dạng:

class-a(X,Y) ←b(X),c(Y)

bao gồm phần đầu (class-a(X,Y)) liên quan đến khái niệm và phần thân liên quan đến các literal (b(X),c(Y)). Thông qua quá trình suy diễn tương ứng với các luật nói trên có thể kiểm nghiệm được khái niệm phù hợp với đối tượng.. Chẳng hạn, luật sau đây tham gia biểu diễn khái niệm ung_thư_vú:

ung_thư_vú (Tuổi,..., Mức độ) ← >(Tuổi, 50), >(Mức độ, 3)

Theo luật này, người phụ nữ được biểu thị thông qua một tập hợp các giá trị của các biến (Tuổi,..., Mức độ) có bệnh ung thư vú nếu bà ta đã hơn 50 ***tuổi*** và ***mức độ*** trầm trọng của bệnh lớn hơn 3 độ.

- Một đặc trưng quan trọng cần được khảo sát là sai sót trong học máy có giám sát. Để đánh giá mức độ tốt của một mô hình học máy, người ta thường đưa ra một bộ các ví dụ kiểm tra (ví dụ test). Một sai sót được phát hiện khi ví dụ đã biết thuộc vào khái niệm x song lại được hệ thống xếp vào khái niệm y mà $x \neq y$. Hiển nhiên, một mô hình được coi là tốt khi số lượng sai sót kiểm tra là ít hoặc không có.

Có rất nhiều công trình khoa học nghiên cứu về học máy có giám sát. Một trong những nội dung cốt lõi của lĩnh vực này là giảm bớt sai sót học máy. Một trong những hướng để giảm thiểu sai sót đang được phát triển là ***học máy mô tả phức*** ([6, 7, 8, 13, 14]). Trong chương 2 và chương 3, một số mô hình điển hình và một số nội dung chính yếu về học máy mô tả phức được trình bày.

I.1.3. Phương pháp điển hình biểu diễn tri thức trong học máy

Như đã trình bày, biểu diễn tri thức đi liền với bài toán học máy ([4]). Nhiều mô hình hệ thống liên quan đến việc kết hợp việc học tự động với thu

nhận tri thức ([2]) đã được đề xuất và đánh giá. Những phương pháp điển hình nhất biểu diễn tri thức trong học máy có thể kể đến là: Phương pháp biểu diễn lôgic, phương pháp biểu diễn theo xác suất và phương pháp biểu diễn theo đối tượng.

Theo phương pháp ***biểu diễn lôgic***, mỗi khái niệm được như một cặp (thể hiện, đặc trưng). Luật Horn-cấp 1 là một ví dụ về việc sử dụng phương pháp biểu diễn này.

Theo phương pháp ***biểu diễn theo xác suất***, mỗi khái niệm được biểu diễn như một hình mẫu phản ánh các đặc trưng chung và tiêu biểu nhất của các thể hiện. Khi đã xác định được các xác suất tiên nghiệm có thể nhận được một xác suất hậu nghiệm kết quả. Các mô hình học máy Bayes sử dụng phương pháp biểu diễn theo xác suất.

Theo phương pháp ***biểu diễn theo đối tượng***, mỗi khái niệm được hiểu và biểu diễn thông qua một tập các thể hiện tiêu biểu. Dạng quá đơn giản về tập các thể hiện là cho biết một tập đối tượng tương thích với khái niệm tương ứng. Mô hình tương ứng thuật toán ***người láng giềng gần nhất*** (k-người láng giềng gần nhất) sử dụng phương pháp biểu diễn theo đối tượng.

Trong mỗi ngữ cảnh áp dụng, thuật toán học máy sẽ chọn một trong ba phương pháp biểu diễn nói trên.

I.2. THUẬT TOÁN ĐIỂN HÌNH TRONG HỌC MÁY

I.2.1. Thuật toán tách nhóm

Các phương pháp tách nhóm (tách đoạn - clustering) tiếp cận tới những vấn đề tách nhóm định địa chỉ. Cách tiếp cận này gán các bản ghi với một số lượng lớn các thuộc tính vào một tập nhỏ có quan hệ giữa các nhóm hoặc các đoạn. Quá trình này được thực hiện một cách tự động bởi các thuật toán tách nhóm nhận dạng các tính chất khác biệt của tập dữ liệu và sau đó phân hoạch vùng không gian n_chiều được định nghĩa bởi các thuộc tính tập dữ liệu phụ thuộc vào các biên chia một cách tự nhiên.

a/ Thuật toán tách nhóm điển hình

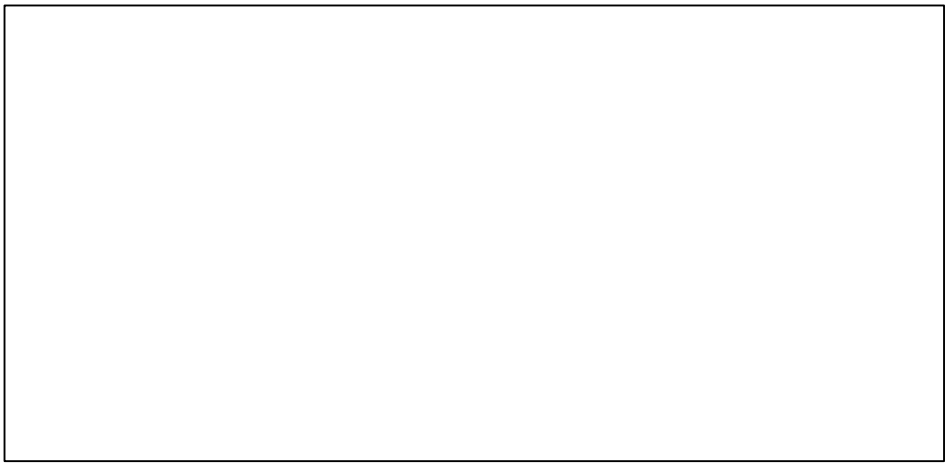
Tách nhóm thực hiện việc nhận dạng nhóm các bản ghi có quan hệ với nhau, các bản ghi này lại có thể được sử dụng như là điểm xuất phát cho việc khai thác các mối quan hệ xa hơn. Kỹ thuật này hỗ trợ cho việc phát triển các mô hình tách nhóm một quần thể tương tự việc tách nhóm các khách hàng dựa trên các tiêu chuẩn của nhân khẩu học. Có thể từ kết quả mong muốn và dựa trên kỹ thuật phân tích chuẩn để xác định được đặc tính của các nhóm. Chẳng hạn, thói quen mua sắm của nhiều nhóm dân cư có thể được so sánh để xác định nhóm nào là mục tiêu của chiến dịch buôn bán mới trong tiếp thị định hướng.

Tách nhóm là phương pháp nhóm những hàng của dữ liệu (bản ghi) theo những hướng giống nhau và vào các mẫu. Trong tách nhóm không có biến phụ thuộc, không có sự mô tả sơ lược về một hướng đặc điểm riêng. Tách nhóm cũng có thể dựa vào mẫu quá khứ ([2]), có nghĩa là, từ các kết quả tách nhóm trước đây để hình thành việc tách nhóm mới.

Kỹ thuật tách nhóm cố gắng tìm sự khác nhau và giống nhau trong tập dữ liệu và phân nhóm những bản ghi giống nhau vào những đoạn hoặc những nhóm. Như vậy, trong tập dữ liệu càng có nhiều sự giống nhau hoặc khác nhau thì tập dữ liệu đó càng được chia nhỏ thành nhiều nhóm. Sau khi dữ liệu đã được tách nhóm, người phân tích sẽ khai thác thông tin và rút ra các tri thức cần thiết thông qua sự giống nhau và sự khác nhau trong các nhóm dữ liệu đó. Chẳng hạn, đối tượng con người thường được phân một cách tự nhiên theo nhân khẩu học thành những nhóm phân biệt theo độ tuổi như: trẻ mới sinh, nhi đồng, thanh thiếu niên, người trưởng thành và người có tuổi. Tính "giống nhau" hoặc "khác nhau" để tách nhóm vừa là kết quả của quá trình tách nhóm vừa là thành tố tham gia vào việc tách nhóm.

Ví du 1.1

Một tập dữ liệu chứa các thông tin về khách hàng có các thuộc tính {"thu nhập", "số con", "Loại ôtô sở hữu"}. Người bán lẻ muốn biết những nét giống nhau tồn tại trong tập khách hàng cơ bản của họ, và như vậy, họ có thể tách ra để hiểu được những nhóm khác nhau về những mặt hàng đã được mua và bán trên thị trường. Người bán hàng sử dụng cơ sở dữ liệu với những bản ghi thông tin về khách hàng và cố gắng tách những nhóm khách hàng. Chẳng hạn, tập dữ liệu có thể chứa đựng rất nhiều khách hàng giầu có mà lại không có con và những khách hàng thu nhập thấp mà có bố mẹ ở cùng. Quá trình khám phá này sẽ tìm ra sự khác nhau có thể được sử dụng để phân chia dữ liệu vào hai nhóm tự nhiên. Nếu tồn tại rất nhiều điểm giống nhau cũng như khác nhau thì tập dữ liệu có thể được chia nhỏ thêm nữa. Chẳng hạn, sau khi phân tích, tập khách hàng được phân thành các nhóm như trong hình 1.



Hình 1. Tách nhóm khách hàng

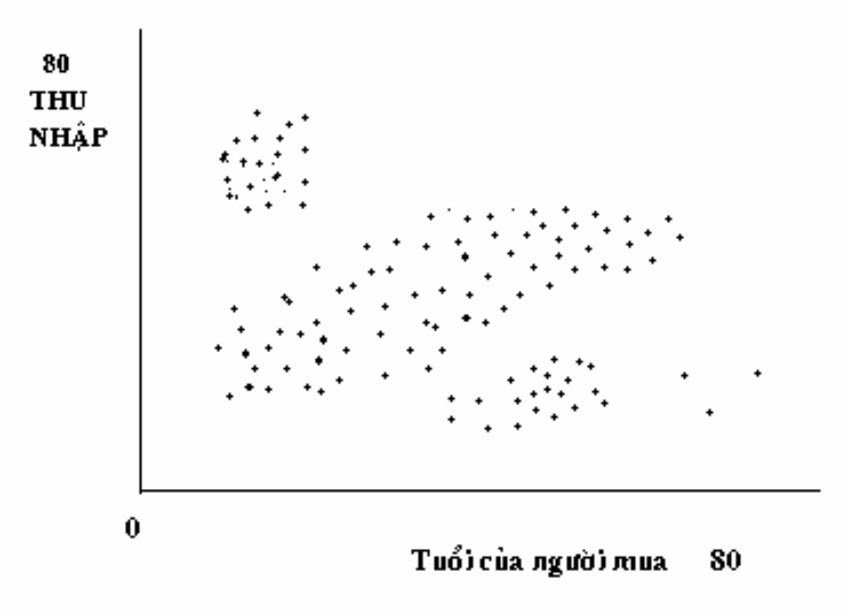
Lược đồ trong hình 1 chỉ ra một cách thức nghiên cứu đoạn mẫu: đưa ra những dữ liệu khách hàng và chia vào các nhóm khác nhau. Lược đồ thể hiện sự cố gắng thu được tri thức về những nhóm dữ liệu trong tập dữ liệu. Từ những nhóm đã được nhận dạng sơ bộ trước đây, một người phân tích có thể hiểu để biểu diễn được sự khác nhau và giống nhau trong những nhóm.

Hình 1 cho thấy có 4 nhóm khách hàng được nhận dạng với tên gọi là Nhóm 1, Nhóm 2, Nhóm 3 và Nhóm 4. Lý do để tách thành những nhóm khác nhau: Nhóm 1 bao gồm những người sở hữu ô tô Luxery, Nhóm 2 bao gồm những người sở hữu ô tô Compact, hai Nhóm 3 và Nhóm 4 bao gồm những người sở hữu ô tô Sedan hoặc Truck. Dữ liệu trong hai nhóm có thể giao nhau, chẳng hạn, trong trường hợp này hai nhóm 3 và 4 có những điểm giống nhau cũng như nhiều điểm khác nhau.

b/ Kỹ thuật hiển thị bằng hình ảnh (Visualization)

Kỹ thuật hiển thị bằng hình ảnh là một phương pháp đơn giản, dễ hiểu nhưng lại rất hữu ích trong việc nhận biết những nhóm dữ liệu khác nhau thông qua việc nhận biết những mẫu ẩn trong dữ liệu. Kỹ thuật này có thể được sử dụng tại thời điểm trước khi tiến hành quá trình khai thác và giúp cho người phân tích thấy sơ bộ về chất lượng của dữ liệu và các mẫu sẽ được tìm thấy trong khoảng nào. Phương pháp hiển thị một cách đơn giản chỉ hiển thị các thuộc tính của dữ liệu lên mặt phẳng theo một cách nào đó. Các kỹ thuật hiển thị đang được phát triển mạnh mẽ và nhanh chóng được cải tiến nhằm cho phép người phân tích lướt qua dữ liệu thông qua không gian dữ liệu nhân tạo. Một kỹ thuật sơ cấp nhưng lại có giá trị là lược đồ phân bố, trong kỹ thuật này thông tin được hiển thị qua hai thuộc tính trên một hệ trục toạ độ hai chiều.

Các phương pháp đơn giản này có thể cho ta rất nhiều thông tin. Lược đồ phân bố có thể được sử dụng để tìm ra các tập dữ liệu con hữu ích trong toàn bộ tập dữ liệu và từ đó ta sẽ tập trung vào phân tích trên các tập con đó trong phần còn lại của quá trình khai thác dữ liệu. Tuy nhiên, các công cụ khai phá dữ liệu (Data Mining) còn được cải tiến để hiển thị dữ liệu thông qua môi trường giao tiếp ba chiều, mỗi chiều tương ứng với một thuộc tính. Hình 2 mô tả một cách hiển thị đơn giản và có thể thông qua phân bố trên mặt phẳng hiện thị để nhận ra được các nhóm dữ liệu.



Hình 2. Một ví dụ về cách hiển thị dữ liệu.

c/ Tách nhóm tối ưu

Một vấn đề đặt ra trong thuật toán tách nhóm là "Nên phân dữ liệu đã cho thành bao nhiêu nhóm thì tối ưu?". Tồn tại các công cụ khác nhau với các cách giải quyết khác nhau giải quyết câu hỏi này. Chẳng hạn, có công cụ cho phép người dùng tuỳ chọn, công cụ khác thì tự động quyết định tuỳ vào từng loại dữ liệu được đưa vào...

Có thể tách thành 2, 3 hay nhiều nhóm. Sau khi tách nhóm sơ bộ như vậy, mỗi nhóm này có thể trở thành vùng tìm kiếm tiếp tục. Ngày nay, tồn tại nhiều cách tiếp cận phân nhóm cho phép người sử dụng quyết định số nhóm trong tập dữ liệu, trong khi đó, cũng tồn tại nhiều cách tiếp cận khác cố gắng đi tới quyết định nhờ việc sử dụng một hoặc nhiều thuật toán.

I.2.2. Thuật toán phân lớp Bayes

a) Thuật toán phân lớp (Classification Algorithm)

Phân lớp là kỹ thuật học có giám sát được ứng dụng phổ biến nhất, sử dụng một tập các mẫu đã được phân loại từ trước để phát triển một mô hình cho phép phân loại thuộc tính của một số lượng lớn các bản ghi.

Theo cách tự nhiên, con người thường có ý tưởng phân chia sự vật thành các lớp khác nhau. Một ví dụ dễ thấy là đối tượng con người thường được phân chia theo độ tuổi thành nhóm khác nhau như: Trẻ sơ sinh, nhi đồng, thiếu niên, thanh niên và người già. Như đã biết, bài toán tách tập đối tượng thành các nhóm khác nhau đã được thuật toán tách nhóm giải quyết. Thuật toán phân lớp đơn giản chỉ là một phép ánh xạ từ một thuộc tính, hoặc một tập hợp các thuộc tính nào đó của dữ liệu sang một miền giá trị cụ thể nào đó. Như trong ví dụ trên, thuộc tính tuổi được ánh xạ sang miền giá trị {“trẻ sơ sinh”, “nhi đồng”, “thiếu niên”, “thanh niên”,...}.

Có thể lấy ví dụ trong các ứng dụng nhằm phát hiện sự gian lận và sự rủi ro về mua bán tín phiếu. Cách tiếp cận này thường xuyên sử dụng thuật toán phân lớp cây quyết định hoặc thuật toán phân lớp dựa trên mạng thần kinh (neural network). Sử dụng thuật toán phân lớp bắt đầu với một tập các cuộc mua bán tập dượt mẫu đã được phân lớp từ trước. Với một ứng dụng phát hiện sự gian lận bao gồm các hồ sơ hoàn chỉnh về cả hoạt động gian lận và hợp lệ, xác định trên cơ sở từng bản ghi một. Đầu tiên, thuật toán sơ bộ phân lớp sử dụng các mẫu đã được phân lớp trước để xác định tập các tham số cần thiết cho việc phân biệt chính xác. Tiếp theo, thuật toán sẽ mã hoá các tham số vào một mô hình được gọi là bộ phân lớp. Cách tiếp cận này chưa tường minh về năng lực của một hệ thống. Ngay sau khi bộ phân lớp có hiệu quả được phát triển, nó được sử dụng trong chế độ có thể đoán trước được để phân lớp các hồ sơ mới vào cùng các lớp đã được định nghĩa sẵn. Chẳng hạn, một bộ phân lớp có khả năng xác định các khoản cho vay có tính rủi ro, có thể được dùng để trợ giúp các quyết định cho các cá nhân vay.

Một ví dụ khác, một cách tiếp cận phổ biến trong doanh nghiệp có mục đích là ”Tôi muốn hiểu điều gì thu hút khách hàng của công ty tôi gắn bó nhiều hơn với công ty“. Để đạt được mục đích đó, giả sử có sẵn hai lớp khách hàng "gắn bó" và "đi khỏi" và với những thông tin có sẵn về khách hàng, cần nhận ra

được đặc trưng từng loại nói trên để có được chính sách tiếp thị tốt hơn. Từ các bảng dữ liệu quá khứ có thể dự đoán về hai lớp đối tượng "gắn bó" và "đi khỏi" nếu vẫn theo chính sách tiếp thị trước đây.

Cột tên trường	Kiểu dữ liệu	Kiểu giá trị	Mô tả
Số_hiệu_khách_hàng	Số	Các giá trị duy nhất	Trường mã phân biệt mỗi khách hàng
Thời_gian_mua_bán	Số	Các giá trị nguyên	Những ngày một khách hàng đến với công ty
Sử_dụng_trực_tuyến	Ký tự	Rất cao, Cao, Vừa, Thấp,Rất_thấp	Số phút được khách hàng sử dụng trong tháng trước
Xu_hướng	Ký tự	Tăng, Tăng_đa_mức, Như_trước, Giảm_đa_mức	Mức độ tăng giảm khách hàng thường xuyên dưới 6 tháng
Trạng_thái	Ký tự	Cao, Được, Thấp, Chưa_rõ	Kết quả điều tra thống kê khách hàng
Kiểu_khách_hàng	Ký tự	Gắn_bó, Đi_khỏi	Khách hàng trung thành với công ty hay đến với công ty cạnh tranh.

Bảng 1. Mô tả đặc trưng của tập dữ liệu khách hàng

Bảng 1 trên đây cho biết tập dữ liệu quá khứ về khách hàng, có các trường với giá trị và kiểu của nó. Chẳng hạn, cột ***Kiểu_khách_hàng*** là cột gồm những bản ghi biểu thị những khách hàng trong quá khứ là trung thành hay nghiêng về công ty cạnh tranh (định rõ từng hàng trong bảng với giá trị ***Gắn_bó*** hoặc ***Đi_khỏi***).

Chú ý, xây dựng mô hình khách hàng đòi hỏi một sự hiểu biết trước về người khách hàng nào là trung thành (***Gắn_bó***) và người nào là không trung thành (***Đi_khỏi***). Kiểu khai thác này được gọi là “học có giám sát” bởi vì mẫu đào tạo được gắn nhãn với các lớp thực sự (***Gắn_bó*** hoặc ***Đi_khỏi***). Cột ***Kiểu_khách_hàng*** được xác định như là một kết quả ra hoặc như là biến phụ thuộc nếu nó được sử dụng như một phần cơ bản của nghiên cứu về bảng dữ liệu khách hàng.

b) Thuật toán phân lớp Bayes

Theo phương pháp Bayes, để cực đại hoá hàm tiện ích U nào đó phụ thuộc vào tác động A và một trạng thái đã biết song chưa đầy đủ của thế giới H, chúng ta đưa ra tác động mà hy vọng tác động đó sẽ làm cực đại hàm tiện ích U nói trên khi tính đến mọi khả năng của thế giới H. Áp dụng trong bài toán phân lớp: Tạo ra sự phân lớp A đưa đến độ chính xác hy vọng U là cực đại với điều kiện đã xem xét trên mọi giả thiết có thể có trong không gian giả thiết của thuật toán học. Trong thực tế, thuật toán chỉ tính được trong một tập con được gọi là "tốt" của không gian giả thiết. Giả sử c là một lớp, τ là tập các giả thiết sinh ra của thuật toán học, x là ví dụ test, $\bar{x}$ là ví dụ cần dạy. Ta cần gán c cho x để cực đại biểu thức:

$$p(c|x,\tau) = \sum_{T in \tau} p(c|x,T)\, p(T|\bar{x}) \qquad (1.1)$$

Điều này có nghĩa là chúng ta phải dự đoán xác xuất hậu nghiệm $p(T|\bar{x})$ của mỗi mô hình học và phải ước lượng một cách chính xác $p(c|x,T)$. Chúng ta xem xét tập con của các luật trong tập các luật của lớp i mà đã thoả mãn ví dụ test x. Độ chính xác của luật chính xác nhất trong đó tập con được sử dụng cho $p(c|x,T)$.

Các hạng thức khác trong phương trình (1.1) là xác suất hậu nghiệm của cây $p(T|\bar{x})$ có thể được tính toán khi sử dụng:

$$p(T|\bar{x}) \alpha\, p(T) \times \prod_{k=1}^{V} \frac{B(n_{1k} + \alpha_1, n_{2k} + \alpha_2)}{B(\alpha_1, \alpha_2)} \qquad (1.2)$$

ở đây $p(T)$ là ưu tiên của cây, B là hàm Beta*, V là số lá của cây, α_1 và α_2 là tham biến và $n_{i,j}$ là kí kiệu số ví dụ cần dạy của lớp i ở lá thứ j của cây. Bên cạnh đó nó còn được sử dụng để phân lớp.

Trong mỗi bài toán ứng dụng cụ thể, việc xác định các công thức tính toán xác suất tiên nghiệm và xác suất hậu nghiệm đối với (1.1) và (1.2) là một trong những nội dung cơ bản nhất của việc ứng dụng phân lớp Bayes.

Trong chương 4 của luận văn sẽ trình bày quá trình giải quyết một loại bài toán phân lớp trong một cơ sở dữ liệu full-text. Các xác suất trong mô hình này thường được biểu diễn dưới dạng tỷ số các tần suất.

I.2.3. Thuật toán phân lớp "k_người láng giềng gần nhất" (k-nearest neighbour)

Cho tập hợp đối tượng Ω, trên Ω có một hàm khoảng cách μ nào đó. Cho tập hợp các mẫu Ω_o đã biết trước và một phân hoạch trên Ω_o trong đó mỗi lớp được đặc trưng bởi một tập con của Ω_o theo phân hoạch nói trên.

Bài toán phân lớp đối với đối tượng w có thể được giải quyết nhờ ***thuật toán người láng giềng gần nhất***. Theo thuật toán này, tìm phần tử w_o của Ω_o thỏa mãn điều kiện:

$$\mu(w, w_o) = \min \{\mu(w, u): u \in \Omega_o\}$$

Lớp được gán cho đối tượng w chính là lớp mà w_o đã thuộc vào.

Tình huống sau đây được đặt ra với thuật toán người láng giềng gần nhất là khi tính khoảng cách nhận được nhiều hơn một đối tượng cùng gần w nhất và chúng lại thuộc các lớp khác nhau. Thuật toán k-người láng giềng gần nhất là sự cải tiến của thuật toán người láng giềng gần nhất được mô tả như sau đây. Với một số k đã chọn trước. Tìm k đối tượng thuộc Ω_o gần với w nhất. Đối với mỗi lớp đã cho, lớp nào có nhiều đối tượng tham gia vào k đối tượng đã tính thì khẳng định đó là lớp cần phân w vào.

Một số nội dung sau đây cần được đặt ra với thuật toán k-người láng giềng gần nhất:

- Việc xác định khoảng cách μ. Khoảng cách nói trên được chọn tùy thuộc vào nội dung của bài toán phân lớp. Chẳng hạn, trong bài toán học mô tả phức HYDRA (được trình bày cụ thể trong chương 2), khoảng cách Ls được chọn theo công thức:

$$ls_{ij}=ls(p,n,p_0,n_0) \approx \frac{(p+1)/(p_0+2)}{(n+1)/(n_0+2)}$$

ở đây p_0 và n_0 tương ứng kí hiệu số các ví dụ dạy tích cực và đối ngẫu (của lớp i) trong toàn bộ tập dữ liệu còn p và n là các ký hiệu tương ứng với p_0 và n_0 song liên quan đến luật.

- Cỡ của số k cũng có ảnh hưởng đến chất lượng của thuật toán: k quá bé thì ảnh hưởng đến độ tin cậy của thuật toán, còn khi k quá lớn sẽ tạo ra độ phức tạp tính toán cao mà độ tin cậy lại không tăng một số đáng kể. Một số phương pháp thống kê có thể được sử dụng để xác định giá trị k thích hợp.

Trong nhiều trường hợp, thuật toán k-người láng giềng gần nhất cho một phương pháp khả thi, hiệu quả tốt mà không quá phức tạp. Mặt khác, khi áp dụng thuật toán người ta thường xem xét "độ gần nhau" giữa các đối tượng thay cho việc xem xét "khoảng cách" giữa chúng.

Một *biến dạng của thuật toán k-người láng giềng gần nhất* thường được sử dụng trong bài toán phân lớp được diễn tả theo tiến trình như sau:

- Lấy một số dương gán tương ứng cho mỗi lớp, được gọi là ngưỡng của lớp,

- Lấy ngẫu nhiên k đối tượng trong tập các đối tượng mẫu,

- Tính độ thuộc của đối tượng cần phân lớp tương ứng với mỗi lớp đã cho,

- Với từng lớp đối tượng, so sánh giá trị kết quả tính toán độ thuộc với ngưỡng: nếu vượt quá ngưỡng thì kết quả đối tượng được phân vào lớp đó; trong trường hợp ngược lại thì xem xét với lớp tiếp theo.

Biến dạng như trên của thuật toán k-người láng giềng gần nhất thường đạt độ chính xác không cao song lại đưa đến tốc độ tính toán nhanh. Tốc độ hoàn

thành thuật toán phụ thuộc nhiều vào việc chọn "ngẫu nhiên" k đối tượng mẫu được coi là "láng giềng gần nhất".

I.2.4. Thuật toán cây quyết định (Decision Tree)

Tạo cấu trúc cây để biểu diễn dữ liệu đã được sử dụng rất nhiều trong khoa học máy tính.

Trước hết chúng ta xem xét một cách đơn giản để xây dựng một cây quyết định (có rất nhiều cách để xây dựng một cây quyết định). Một số cây quyết định mang một số đặc trưng sau đây:

+ Cây quyết định chỉ có hai nhánh tại một nút trong.

+ Cây quyết định sử dụng kết hợp các cách tiếp cận.

Các cây quyết định có khác nhau nhưng đều qua một quá trình xử lý tương tự nhau, chúng được ứng dụng trong nhiều thuật toán học khác nhau để xác định nhóm và phân loại sự quan trọng của các biến khác nhau.

Các bước trong quá trình xây dựng cây quyết định:

Bước 1: Các biến được chọn từ nguồn dữ liệu. Từ các biến được biểu diễn trong nguồn dữ liệu, một biến phụ thuộc được chọn ra bởi người sử dụng. Chẳng hạn, ***Biến phụ thuộc*** là số người mắc phải bệnh cao huyết áp, ***biến vào*** là chiều cao, cân nặng...

Bước 2: Các biến có ảnh hưởng đến kết quả sẽ được kiểm tra. Một quá trình sáng tạo sẽ nhóm các biến phụ thuộc trên các khoảng giá trị mà các biến thuộc vào. Ví dụ, giá trị biến Chiều_cao sẽ gộp thành hai nhóm (143-166 cm) và (167-190 cm). Việc xác định chia ra thành 2 nhóm, 3 nhóm, hay 4 nhóm phụ thuộc vào chức năng kiểm tra được sử dụng để nhóm dữ liệu.

Bước 3: Sau khi giá trị các biến đã được gộp thành các nhóm, một biến có khả năng dự đoán kết quả tốt nhất sẽ được chọn ra để tạo các nút lá của cây. Thông tin về tần suất thường được sử dụng để biểu diễn số lần xuất hiện của biến phụ thuộc.

CHƯƠNG 2. HỌC MÁY MÔ TẢ PHỨC

II.1. MÔ HÌNH HỌC MÁY MÔ TẢ PHỨC

II.1.1 Sơ bộ về mô hình học máy mô tả phức

Một trong những bài toán quan trọng trong học máy có giám sát là bài toán rút gọn được số lỗi của học máy. Một trong những hướng nghiên cứu quan trọng về học máy nhằm giải quyết bài toán trên là mô hình học máy mô tả phức. Theo hướng này đã có rất nhiều công trình nghiên cứu thành công, đặc biệt là các công trình của nhóm nghiên cứu về học máy tại trường Đại học Tổng hợp California, Ivrin ([5-13]).

Học máy mô tả phức tiếp nhận đầu vào là một tập các khái niệm phân hoạch tập dữ liệu (qua đó phân hoạch tập đối tượng), các ví dụ mẫu của mỗi khái niệm và một tập các "khái niệm nền". Khái niệm nền là khái niệm được coi là biết trước, được công nhận rộng rãi và không cần giải thích. Đầu ra của mô hình là các mô tả cho mỗi lớp theo khái niệm. Những mô tả này sau đó được sử dụng để phân lớp một ví dụ đối với một khái niệm. Phương pháp học máy mô tả phức khái niệm sẽ tương ứng một khái niệm với một tập các luật và cho phép kết hợp những mô tả khái niệm liên quan đến nhiều tập dữ liệu khác nhau. Hình 2.1 minh họa về mô hình đơn và các mô hình phức trong bài toán học máy.

Bằng thực nghiệm, Ali K. và Pazzani M. [5] đã chỉ ra rằng kết quả phân lớp theo mô hình học máy mô tả phức đạt độ chính xác cao hơn nhiều khi so sánh với mô hình dựa trên mô tả khái niệm đơn lẻ đối với cùng bộ dữ liệu và cùng áp dụng thuật toán tìm kiếm leo đồi ngẫu nhiên theo bề rộng. Các tác giả nói trên chỉ ra rằng các kết quả nghiên cứu theo các mô hình cụ thể như dự đoán cấu trúc lưới phần tử hữu hạn, học theo nội dung King-Rook-King (viết tắt là KRK), phân loại khối tài liệu v.v. cho kết quả là học máy mô tả khái niệm phức làm tăng độ chính xác cho mô tả khái niệm không có ưu tiên (tức là, cây quyết

định) mà theo đó hoặc mỗi mô tả là một tập các luật hoặc học mô tả các khái niệm phức với những khái niệm dạng quan hệ (khái niệm tương ứng với những tập các luật dạng quan hệ nếu nó có thể được mô tả thông qua việc sử dụng các quan hệ này, xem mục II.2.2).

Các nghiên cứu mô hình học máy mô tả phức [5-11] đã khái quát hoá được các điều kiện mà theo đó, học máy mô tả phức có lợi hơn so với các mô hình học máy trước đây theo tiêu chuẩn đảm bảo độ chính xác. Hơn nữa, thông qua việc sử dụng lý thuyết xấp xỉ Bayes, yêu cầu về độ chính xác tối ưu đã giải quyết được vấn đề tạo ra sự phân lớp dựa trên kết quả thăm dò từ tất cả các giả thiết, trong đó kết quả thăm dò được định giá trị bằng xác suất hậu nghiệm của giả thiết. Trong thực tế, chỉ có thể sử dụng một phần nhỏ các giả thiết (do trong hệ thống bao gồm số lượng lớn các đối tượng), vì vậy phải tìm ra được một số lượng nào đó các mô tả thích hợp nhất. Các nghiên cứu nói trên cũng đã chỉ ra rằng: việc sử dụng tập luật phức là hữu hiệu hơn so với việc sử dụng các luật phức riêng biệt. Điều đó được giải thích như sau. Các phương pháp sử dụng luật phức mô hình hoá mỗi lớp bằng luật đơn, liên kết luật. Tuy nhiên tồn tại rất nhiều lớp không thể mô hình hoá chính xác chỉ với những luật đơn thông qua những tập hợp khái niệm nền cho trước.

Trong các mô hình học máy mô tả phức đầu tiên (mô hình FOIL - mục II.3.1, và FOCL - mục II.3.2) chưa xây dựng việc học máy với tập luật phức cho mỗi lớp. Kết quả cho thấy rằng, nhiều khái niệm không thể được mô phỏng một cách chính xác bởi chỉ các luật riêng, và điều đó đã chỉ ra phương hướng xây dựng phương pháp sử dụng tập luật với khả năng cho độ chính xác cao hơn trong việc học các khái niệm như vậy. Ngoài ra, cách học như thế vẫn còn cho khả năng làm việc tốt tương đương đối với các khái niệm còn lại (ngoài những khái niệm dùng để đối sánh với mô hình đơn). Trong các công trình [5-13], thông qua thực nghiệm, các tác giả đã minh chứng cho các khẳng định trên đây. Những khái niệm chỉ có thể mô phỏng một cách chính xác bởi việc sử dụng không ít

hơn một luật thì cần có sự phân rã phức tương ứng với một tập cho trước các khái niệm nền. Chính xác hơn nữa, một khái niệm được gọi là chứa đựng sự phân rã phức nếu không có các luật kết nối thuần túy cho các khái niệm đó tương ứng với một tập xác định các biến và ngôn ngữ giả thiết được nhất quán với tất cả các ví dụ và phản ví dụ của khái niệm này. Các mô hình học máy HYDRA và HYDRA-MM (mục II.3.3 và mục II.3.4) đã thể hiện được các nội dung về tập luật phức cho mỗi lớp.

Hai đặc trưng chính của học máy mô tả phức khái niệm là:

- Mỗi khái niệm được xác định thông qua một tập các luật mà không phải là dạng luật đơn như học máy thông thường,
- Mỗi khái niệm (dạng trình bày đặc biệt là lớp) không chỉ được học máy trong chỉ một tập dữ liệu mà được học máy thông qua nhiều tập dữ liệu có liên quan đến khái niệm nói trên. Theo Ali K. và Pazzani M. [5], các thực nghiệm về học máy mô tả phức thực tế làm việc với không quá năm tập dữ liệu đối với một khái niệm.

II.1.2. Một số nội dung của học máy mô tả phức

Ba nội dung chính trong học máy mô tả phức là: lựa chọn kiểu của mô hình, phương pháp để đưa ra những mô hình phức từ theo một tập dữ liệu và phương pháp để kết hợp chứng cứ từ các mô tả (theo nhiều tập dữ liệu).

a/ Lựa chọn kiểu mô hình

Mô hình đơn | Mô hình các mô tả phức | Mô hình các tập các mô tả phức

Hình 2.1. So sánh ba thuật toán trên cùng một miền, trong đó lớp thứ nhất đang được quan tâm (vùng chứa trong các hình tròn đậm nét) chứa hai

đoạn tách nhau (hai đường tròn đậm nét). Các đường mảnh hơn chỉ rõ tập phủ bởi các luật học theo ba thuật toán này.

Trong các công trình nghiên cứu, đặc biệt là nghiên cứu của Ali K., Brunk C. và Pazzani M. trong [8], các tác giả xem xét vấn đề chọn lựa việc học với các luật phức hay các tập luật phức. Các tác giả đã chỉ ra rằng có hai động cơ định hướng phải học với tập luật phức. Thứ nhất, qua nhiều thử nghiệm được tiến hành nhằm học một luật cho mỗi phân rã của mỗi lớp đã khẳng định được là kết quả đã tốt hơn song cũng cho thấy cần phải cải tiến mô hình. Thứ hai, mỗi sự phân rã phụ (một phân rã có thể tương ứng với một phần nhỏ các ví dụ của một lớp) được mô hình hoá bởi một luật. Hình 2.1 trên đây minh hoạ một khái niệm chứa đựng một sự phân rã chính (đường đậm nét) và một sự phân rã phụ (đường mảnh nét). Những đường mảnh chỉ dẫn vùng được gộp vào của luật học mà tính xấp xỉ của phân rã được nhấn mạnh. Hình vẽ bên trái ở đây (mô hình đơn) minh hoạ vấn đề học máy sử dụng kỹ thuật chia nhỏ và chế ngự (tức là mô hình FOIL, xem dưới đây) trong đó học các luật xấp xỉ cho sự phân rã đầu tiên và sau đó loại trừ khỏi tập dạy những ví dụ phủ bởi luật đó nhằm mục đích học những luật kế tiếp. Trong phương pháp chia nhỏ và chế ngự, mỗi luật cố gắng mô hình hoá một phân rã đối với khái niệm. Hình vẽ ở giữa (luật phức) minh hoạ cho phương pháp học theo các luật phức, mỗi luật cố gắng mô hình hoá toàn bộ khái niệm (cả hai sự phân rã). Hình vẽ này cho thấy phương pháp học đang cố gắng phủ cả hai phân rã với chỉ một luật. Bởi vì điều này không thể làm tốt được với các hạng thức của một tập xác định các khái niệm nền, kết quả là các luật học máy chung chung và phủ khu vực ngoài của lớp thứ nhất (đường ô van chéo). Vì vậy nó sẽ cho kết quả không như mong muốn đối với những ví dụ test của lớp thứ hai. Cuối cùng, hình bên phải (học với tập các luật phức) cho thấy mô hình học máy theo tập luật phức áp dụng chiến lược chia nhỏ và chế ngự nhiều lần, học xấp xỉ nhiều hơn cho mỗi phân rã. Do vậy, mô hình tập luật phức đáp ứng được cả tiêu chuẩn cho xấp xỉ phức lẫn tiêu chuẩn cho mô hình các phân rã phụ.

Như vậy, các mô hình dần được cải tiến từ mô hình mô tả phức đối với cùng một tập dữ liệu tới mô hình mô tả phức đối với nhiều tập dữ liệu. Trong phần dưới đây sẽ phác hoạ những nét cơ bản nhất về các loại mô hình này và trong các mục sau, nội dung các mô hình trên sẽ được trình bày chi tiết hơn.

b/ Các phương pháp mô tả phức theo một tập dữ liệu

Trong các mô hình học máy mô tả phức, các tác giả đã xem xét vấn đề lựa chọn phương pháp để đưa ra mô tả phức trên chỉ một tập dữ liệu. Những phương pháp đưa ra sự mô tả khái niệm phức là: tìm kiếm chùm [5, 19], can thiệp người sử dụng [13], đánh giá chéo n-nếp (n-fold cross validation) [11] và tìm kiếm ngẫu nhiên.

Phương pháp tìm kiếm chùm có nội dung thực hiện việc thu thập N luật tốt nhất theo xếp hạng thông qua một độ đo thu thập thông tin nào đó [17]. Bởi vì đây là phương pháp luật phức cho nên còn chứa đựng một số thiếu sót về tỷ lệ lỗi học máy. Trong [17], Shankle W. S., Datta P., Pazzani M. và Michael D. đã cho các đánh giá cụ thể về sai sót học máy của phương pháp này.

Phương pháp dùng sự can thiệp của người sử dụng có nội dung cho phép người sử dụng kiểm tra các điểm nút quyết định quan trọng nhất được đưa ra đối với việc học một cây quyết định và sau đó cho phép người sử dụng quyết định nên dùng nút nào học các cây đặc biệt. Hạn chế của phương pháp này là người sử dụng chỉ có thể được tham khảo một vài lần.

Phương pháp đánh giá chéo n-nếp có nội dung phân chia tập dạy thành nhiều tập con cân bằng nhau sau đó sử dụng một trong số các tập con để tạo ra n tập luật. Trong phương pháp này, cần tách từng tập con một: tập con thứ i được loại bỏ khỏi tập dạy khi học tập luật thứ i cho một khái niệm. Theo Shankle W. S., Datta P., Pazzani M. & Michael D. [17], một số tác giả đã sử dụng một phiên bản của phương pháp này, trong đó việc học sử dụng tất cả các dữ liệu và các luật chỉ được xem xét nếu chúng xuất hiện đa phần trong n tập luật đã được học trước đây.

Phương pháp này có nhược điểm là đầu ra chỉ là một mô hình đơn chứ không phải là một tập các mô hình và hầu hết các tìm kiếm trong học máy mô tả phức đã chỉ ra rằng sẽ không có kết quả tốt khi chưa sử dụng mô hình phức.

Phương pháp tìm kiếm ngẫu nhiên có nội dung nhằm đưa ra được mô tả phức, trong đó tìm kiếm ngẫu nhiên có liên quan đến thay đổi tìm kiếm theo bề rộng. Theo cách như vậy, thay vì phải luôn luôn lựa chọn đường đi tốt nhất, thì thuật toán chỉ ra rằng những đường đi tối ưu (đường đi MAX- BEST, xem nội dung mô hình HYDRA-MM) là lựa chọn tiếp theo và sự lựa chọn ngẫu nhiên có căn cứ từ những tập hợp của các đường đi như vậy được thực hiện. Phương pháp này có hạn chế là đòi hỏi ước đoán logic về giá trị của đường đi tối ưu MAX-BEST nhưng lại có ưu điểm là tạo ra các mô tả với sự phân lớp cuối cùng chính xác hơn những phân lớp tiến hành bởi kết hợp minh chứng từ mô tả được học bởi phương pháp đánh giá chéo n-nếp ([5]).

c/ Kết hợp chứng cứ

Phương pháp kết hợp chứng cứ liên quan đến vấn đề minh chứng đối với các mô tả và được áp dụng trong các mô hình học máy mô tả phức với nhiều tập dữ liệu. Theo phương pháp này, người ta xem xét hai cách thức kết hợp minh chứng: dạng phần dư của luật Bayes và đánh giá độ tin cậy theo xác suất hậu nghiệm của mô hình đưa ra các dữ liệu dạy. Trong mô hình HYDRA-MM (xem mục II.3.4), các nội dung này được trình bày cụ thể hơn.

II.2. MỘT SỐ KHÁI NIỆM VÀ TRÌNH BÀY TRI THỨC TRONG HỌC MÁY MÔ TẢ PHỨC

II.2.1. Một số khái niệm

Khẳng định (vị từ: predicate) là một hàm Boolean. Khẳng định có thể được xác định theo cách ***dàn trải*** dưới dạng một danh sách các bộ theo đó khẳng định là true, hoặc theo cách ***bổ sung***, như là một tập các luật Horn để tính toán khẳng định là true hay không.

Chẳng hạn, các khẳng định theo dạng dàn trải có dạng ***màu*** (X, Y), ***đỏ*** (Y) đối với các ví dụ X, Y nào đó. Luật Horn sẽ được giới thiệu ở ngay dưới đây.

Literal là một khẳng định hoặc là đối của nó (tức là hàm Boolean mà là phủ định của khẳng định). Literal là khẳng định không âm được gọi là literal dương. Literal là phủ định của khẳng định được gọi là literal âm.

Luật Horn bao gồm một đầu luật (chính là một khẳng định), dấu kết nối "←" và một thân luật. Thân luật là một liên kết giữa các literal. Một luật Horn có dạng:

$P \leftarrow L_1, L_2, ...$ trong đó, P là một khẳng định, các L_i là các literal.

Luật đối với P là kết nối các luật Horn có đầu luật là P.

Một k-bộ là dãy k hằng kí hiệu bởi $(a_1, a_2, ..., a_k)$. ***Ngữ nghĩa của một luật*** có khẳng định đầu luật với k đối số là tập các k-bộ bảo đảm khẳng định. Một k-bộ được gọi bảo đảm một luật nếu nó bảo đảm một luật Horn xác định luật đó. Một k-bộ bảo đảm một luật Horn nếu tồn tại ánh xạ φ của các biến trong đầu luật vào bộ và một phần mở rộng φ' của các biến trong literal dương của thân luật vào các hằng sao cho đối với mỗi literal trong thân luật thì theo φ' đi tới kết quả là một literal phù hợp.

II.2.2 Trình bày tri thức trong học máy mô tả phức

a/Mô tả quan hệ

Có rất nhiều những khái niệm không thể học được một cách dễ dàng bởi mô tả thuộc tính giá trị nhưng lại có thể hiểu dễ dàng thông qua những mô tả dạng quan hệ. Những luật mang thuộc tính giá trị gồm các literal (chẳng hạn, >(Tuổi, 50)) thì có thể chỉ so sánh với một biến (chẳng hạn, Tuổi) đối với một giá trị (chẳng hạn, 50). So sánh biến với biến là không hợp lệ. Ví dụ dưới đây mô tả về luật mang thuộc tính giá trị (tên bắt đầu bởi một chữ hoa là kí hiệu một biến: Tuổi, Mức_độ ...):

ung_thư_vú(Tuổi,..., Mức_độ) ← >(Tuổi, 50), >(Mức_độ, 3)

Luật này kết luận rằng người phụ nữ được biểu thị bởi một tập hợp các giá trị của các biến (Tuổi,..., Mức_độ) bị ung thư vú nếu bà ta hơn 50 tuổi và mức độ trầm trọng của bệnh lớn hơn 3. Chú ý rằng, ***dấu quan hệ ">"*** chính là ***một khái niệm nền***. Trong nhiều trường hợp, để dễ nhìn hơn, luật Horn trên đây được viết lại là:

ung_thư_vú(Tuổi,..., Mức_độ) ← (Tuổi, > 50), (Mức_độ, >3)

Trình tự kiểm nghiệm một luật Horn được diễn tả như sau. Lần lượt, luật đó nhận một ví dụ là một dãy các giá trị của biến và kiểm tra các giá trị này có thoả mãn các điều kiện hay không. Nếu đúng, chúng ta nói rằng luật bao gồm hoặc đi đôi với ví dụ và ví dụ thoả mãn luật (còn được gọi là ***ví dụ tích cực***). Để làm rõ thuật ngữ đã được dùng trước đây thì nhiệm vụ học là phân lớp các ví dụ đối với một trong hai lớp (ung_thư-vú, không_ung_thư_vú) và dấu > là ví dụ về khái niệm nền. Trong trường hợp này, vì chỉ một thực thể có liên quan đến luật với giá trị thuộc tính nên đôi khi luật này được viết dưới dạng sau (đầu luật không có biến):

ung_thư_vú ←Tuổi>50, Mức độ >3

Hơn nữa, luật quan hệ có thể liên quan tới nhiều hơn một thực thể, chẳng hạn (chú ý có sự phân biệt giữa khẳng định ***tuổi*** với biến ***Tuổi***):

ung_thư_vú(W1)←tuổi(W1,Tuổi),>(Tuổi,50), mẹ(W1,W2), ung_thư_vú (W2)

Luật quan hệ này kết luận rằng người phụ nữ (thực thể W1) là bị ung thư vú nếu bà ta hơn 50 tuổi và mẹ bà ta (thực thể W2) bị ung thư vú. Luật này sử dụng các quan hệ hai ngôi ***tuổi, >*** và ***mẹ***, và một quan hệ một ngôi ung_thư_vú. Luật này là luật đệ quy bởi vì khái niệm ung_thư_vú vừa như là kết luận vừa như là điều kiện của luật.

Việc học quan hệ tổng quát được định nghĩa như sau:

• Input:

(1) tập các ví dụ thuộc một tập các lớp đặc biệt (tức là ung_thư_vú, không_ung_thư_vú) mà phân chia không gian các ví dụ,

(2) tập các quan hệ nền của các khái niệm nền (tức là ***mẹ***(-,-)) trong đó những định nghĩa mở rộng đầy đủ được cung cấp cho thuật toán học máy. Một định nghĩa mở rộng là tập hợp tất cả các dãy về độ dài của hai kí hiệu mà ở đó các mối liên hệ "người mẹ " là có thực. Ví dụ (Hương, Hà) sẽ là thác triển xác định của ***mẹ*** nếu Hà là mẹ của Hương.

• Output:

Xây dựng một mô tả khái niệm cho mỗi lớp sử dụng kết hợp các quan hệ nền.

Một luật dạng class-a(X,Y) ←b(X),c(Y) bao gồm phần đầu (class-a(X,Y)) và phần thân là phép hội các literal (b(X),c(Y)). Phân lớp một ví dụ kiểm tra mới được tiến hành như sau: cố gắng tạo ra ví dụ phù hợp với mỗi luật cho mỗi lớp. Hy vọng rằng chỉ những luật cho một lớp sẽ phù hợp với ví dụ và do đó nó sẽ được phân vào lớp đó. Tuy nhiên, vấn đề nảy sinh là ví dụ kiểm tra lại hoặc phù hợp với những luật của quá một lớp hoặc lại không phù hợp với bất kỳ luật nào của bất kỳ một lớp nào (liên quan đến ***tính nhập nhằng*** hoặc ***tính không đầy đủ*** của tập luật trong học máy).

b/ Phân lớp Bayes

Chương 1 đã trình bày thuật toán phân lớp Bayes. Chúng ta biến đổi phương trình (1.2) trong chương 1 để sử dụng vào việc phân lớp qua tập hợp luật. Một tập luật có thể nhận thấy được nhờ cây quyết định nhị phân một phía với các phép thử phức. Tại các điểm nút của cây, mỗi phép thử tương ứng với thân một luật. Các dạng khác nhau của các luật sẽ tương ứng với các cây khác nhưng tất cả các cây đó sẽ phục vụ cho sự phân lớp đặc trưng. Trong [6] đã lưu ý rằng xác xuất hậu nghiệm cũng có thể sử dụng như một metric bổ sung trong quá trình học máy. Metric được sử dụng trong học máy được lựa chọn thêm vào nút quy định vào cây để xác suất hậu nghiệm của cây mới là cực đại. Với học máy bởi cây nhị phân từ hai lớp theo hệ quả của phương trình (1.2) xác định metric bổ sung.

$$pr_2(n_{11}, n_{21}, n_{12}, n_{22}) = p(T) \times \frac{B(n_{11} + \alpha_1, n_{2,1} + \alpha_2)}{B(\alpha_1, \alpha_2)} \times \frac{B(n_{12} + \alpha_1, n_{22} + \alpha_2)}{B(\alpha_1, \alpha_2)} \qquad (2.1)$$

trong đó n_{11} và n_{21} tức là kí hiệu số ví dụ tích cực và đối ngẫu của nó trong nhánh trái của điểm nút và n_{12}, n_{22} là kí hiệu số nhánh phải. $p(T)$ kí hiệu xác xuất ưu tiên của cây có được từ việc thêm vào điểm nút quy định. Các metric bổ sung này được gọi là metric Bayes. Quá trình học n mô tả khái niệm có khả năng nhất với khả năng xảy ra của chúng được đánh giá một cách tổng thể thay vì việc xử lí kết quả của tìm kiếm theo bề rộng.

Cho $n_{1,i\,j}$ và $n_{2,i\,j}$ tương ứng biểu thị số lượng ví dụ cần dạy tích cực và đối ngẫu được phủ bởi luật thứ j của lớp thứ i và V là tập các luật trong mô hình. Có thể sử dụng phương trình (2.1) để tính xác suất hậu nghiệm $p(M|\bar{x})$ của một mô hình M được học bởi HYDRA (xem mục II.3.3 dưới đây).

$$p(M|\bar{x}) \alpha \; p(M) \times \prod_{ij \in V} \frac{B(n_{1,ij} + \alpha_1, n_{2,ij} + \alpha_2)}{B(\alpha_1, \alpha_2)} \qquad (2.2)$$

Chúng ta xem xét việc dùng lí thuyết Bayes cho các tập luật học máy sử dụng sự tương tự giữa các tập luật và các cây quyết định, thêm vào một điều kiện cho một luật cũng tương tự như bổ sung điều kiện cho những phép thử phức tại các điểm nút quyết định. Do đó, sự thay đổi trong pr_2 (phương trình 2.1) đo sự tăng của xác suất hậu nghiệm như là kết quả của việc bổ sung điều kiện. Khó khăn cho việc sử dụng pr_2 trực tiếp trong các luật học máy ở chỗ: pr_2 là đối xứng vì vậy luật phủ 5(P) trong số 10(P_o) ví dụ tích cực và 1(n) trong số 10(n_o) các ví dụ đối ngẫu sẽ nhận cùng một kết qủa như là luật phủ 5 trong số 10 các ví dụ đối ngẫu và một trong số 10 ví dụ tích cực. Do vậy cần sử dụng một hàm pr_2 đã được biến đổi: luật mà ở đó pr_2 được gán là 0 nếu $P/r \leq P_o/n_o$. Dùng giá trị 1 cho α_1 và α_2 bởi vì giá trị đó đồng nhất với độ ưu tiên được dùng trong luật Laplace về sự kế thừa.

Xác suất hậu nghiệm của mô hình, $p(T|\bar{x}, \bar{c})$ được tính toán như sau (trong công trình của Buntine, 1990) khi sử dụng luật Bayes để viết giá trị:

$$p(T|\bar{x},\bar{c}) \ \alpha \ p(\bar{x},\bar{c}|T) \times p(T) \tag{2.3}$$

p(T) là xác suất tiên nghiệm của mô hình T. Bổ sung một số giả định rằng các ví dụ dạy trong mô hình là độc lập, ta nhận được:

$$p(\bar{x},\bar{c}|T) = \prod_{i=1}^{N} p(x_i, c_i|T) \tag{2.4}$$

ở đây N chính là kích thước của tập dạy. Có thể chia tập hợp dạy thành các tập hợp nhỏ tương ứng với các kiểu khác nhau của các ví dụ dạy. Để coi V như là các tập hợp con và $n_{j,k}$ biểu thị số lượng các ví dụ dạy của lớp j trong tập hợp con thứ k. Do đó, có thể viết:

$$p(\bar{x},\bar{c}|T) = \prod_{k=1}^{V}\prod_{j=1}^{C} \Phi_{j,k}^{n_{j,k}} \tag{2.5}$$

ở đây Φ_{ik} thể hiện xác suất của việc đưa ví dụ đơn của lớp j ở tập hợp con thứ k và C là số lượng lớp. Một vấn đề được chỉ ra sau đó (Buntine, 1990) là sự đóng góp đối với xác suất hậu nghiệm từ tập con thứ k có thể mô hình hoá bởi:

$$\frac{B_C(n_{1,k}+\alpha,\ldots,n_{C;k}+\alpha)}{B_C(\alpha,\ldots,\alpha)} \tag{2.6}$$

ở đây Bc là hàm beta theo thứ nguyên ***c*** và α là thông số biểu thị "độ tin cậy" (trong một số ví dụ) mà phải được đi cùng với tiên đoán tiên nghiệm (1/c) của $\Phi_{j,k}$: đặt các phương trình (2.5) và (2.6) cùng nhau. Từ hai phương trình đó nhận được:

$$p(\bar{x},\bar{c}|T) = \prod_{k=1}^{V} \frac{B_C(n_{1,k}+\alpha,\ldots,n_{C,k}+\alpha)}{B_C(\alpha,\ldots,\alpha)} \tag{2.7}$$

Bởi vì $p(\bar{x},\bar{c}|T)$ có thể được tính toán, sau đó sử dụng phương trình 2.1, xác suất hậu nghiệm $p(\bar{x},\bar{c}|T)$ có thể được tính, do vậy, xác suất hậu nghiệm kỳ vọng có thể được tính toán. Các giải thích trên đây cho phép tính toán xác suất hậu nghiệm của mô hình là cây quyết định. Điều đó phụ thuộc sự quan sát theo đó các ví dụ dạy được chia thành V tập hợp con rời nhau. Khi bổ sung nó cho cho

các kiểu của các mô hình được xem xét, một mô tả tách biệt thì được học cho mỗi lớp bằng quan sát mô hình như vậy chia ví dụ dạy C lần (số lượng của các lớp). Sau đó, để tính toán xác suất hậu nghiệm của mô hình như vậy, có thể đơn giản là lấy trung bình hình học của các xác suất hậu nghiệm của các mô tả lớp:

$$p(T|\overline{x},\overline{c}) \ \alpha \ p(T) \times \Big(\prod_{i=1}^{C} \prod_{i,j \in R_i} \frac{B(n_{1,ij}+\alpha, n_{2,ij}+\alpha)}{B(\alpha,\alpha)} \Big)^{1/C} \qquad (2.8)$$

R_i biểu thị mô tả lớp thứ i trong mô hình T và ij chỉ ra các luật riêng. Do vậy, trong phạm vi mô tả lớp cho lớp thứ i, các lớp được nhóm thành 2 lớp giả (lớp i được gọi là lớp "tích cực", tất cả các lớp khác được kết hợp thành lớp "tiêu cực"), và có thể sử dụng k=2 ở phương trình 2.6 để thu được các số hạng hàm beta ở phương trình 2.8.

c) Chiến lược chia nhỏ và chế ngự

Các phương pháp học máy mô tả phức sử dụng chiến lược điều khiển chia nhỏ và chế ngự dựa trên FOIL (xem mục II.3.1). Trong chiến lược này, các luật được học một lần. Ví dụ cần dạy được phủ bởi một luật chuyển từ tập dạy và các luật kế tiếp sau được học để phủ lên tất cả các ví dụ còn lại.

Một luật cho một lớp xác định như class-a(V_1, V_2) thì được học bởi một chiến lược tìm kiếm theo bề rộng:

- Bắt đầu với một thân luật rỗng mà phủ toàn bộ ví dụ tích cực và tiêu cực còn lại.

- Xem xét tất cả các literal mà có thể thêm vào thân luật và định giá thông tin thu được bằng cách bổ sung của nó cho thân của luật có thể bao trùm nhiều ví dụ tích cực và loại bỏ nhiều ví dụ tiêu cực. Quinlan ([18]) định nghĩa nội dung thông tin của mỗi luật phủ p_0 ví dụ tích cực và n_0 ví dụ tiêu cực như sau:

$$l(p_0,n_0)=\log_2 \frac{p_0}{p_0+n_0}$$

và thông tin thu được bởi bổ sung thêm literal vào thân một luật như vậy để bây giờ luật phủ p_1 ($\leq p_o$) ví dụ tích cực và n_1($\leq n_o$) ví dụ tiêu cực là:

$$p_1 *(l\ (p_0,n_0)-l\ (p_1,n_1))$$

Chiến lược tiếp tục bổ sung literal để loại trừ ví dụ đối ngẫu cho đến khi kết luận không còn chứa bất kỳ một ví dụ đối ngẫu nào hoặc không có literal nào cho phép thu thêm những thông tin tích cực (các điều kiện tiếp theo có thể xẩy ra khi các tập hợp dữ liệu bị nhiễu). Các ví dụ tích cực đã được luật bao trùm sẽ được loại khỏi tập dạy và tiếp tục xử lý để học các ví dụ còn lại, quá trình kết thúc khi không còn ví dụ tích cực nào.

Sau đó việc học máy không thực hiện đối với từng luật cho mỗi lớp mà học một tập hợp luật cho mỗi lớp và do đó, mỗi tập hợp có thể so sánh để phân lớp các ví dụ test. Trong [8] đã chỉ ra rằng điều này cho phép học máy chính xác hơn trong trường hợp dữ liệu bị nhiễu. Hơn nữa, cần xem xét tới mức độ đầy đủ về mặt lôgic (trong thuật toán dùng ls là độ đo tin cậy của việc phân lớp) đối với mỗi luật. Đã cải tiến việc xác định khoảng cách (ls-nội dung) để sắp xếp các literal tương ứng với phạm vi bao phủ các ví dụ tích cực là tiến bộ hơn so với xác định khoảng cách trước đây. Tuy nhiên những cải tiến trên không áp dụng được cho các mô hình dữ liệu lớn.

Đối với những mô hình dữ liệu lớn, thuật toán học cần kết hợp nhiều giải pháp khác nhau để tăng cường độ chính xác (mô hình HYDRA-MM xem II.3.4).

II.3. MỘT SỐ MÔ HÌNH HỌC MÁY MÔ TẢ PHỨC

II.3.1. Mô hình FOIL

FOIL được đề xuất và phát triển bởi Quinlan (Quinlan, 1990). Giả mã của FOIL được giới thiệu trong bảng 2.1. Thực chất FOIL chưa phải là mô hình học máy mô tả phức song nhiều mô hình học máy mô tả phức được cải tiến từ FOIL. FOIL có 4 tham số là POS, NEG, Metric và Concept.

Bảng 2.1. Giả mã của FOIL

```
FOIL( POS, NEG, Metric, Concept):
    Let POS be the positive examples.
    Let NEG be the negative examples
```

```
Separate:                                   /begin a new rule/
Until POS is empty do:
  Let NewRule be the output of Build-rule (POS, NEG,Metric, Concept)
  Remove from POS all positive examples that satisfy NewRule.
End FOIL
-----
Build-rule (POS, NEG, Metric, Concept)
    Set NewRule to " Concept if TRUE" /this rule for all POS and NEG/
     Until NEG is empty do:
          Conquer: (build a rule body)
           Choose a literal L using Metric
           Conjoin L to body of NewRule.
           Remove from NEG examples that don't satisfy NewRule.
     Return NewRule
    End Build-rule.
```

FOIL học các tập dữ liệu chỉ bao gồm hai lớp, trong đó một lớp được gọi là "tích cực". FOIL học mô tả lớp đối với lớp "tích cực". Như vậy, FOIL học mô hình đơn bao gồm một mô tả lớp đơn. Thêm vào đó, FOIL sử dụng giả thiết thế giới-đóng đối với sự phân lớp (Lloyd, 1984).

Cho các ví dụ tích cực và tiêu cực về một nội dung nào đó, và một tập các khẳng định nền được xác định theo dạng dàn trải, FOIL sinh một cách quy nạp các định nghĩa khái niệm lôgic hoặc luật đối với khái niệm. FOIL có một hạn chế là luật quy nạp không được chứa bất cứ ký hiệu hằng hoặc ký hiệu biến nào (*ví dụ*, chúng ta không viết *color(X,red)* mà viết là *color (X,Y), red(Y)* song lại cho phép khẳng định âm). Theo cách hạn chế, FOIL cũng cho phép dùng khẳng định được học. Theo cách này, FOIL có thể học các khái niệm đệ quy. FOIL là mô hình học máy không tăng trong thuật toán "leo đồi" sử dụng metric dựa theo

lý thuyết thông tin xây dựng một luật bao trùm lên dữ liệu. FOIL sử dụng cách tiếp cận "tách rời và chế ngự" hơn là cách tiếp cận "chia nhỏ và chế ngự".

Pha "tách rời" của thuật toán bắt đầu từ luật mới trong khi pha "chế ngự" xây dựng một liên kết các literal làm thân của luật. Mỗi luật mô tả một tập con nào đó các ví dụ tích cực và không có ví dụ tiêu cực. Lưu ý rằng, FOIL có hai toán tử: bắt đầu một luật mới với thân luật rỗng và thêm một literal để kết thúc luật hiện tại. FOIL kết thúc việc bổ sung literal khi không còn ví dụ tiêu cực được bao phủ bởi luật, và bắt đầu luật mới đến khi tất cả mỗi ví dụ tích cực được bao phủ bởi một luật nào đó.

Các ví dụ tích cực được phủ bởi mệnh đề sẽ được tách ra khỏi tập dạy và quá trình tiếp tục để học các mệnh đề tiếp theo với các ví dụ còn lại, và kết thúc khi không có các ví dụ tích cực thêm nữa.

Để giải thích việc bổ sung literal trong thuật toán FOIL, chúng ta xem xét sơ bộ ví dụ FOIL học mối quan hệ *Ông(X,Y)* từ các quan hệ *Cha(X,Y)* và *Chamẹ(X,Y*), được xác định theo dạng dàn trải. Hơn nữa, giả sử rằng luật hiện tại (NewClauseBody trong bảng 2.1) là *Ông(X,Y*) ← *Chamẹ(X,Z)*. Sự mở rộng của luật này có thể đạt được bởi việc kết nối phần thân với một số literal *Cha(X,X), Cha(Y,Z), Cha(U,Y), Cha(Y,Z), Cha(Y,Y)* là tốt như nhau. Từ ví dụ này chúng ta có thể thấy rằng, để tạo một literal mở rộng một luật, không chỉ cần lựa chọn một ***tên-khẳng định*** mà còn cần một ***tập các biến riêng cho tên-khẳng định*** đó. Chúng ta gọi sự lựa chọn của các biến cho tên- khẳng định là *variablization (biến đổi)* của khẳng định. Nếu các biến được lựa chọn xuất hiện trong một literal không âm của luật thì được gọi là *cũ (old)*. Các trường hợp khác biến được gọi là *mới (new)*. Một đòi hỏi của FOIL đối với literal là literal cần chứa đựng ít nhất một biến cũ.

Nếu sự mở rộng luật được thiết lập bằng cách kết hợp một literal chỉ sử dụng các biến cũ thì tập hợp mới các ví dụ tích cực và tiêu cực sẽ là tập con của

các ví dụ cũng là tích cực và tiêu cực cũ bảo đảm khẳng định được bổ sung. Tình hình sẽ khác đi nếu sự mở rộng của luật bao gồm các biến mới.

Giả sử FOIL mở rộng một luật ***Ông***(X,Y) ← true bằng cách liên kết literal ***Cha***(X,Z), trong đó có đưa vào biến mới Z. Bây giờ các ví dụ tích cực bao gồm các giá trị <X, Y, Z> chẳng hạn ***Ông***(X,Y) là true và ***Cha***(X,Z) là true. Bộ <X, Y, Z> như vậy được gọi là bộ tích cực (dương). Cho trước cặp <X, Y> có thể không nhận hoặc nhận nhiều giá trị của Z mà ***Chamẹ***(X,Z) là true. Hoàn toàn tương tự, tập các bộ tiêu cực (âm) chứa các giá trị của <X,Y,Z> như là ***Ông***(X,Y) là false nhưng ***Chamẹ***(X,Z) là true. Để có hiệu quả, một ví dụ là một bộ sắp thứ tự các ràng buộc cho các biến của luật. Khi một biến mới được đưa vào, bộ đó mở rộng để bao hàm các giá trị của biến đó.

Với sự chuẩn bị như vậy, xem xét hoạt động của thuật toán nguồn trong bảng 2.1. Để cho đơn giản, coi các ví dụ tích cực nguồn như là bộ tích cực.

Ở mức độ tóm tắt thật gọn, FOIL khá đơn giản. Nó sử dụng thuật toán leo đồi để bổ sung các literal với thông tin thu được lớn nhất vào một luật. Với mỗi biến đổi của một khẳng định P, FOIL đo lượng thông tin đạt được. Để lựa chọn literal với thông tin đạt được cao nhất, nó cần biết bao nhiêu bộ tích cực và tiêu cực hiện tại được bảo đảm bởi các biến đổi của mỗi khẳng định được xác định theo cách dàn trải.

Phân tích FOIL

Nhìn chung, giá để thực hiện tìm kiếm leo đồi như FOIL tiến hành là sự kiện rẽ nhánh nhiều lần theo độ sâu ở đó một giải pháp được tìm ra. Thông thường, sự kiện rẽ nhánh không phải là hằng số thì ít nhất cũng bị ràng buộc. Trong FOIL, sự kiện rẽ nhánh phát triển rất nhanh theo số mũ trong đối của các khẳng định, đối và độ dài của luật đang được học.

Bắt đầu, thuật toán ước lượng giá của việc bổ sung một literal đơn vào một luật. Có hai độ đo được dùng để ước lượng giá này. Độ đo thứ nhất gọi là giá-lý thuyết (theory-cost), chỉ ra số các literal khác nhau có thể được chọn để mở rộng

thân của một luật cho trước. Độ đo thứ hai gọi là giá-ước lượng (value-cost), đo giá của việc tính toán thông tin đạt được của literal. Trong hai độ đo này, giá-ước lượng là một hàm của các ví dụ dạy còn giá-lý thuyết thì không phải.

II.3.2. Mô hình FOCL

FOCL (First Order Combined Learner) được Pazzani M. và Kibler D. đề xuất vào năm 1992 ([19]). FOCL là một hệ thống học máy mở rộng hệ thống FOIL của Quinlan bằng cách cho các giải thích tương thích dựa trên các thành phần được học. FOCL học câu Horn từ các ví dụ và tri thức nền. FOCL được thể hiện trong Common Lisp và chạy trên khá đa dạng máy tính. Giả mã của FOCL được cho trong bảng 2.2.

Bảng 2.2. Giả mã của FOCL

```
Let P be the predicate to be learned.
Let POS be the positive tuples.
Let NEG be the negative tuples
Let IR in the initial rule.
Let Body be empty.
Until POS is empty
        Call LearnClauseBody
        Remove from POS those tuples covered by Body
        Set Body to empty
Procedure LearnClauseBody:
        If a ClauseBody of IR has positive gain
                Select it,      /xem chú thích 1/
                Operationalize it (if necessary), /xem chú thích 3/
                Conjoin it with Body,
                Update POS and NEG,
                Call ExtendBody                 /xem chú thích 2/
        Else
```

Choose best literal,
Operationalize it (if necessary), /xem chú thích 3/
Conjoin result with Body,
Update POS and NEG,
Call LearnClauseBody.

Procedure ExtendBody:
While NEG is non-empty
Choose best literal /xem chú thích 3/
Operationalize it,
Conjoin it with Body,
Update POS and NEG,

Các chú thích:

1: nhận các lợi thế của các luật có trước tốt

2: cho phép hiệu chỉnh thân các luật cũ

3: cho phép sử dụng các khẳng định không thao tác

FOCL hoạt động tương tự như FOIL trong việc học một tập các luật. Tuy nhiên, nó học một tập hợp các luật cho mỗi lớp làm cho nó có thể đối phó với các vấn đề có nhiều hơn hai lớp. Thuật toán học luật được chạy cho mỗi lớp, xử lý các ví dụ cho lớp đó như là các ví dụ tích cực và các ví dụ của lớp khác như là những ví dụ tiêu cực. Điều này cho ta một tập hợp luật cho mỗi lớp.

Bản FOCL trên máy Macintosh cho một giao diện đồ hoạ các đồ thị không gian tìm kiếm được khảo sát bởi FOCL, và đó là một tool sư phạm hữu dụng để giải thích đối với học dựa theo sự giải thích và cảm hứng. Hơn nữa, trong FOCL cho phép dễ dàng khởi tạo và biên tập đồ thị các cơ sở tri thức, luật dẫn và các giải thích sinh, và do đó phiên bản của FOCL trên Macintosh có thể được sử dụng như một hỗ trợ hệ chuyên gia.

FOCL mở rộng FOIL theo nhiều cách. Mỗi sự mở rộng này chỉ tác động đến việc FOIL chọn các literal nào để kiểm tra trong khi mở rộng một câu (có thể rỗng) đang xây dựng. Những mở rộng này cho phép FOCL có ưu thế của lĩnh vực tri thức để xử lý bài toán. Mỗi lớp của sự mở rộng cho phép FOCL sử dụng các ràng buộc hạn chế không gian tìm kiếm. Loại mở rộng thứ hai cho phép FOCL sử dụng các khẳng định được xác định theo cách bổ sung (ví dụ, khẳng định được xác định bởi một luật thay cho một tập các ví dụ) theo cách tương tự đối với khẳng định được xác định dàn trải trong FOCL. Một tập của các khẳng định xác định theo cách bổ sung thì chứng minh cho lý thuyết miền của EBL (Mitchell, Keller & Kedar-Cabelli, 1986). Cuối cùng sự mở rộng cho phép FOCL chấp nhận là đầu vào một phần, luật có thể không đúng mà nó là một sự xấp xỉ ban đầu của khẳng định được học, nó giống như một định nghĩa khái niệm riêng lẻ được xây dựng bởi một hệ thống học quy nạp tăng. Nếu luật này được định nghĩa trong hạng thức của những khẳng định được xác định bổ sung, nó giống như khái niệm đích của EBL. Thật vậy, khi chúng ta thảo luận dựa trên sự giải thích các mở rộng của FOCL, chúng ta sẽ sử dụng các hạng thức "non-operational" và "intensionally defined" cùng một nghĩa. Tương tự các khẳng định được xác định dàn trải tương ứng với các sự kiện quan sát (hoặc các toán tử khẳng định) của EBL. Mục đích của FOCL giống như FOIL là tạo ra một luật (ví dụ một tập các câu) trong hạng thức của các khẳng định được xác định dàn trải bao phủ toàn bộ các ví dụ tích cực và không chứa ví dụ tiêu cực.

Sau đây sẽ mô tả các mở rộng này chi tiết hơn và đánh giá hiệu quả của mỗi sự mở rộng trên số literal được kiểm tra bởi FOCL hoặc độ chính xác của FOCL. Để minh hoạ những mở rộng này, sử dụng 2 miền như dưới đây. Miền thứ nhất - việc học khẳng định *Member,* minh hoạ một khái niệm đệ quy đơn như thế nào có thể được học. FOIL đã giới thiệu các ví dụ tích cực và tiêu cực của khẳng định *member* và khẳng định *component* và học định nghĩa đệ quy đúng cho *member* như trong bảng 2.3.

Bảng 2.3. Các luật cho hàm member

1. member(X,Y) ← component(X, Z, Y).

2. *member(X,Y) ← component(X, Z, Y).*

member(X,Y) ← component(A, B, Y).

member(X,Y) ← component(X, Y, Z).

Miền thứ hai phức tạp hơn nhiều và đã được giới thiệu bởi Muggeleton và cộng sự (1989) trong việc học các nước cờ. Miền này khẳng định rằng FOCL có thể điều khiển làm giảm kích thước các miền thực. Hàng trăm ví dụ được dùng để xây dựng một mô tả khái niệm khác nhau từ 4 đến 11 câu, dựa vào sự mở rộng các khẳng định được cung cấp. Khẳng định hoặc khái niệm được học là *illegal(*A,B,C,D,E,F). Đó là sự thật nếu bàn cờ bao gồm một vua trắng, xe trắng và vua đen ở trong một trạng thái *illegal* (trái luật). Một trạng thái là *illegal* nếu hoặc là vua bị chiếu hoặc là nhiều hơn một vị trí chiếm giữ cùng một không gian. A, B là vị trí vua trắng (hàng và cột), C, D là vị trí xe trắng và E, F là vị trí vua đen. Các hàng và cột được biểu diễn bởi các số từ 1 đến 8. Trong ví dụ này, các toán tử khẳng định sử dụng là: *giữa(X,Y,Z)* (giá trị của Y là giữa giá trị X và Z), *kề(X,Y)* (giá trị của X hoặc lớn hơn hoặc nhỏ hơn giá trị của Y), *bằng(X,Y)* (giá trị của X và Y như nhau).

Bảng 2.4: Đặc tả tổng kết FOCL

Input:

1. *Tên của khẳng định của đối số đã biết.*
2. *Một tập các bộ dương.*
3. *Một tập các bộ âm.*
4. *Một tập các khẳng định được xác định theo cách dàn trải.*
5. *Một tập các khẳng định được xác định theo cách bổ sung.*
6. *Một tập các ràng buộc (ví dụ typing) trên các khẳng định bổ sung và dàn trải.*

7. *Một luật (toán tử hoặc phủ định) ban đầu.*

Output:

Luật trong các hạng thức của khẳng định dàn trải mà không câu nào phủ một ví dụ tiêu cực và một số câu phủ mọi ví dụ tích cực.

Sau đây là giải thích mỗi thành phần của FOCL và chỉ ra chúng điều chỉnh lẫn nhau như thế nào trong bảng 2.4. Đây là thiết kế ở mức độ cao nhằm nhấn mạnh sự khác biệt với FOIL. FOCL mở rộng FOIL theo một số cách. Đầu tiên, có những ràng buộc trong quá trình quy nạp vì rằng không phải tất cả các sự biến đổi của một khẳng định cần được kiểm tra. Thứ hai, FOCL có thể tính toán thông tin đạt được của các khẳng định được xác định theo cách bổ sung (bổ sung vào các khẳng định được xác định theo cách dàn trải). Thứ ba, FOCL có thể dùng các khẳng định được xác định theo cách bổ sung bởi việc tìm một toán tử đặc biệt mà phủ nhiều ví dụ tích cực và một số it ví dụ tiêu cực. Thứ tư, FOCL có thể tính toán thông tin đạt được của một luật (toán tử hoặc phủ định) ban đầu cho khái niệm được học và quyết định sử dụng điều đó trong favor của quy nạp. Giá trị của luật ban đầu (ví dụ khái niệm đích) chỉ ra sự biến đổi của khẳng định phủ nhận tức là giống như được sử dụng. Bảng 2.4 biểu diễn những nét chung của thuật toán FOCL.

Metric thu thập thông tin đồng bộ cho FOCL khả năng giải quyết lý thuyết miền chưa đầy đủ và chưa đúng do đáp ứng cả hai dạng literal phân tích và literal quy nạp. Chỉ có một ít khác nhau giữa hai dạng này là việc tìm kiếm một trong các literal dạng phân tích chủ đạo. Quyết định việc sử dụng quy nạp hay phân tích để mở rộng một câu được căn cứ vào lợi ích trong sản xuất và độ chính xác giả thiết, và được đo bởi metric thu thập thông tin.

II.3.3. Mô hình HYDRA

HYDRA được bắt nguồn từ FOCL ([17]) và bổ sung cho FOCL khả năng học sử dụng tri thức nền được xác định trong các hạng thức của các luật. Giả mã của HYDRA được trình bày trong bảng 2.5. HYDRA dựa trên sự mở rộng của

FOIL (Quinlan, 1990) mà Ali và Pazzani (1993), Pazzani và cộng sự (1991) đã có nhiều cải tiến sửa đổi, bổ sung cho việc học một số mô hình. Trong thân của HYDRA cũng có hạt nhân là FOIL (xem bảng 2.5).

Bảng 2.5: Giả mã của HYDRA

```
HYDRA ( Metric, POS_1,. . ., POS_n):
        For i  in  classes  1  to  n  do
                POS = POS_i
                NEG= (POS_1  union . . . POS_n) - POS_i
                ConceptDescription_i = FOIL(POS, NEG, Metric)
                For  rule  R  in  ConceptDescription_i  do
                        Augment  R with  its  LS
```

HYDRA khác với FOCL ở ba điểm chính:

• HYDRA học một tập các luật cho mỗi lớp do đó mỗi tập hợp có thể so sánh để phân lớp các ví dụ test. Ali K. & Pazzani M. [5] đã chỉ ra rằng điều này cho phép HYDRA học máy với các bộ dữ liệu bị nhiễu được chính xác hơn.

• HYDRA gắn liền một độ đo có tính đầy đủ về mặt lôgic (ls- độ đo tin cậy của việc phân lớp) đối với mỗi luật.

• Metric được sử dụng bởi HYDRA (là metric ls-nội dung) để sắp xếp các literal tương xứng với phạm vi phủ ví dụ tích cực với độ chính xác dạy tạo ra các ưu điểm về bao phủ lớn hơn trường hợp thực hiện bởi metric thu thập thông tin (có trong mô hình FOCL). Ưu điểm này cũng không được khuyếch trương khi làm việc với các bộ dữ liệu có nhiễu quá lớn.

Biểu diễn tri thức trong HYDRA

Các luật học máy đối với HYDRA đi kèm với độ đo mức độ đầy đủ về logic (đo độ tin cậy trong phân loại):

$$ls_{ij} = \frac{p(rule_{ij}(T) = True / T \in class_i)}{p(rule_{ij}(T) = True / T \notin class_i} \qquad (2.9)$$

ở đây T là đại diện cho một ví dụ tuỳ ý. Mức độ đầy đủ về mặt lôgic là sự khái quát hoá của khái niệm đầy đủ mà phần thân của một luật có chứa phần đầu của luật đó. Muggleton và các tác giả ([15]) chỉ ra rằng phạm vi phủ các cí dụ cần dạy cho một biểu thị tin cậy về độ chính xác thật sự của luật hơn là các tham số đo độ chính xác từ dữ liệu cần dạy. Vì thế mô hình lựa chọn sử dụng Ls như là độ đo tin cậy bởi vì nó có ưu điểm là đo cả độ bao phủ và độ chính xác.

Để hiểu cách HYDRA phân lớp một ví dụ test như thế nào thì cần hiểu về khái niệm luật biểu diễn. Luật biểu diễn của một tập các luật R và một ví dụ test x là luật có độ tin cậy cao nhất được lựa chọn từ tập con của R mà thoả mãn x. Luật biểu diễn được chọn lọc từ mỗi lớp mà ở đó x thoả mãn ít nhất một luật. Ví dụ test được phân loại theo các lớp mà các luật biểu diễn nó có độ tin cậy cao nhất. Nếu có quá một luật biểu diễn trong một tập các luật thoả mãn thì vẫn chưa chắc chắn là ví dụ test thuộc lớp này. Các thử nghiệm của Ali K. và Pazzani M. [5] đã chỉ ra rằng phương pháp này được áp dụng tốt nhất khi việc mô tả định hướng tới một khối tương ứng với một tập xác định các khái niệm nền. Việc mô tả định hướng khái niệm (tương ứng với một số tập hợp xác định các khái niệm nền) thì được xác định như là sự mô tả nhất quán ngắn nhất cho khái niệm đó trong các hạng thức của tập xác định các khái niệm nền.

HYDRA cũng sử dụng chiến lược chia nhỏ và chế ngự như FOCL. Mặc dù vậy, HYDRA sử dụng metric ls-nội dung để sắp xếp các literal. Ls-nội dung được xác định như sau: Giả sử luật thứ j đang được học và nó phủ p ví dụ dạy tích cực và n ví dụ tiêu cực, p_j, n_j tương ứng kí hiệu số ví dụ tích cực và ví dụ tiêu cực không phủ bởi j-l luật được học đầu tiên. Sau đó ls-nội dung được xác định như là một sản phẩm của độ tin cậy của luật LS_j và phủ (p) luật :

$$\text{ls-nội dung } (p,n,p_j,n_j) = ls(p,n,p_j,n_j)*p$$

ở đây $ls(p,n,p_j,n_j)$ cho tỉ lệ các xác suất trong định nghĩa của Ls (phương trình 2.9) được tính toán từ dữ liệu. Sử dụng metric, HYDRA chọn bổ sung literal vào thân luật là literal đo làm cực đại hoá kết quả thu được đối với ls-nội dung. Khi

không còn literal tạo nên sự tăng trong ls-nội dung hoặc nếu như luật không phủ bất kì ví dụ tiêu cực nào, HYDRA tách các ví dụ phủ bởi luật từ tập dạy và học các luật kế tiếp đối với các ví dụ còn lại. HYDRA học các luật thích hợp qua việc sử dụng ls-nội dung hơn là việc sử dụng thông tin thu được ([2]) cho tập dữ liệu bị nhiễu bởi vì ls-nội dung coi trọng các luật học máy mà phủ nhiều ví dụ hơn và từ đó làm thích hợp các dữ liệu nhiễu với mức độ nhỏ hơn.

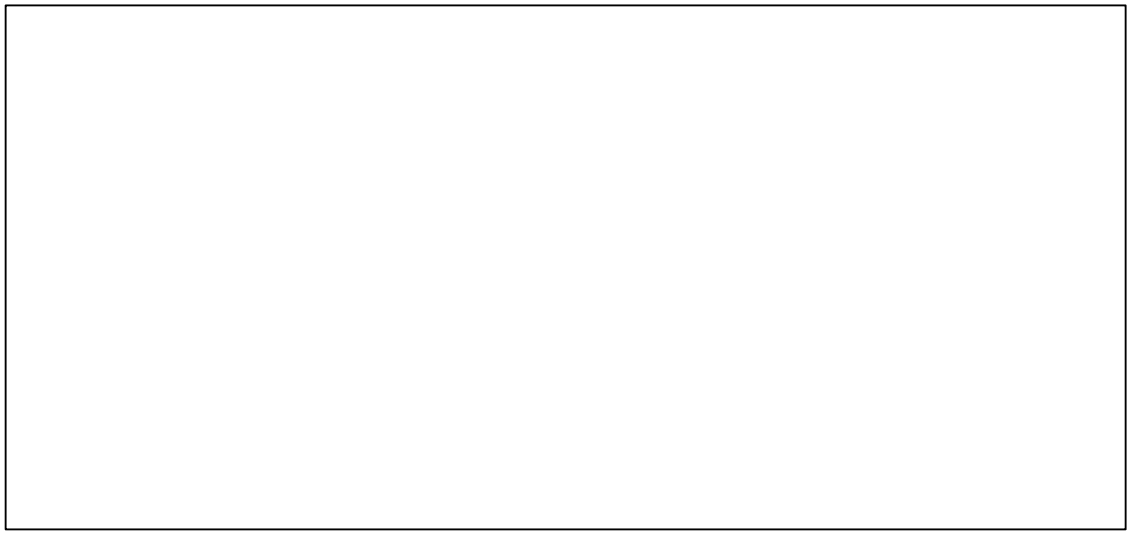
Sau khi tất cả các luật được học, HYDRA sẽ đưa ra một ước đoán về mức độ đầy đủ về mặt logic $ls_{i\,j}$ kết hợp với luật j của khái niệm i. Lưu ý rằng ls được sử dụng như là một độ đo trong khi học và nó cũng được sử dụng để phân lớp một cách tin cậy. Sau khi học, ls được đánh giá qua một tập gốc các ví dụ dạy, không chỉ từ các ví dụ không phủ bởi các luật học máy trước đây sẽ được sử dụng như phương trình (2.10) dưới đây. Mô hình sử dụng đánh giá Laplace cho xác suất để tính toán tỉ lệ các xác suất trong định nghĩa của Ls. Tiên đề Laplace về xác suất của một sự kiện xảy ra ngẫu nhiên f từ kết quả của T thí nghiệm là $\frac{f+1}{T+2}$. Do đó việc thay thế các xác suất trong định nghĩa của ls bởi các đánh giá Laplace của nó sinh ra:

$$ls_{i\,j}=ls(p,n,p_0,n_0) \approx \frac{(p+1)/(p_0+2)}{(n+1)/(n_0+2)} \qquad (2.10)$$

ở đây p_0 và n_0 tương ứng kí hiệu số các ví dụ dạy tích cực và đối ngẫu (của lớp i) trong toàn bộ tập dữ liệu và p và n ký hiệu được phủ bởi luật. Các luật với $Ls_{i\,j} \le l$ sẽ bị loại bỏ.

II.3.4. Mô hình HYDRA-MM

Để học với mô tả khái niệm phức đối với mỗi lớp, HYDRA được cải tiến thành HYDRA-MM nhằm cho phép học chính xác với ngữ cảnh có "tiếng ồn" đúng như trong “thế giới thực”. Hầu hết các chương trình học, bao gồm cả HYDRA, học chỉ một mô hình của dữ liệu. HYDRA-MM học với các mô hình phức của dữ liệu, mỗi mô hình là kết hợp của các mô tả khái niệm.



Hình 2.2. Mô hình học dữ liệu có trong hai lớp

Một mô tả khái niệm là một tập luật bao gồm tất cả các kết luận cho cùng một lớp. Một mô hình (model) là một tập các khái niệm, mỗi cái ở một lớp trong dữ liệu. Học máy mô tả khái niệm phức và kết hợp các tập hợp dữ liệu là một trong những phương pháp để nâng cao độ chính xác trong khái niệm "thế giới thực" mà nó không thể diễn tả độ chính xác bởi một tập luật đầy đủ và chính xác nhưng nó cần kết hợp dữ liệu từ nhiều nguồn. Các luật không hoàn toàn đầy đủ thì được mô phỏng trong HYDRA bởi gán thêm một độ đo logic đầy đủ (ls,[7]) cho mỗi luật. Mô tả phức của một khái niệm đưa ra một cách thức nhằm tạo ra một quyết định dựa trên một tổ hợp rõ ràng. HYDRA-MM cũng học mô tả phức đối với khái niệm đã cho.

Hình 2.2 cho một ví dụ, chỉ ra mô hình học từ dữ liệu từ hai miền. Chú ý rằng các luật là mệnh đề Horn bậc 1 và có thể đệ quy. Hình 2.2 chỉ rõ một cách rất điển hình là mô tả khái niệm đối với lớp đã cho là tương quan, và có thể kéo theo sự thay thế một literal chẳng hạn c(Z,Y) thành một literal khác gần giống với thành phần đầu tiên: h(Z,Y). Tất nhiên điều đó không loại trừ việc hợp lý hóa sự sai lệch giữa các mô tả khái niệm đối với lớp đã cho.

Trong cách thức sử dụng luật Bayes, mô hình sử dụng độ đo Ls của các luật để tính toán phần dư (odds) hậu nghiệm của một lớp. Cách thức này được

dùng khi các luật học máy sử dụng ls- nội dung. Sự phân lớp sử dụng mô tả khái niệm phức được học máy với metric ls-nội dung sử dụng dạng dư của luật Bayes ([7]), trong đó, metric có phần dư tiên nghiệm của lớp (O(lớp i)) được nhân lên bởi odds multipliers O(lớp i/$M_{i\,j}$)) của mô tả khái niệm cho lớp đó để sinh ra các phần dư tiện nghiệm cho lớp (O(lớp i / $M_{i)}$):

$$O(class_i / M_i) \alpha\ O(class_i) *\ {}_j\pi\ O(class_i / M_{i\,j}) \qquad (2.11)$$

Trong thực tế, qua odds multipliers của tập luật M_{ij}, người ta đã sử dụng độ đo Ls của luật biểu diễn cho ví dụ test hiện tại.

Cách thức thứ hai có nội dung kết hợp các chứng cứ sử dụng các xác suất hậu nghiệm. Phương pháp này được dùng khi các luật được học sử dụng metric đạt được Bayes.

Xem xét sơ bộ một số nội dung trong học máy theo HYDRA-MM. Áp dụng metric đạt được Bayes, có thể sử dụng ước lượng Laplace về độ chính xác dạy của một luật. Với một luật phủ p ví dụ dạy tích cực là n ví dụ không tích cực thì ước lượng Laplace được chú ý bây giờ là $\frac{p+1}{p+n+2}$. Biểu thức ước lượng này là có lợi đối với toàn bộ việc cực đại hoá hơn ước lượng có thể dùng là $\frac{p}{p+n}$, trong đó hệ thức này không tự động gán độ chính xác **l** cho luật phủ đã nói khi có nhiều hơn một ví dụ dạy tích cực và không ví dụ dạy tiêu cực. Điều đó là khác biệt với trường hợp một luật có độ chính xác **l** trong ví dụ test. Với hệ thức tính toán trên đây, việc kết hợp các chứng cứ từ các mô tả phức của một lớp được học sử dụng metric Bayes sau đó được tiến hành như giải thích dưới đây.

Nếu $a_{i,j}$ kí hiệu độ chính xác Laplace của luật biểu diễn của khái niệm thứ j cho lớp i và $p_{i,j}$ biểu thị xác suất hậu nghiệm của việc mô tả khái niệm thứ j của lớp i, toàn bộ xác suất hậu nghiệm của lớp i sẽ là:

$$\text{Hậu nghiệm (i)}=a_{i,1} * p_{i,1} +... a_{i,n} * p_{i,n} \qquad (2.12)$$

HYDRA-MM sau đó sẽ gán với ví dụ test cho lớp có xác suất hậu nghiệm cao nhất.

Việc sử dụng mô hình phần dư của luật Bayes do mô hình này nhất quán với mức độ đầy đủ và logic (Ls, [7]). Mức độ đầy đủ về mặt logic cho lớp C cũng được gọi là phần dư phức hợp bởi vì nó được nhân lên bởi tỷ số odds của C để tạo ra phần dư hậu nghiệm của C. Ví dụ được phân lớp theo giá trị phần dư hậu nghiệm lớn nhất.

Các kết quả thử nghiệm

Ali K. và Pazzani M. đã kiểm nghiệm giả định liệu các mô tả khái niệm phức có cần thiết nhất cho các không gian giả thiết mà ở đó có nhiều luật tốt như nhau trong việc học máy theo metric thu được. Các tác giả đã chỉ ra được ưu điểm của việc tính xấp xỉ theo bề rộng đối với xác suất và so sánh các tập hợp luật phức với các phương pháp các luật phức. Trong nghiên cứu đó, cũng tập trung đến việc so sánh của Bayes và các metric ls-nội dung. Nếu sử dụng các tiên nghiệm không đồng bộ là thuận lợi nhờ việc sử dụng tiên nghiệm đồng bộ Bayes biểu thị việc học máy với metric Bayes thu được và độ tin cậy về độ chính xác của các luật và xác suất của mô hình (phương trình 2.12). Trong phương trình đó, Accu là đơn giản hoá của Bayes tại đó, các mô tả khái niệm sử dụng độ chính xác của luật mà tương đương đối với tất cả các mô hình có xác suất hậu nghiệm bằng nhau.

Bên cạnh việc thử nghiệm HYDRA-MM theo dạng quan hệ, Ali K. & Pazzani M. [5] cũng tiến hành thử nghiệm theo các miền xác định là: các miền giá trị thuộc tính “ung thư vú và u lành", dự đoán trước người thúc đẩy DNA, vấn đề chơi cờ RRKP. Trong miền xác định phần thúc đẩy DNA, cần thiết thêm mối quan hệ Y(x) mà là đúng nếu nuclcotide x là t hoặc là c và vị ngữ r(x) là đúng nếu x là a hoặc g ở đó meclcotide (nguyên tử) là một trong số {a,g,t,c}

Phương pháp trắc nghiệm

Với tập dữ liệu lưới phần tử hữu hạn, các ví dụ được bắt đầu từ 5 đối tượng Dzei và Bratko ([8]) sử dụng "loại một đối tượng ra ngoài" kiểm tra chiến lược mà ở phép thử thứ i, các ví dụ tích cực từ đối tượng i được sử dụng cho việc trắc nghiệm và các ví dụ tích cực và tiêu cực của các đối tượng khác được sử dụng cho việc dạy. Thực tế mà trắc nghiệm không bao giờ xảy ra với các ví dụ tiêu cực rất quan trọng trong việc giải thích một số kết quả về miền xác định này. Ali K. & Pazzani M. [5] đã sử dụng nhiều thực nghiệm để đánh giá việc giảm lỗi trong mô hình HYDRA-MM.

CHƯƠNG 3. RÚT GỌN LỖI TRONG HỌC MÁY MÔ TẢ PHỨC

III.1. SƠ BỘ VỀ RÚT GỌN LỖI TRONG HỌC MÁY MÔ TẢ PHỨC

III.1. 1. Một số khái niệm

Để đánh giá mức độ tốt của một mô hình học máy có giám sát, người ta thường đưa ra bộ các ví dụ kiểm tra (ví dụ test). Ta kí hiệu các ví dụ test là $test_1$, $test_2$, ..., $test_m$ trong đó m là số lượng ví dụ cần kiểm tra mô hình.

Định nghĩa 3.1

Mô hình phân lớp được gọi là lỗi đối với ví dụ $test_i$ đã biết thuộc khái niệm x nếu như mô hình phân $test_i$ thuộc vào khái niệm y mà $x \neq y$.

Số lượng các ví dụ kiểm tra bị lỗi trong mô hình phân lớp được gọi là ***lỗi tuyệt đối*** đối với bộ kiểm tra đã cho.

Trong các mô hình học máy mô tả phức, để đánh giá lỗi còn cần xem xét lỗi tương đối khi đối chứng với mô hình học máy đơn.

Định nghĩa 3.2 (Lỗi tương đối)

Tỷ số giữa lỗi tuyệt đối của một mô hình học máy đối với một kiểm tra với số lượng các ví dụ kiểm tra trong bộ kiểm tra đó được gọi là lỗi tương đối của mô hình đối với bộ kiểm tra.

Trong cách tiếp cận mô hình phức, việc học máy được xem xét theo một số mô hình đối với một tập cần dạy. Mỗi mô hình chứa một mô tả cho mỗi lớp trong đó mỗi mô tả là một tập hợp các luật cho lớp đó (mỗi mô tả lớp là một tập các luật Horn-cấp 1 cho lớp đó). Tập các mô hình học được xem xét được gọi là một toàn cảnh (ensemble) theo cách gọi của Hansen và Salamon vào năm 1990.

Định nghĩa 3.3. (Lỗi toàn cảnh)

Lỗi được xem xét trên tất cả các mô hình học trong một toàn cảnh được gọi là lỗi toàn cảnh.

Định nghĩa 3.4 (Tỷ lệ lỗi)

Hai độ đo tường minh so sánh lỗi toàn cảnh (E_e) với lỗi mô hình đơn (E_s) có độ khác nhau là (E_s-E_e) và tỉ lệ lỗi ($E_r = E_e/E_s$).

Thông thường tỉ lệ lỗi được sử dụng vì nó phản ánh thực tế là sẽ rất khó khăn để nắm bắt rút gọn lỗi theo cách tiếp cận mô hình đơn khi mà lỗi này xấp xỉ 0. Tỉ lệ lỗi <1 chỉ ra rằng cách tiếp cận mô hình phức cho lỗi thấp hơn phương pháp mô hình đơn. Tỷ lệ lỗi càng bé thì sự rút gọn lỗi càng tăng.

Định nghĩa 3.5. (Lỗi tương quan)

Hai mô hình được gọi là tạo ra "lỗi tương quan" nếu như cả hai đều phân lớp một ví dụ của lớp i lại thuộc về lớp j, $j \neq i$.

Liên quan đến lỗi tương quan thường sử dụng một metric, thường được ký hiệu là ϕ_e, với ý nghĩa "chia nhỏ đều lỗi (tương quan)" dùng để đo tỉ lệ các ví dụ kiểm tra với các thành viên của toàn cảnh, tạo nên cùng kiểu lỗi không phân lớp.

Ali K. & Pazzani M. [5] đã kiểm nghiệm một số giả thiết về sự rút gọn lỗi hầu hết trong các miền mà lỗi được tạo ra bởi mô hình trong toàn cảnh là được sinh ra từ cách thức không tương quan.

III.1.2. Sơ bộ về rút gọn lỗi trong học máy mô tả phức

Breiman (1994) đã cung cấp đặc trưng của việc học máy theo các thuật toán tuân theo cách tiếp cận các mô hình phức. Ông đi tiên phong trong việc đề xuất khái niệm về thuật toán không ổn định- thuật toán mà chỉ với nhiễu nhỏ của tập dạy sẽ dẫn đến sự khác nhau đáng kể trong các phân lớp được tiên đoán với tập hợp ví dụ độc lập. Breiman chỉ ra rằng thuật toán cây quyết định và thuật toán mạng neuron là không ổn định trong khi thuật toán người láng giềng gần nhất về cơ bản là ổn định (theo nghĩa nhiễu nhỏ sẽ gây sai lệch nhỏ). Ali K. & Pazzani M. [5] đưa ra các ***đặc trưng theo miền giá trị*** mà cách tiếp cận các mô hình phức là có lợi (các đặc trưng của miền giá trị như là nhiều thuộc tính không thích hợp, các mức độ nhiễu thấp) trong khi Breiman đưa ra các đặc trưng ***theo thuật toán học máy***.

Thuật toán Boosting của Schapire vào năm 1990 định hướng theo các mô hình học máy có tạo ra các lỗi độc lập về mặt thống kê. Thuật toán Boosting học theo một "dòng" các ví dụ. Các mô hình về sau được xây dựng trên các tập dạy làm tăng thêm số lượng các ví dụ không được phân lớp bởi các mô hình trước đây, chú trọng vào các ví dụ khó. Tuy nhiên, phương pháp Schapire được sử dụng để nghiên cứu các mô hình chỉ với kích cỡ vừa phải của tập dạy vì sự đòi hỏi tăng quá nhanh số lượng ví dụ dạy theo độ chính xác của mô hình.

Freund & Schapire vào năm 1995 đã cải tiến phương pháp Boosting bằng việc chú trọng tới các tập hợp dữ liệu nhỏ mà chưa được chứng minh bằng thực nghiệm và chú trọng tới các ví dụ nhiễu trong các tập dạy khó. Kovacic (năm 1994) đã chỉ ra rằng, học các mô hình phức bằng cách chạy m FOIL (Dzeroski, 1992) cho tỉ lệ lỗi thấp hơn một vài lần đáng kể so với chạy m FOIL trong KRK (King-Rook-King) và các tập hợp dữ liệu có ít phân tử.

Kwok và Carter (năm 1990) khi tiến hành đa dạng hoá đối với các mô hình phức đã chỉ ra rằng khi cho phép nhánh của cây quyết định thay đổi từ miền này sang miền khác sẽ tạo ra sự đa dạng cao hơn và các tập hợp thích hợp hơn là sự biến đối chỉ theo sự suy giảm cây. Trong công trình của mình, Ali K. & Pazzani M. [5] đã chỉ ra rằng, trong một số trường hợp, một khi việc đa dạng hoá tương xứng với độ chính xác -trong trường hợp như vậy, nhiều mô hình chính xác và khác nhau về mặt cú pháp có thể không tồn tại. Trước đó, Buntine (1990) cũng giới thiệu các kết quả trong đó các cây lựa chọn thì có thể thu được các tỉ lệ lỗi tốt hơn là tập hợp của các cây thu được bằng cách khác nhau của việc chọn lọc cây đơn ban đầu (gốc). Ông giả định điều này vì những chọn lọc khác nhau như thế không làm cho các cây có sự khác nhau như những cây thu được bởi sự biểu diễn cây lựa chọn.

Theo các nghiên cứu kinh nghiệm sử dụng mô hình phức (chẳng hạn, Buntine, 1990; Kononenko và Kovacic, 1992), công việc chủ yếu chỉ tập trung vào việc chứng minh rút gọn lỗi thông qua việc sử dụng các mô hình phức và

nghiên cứu về các phương pháp mới trong việc đưa ra các mô hình và tổ hợp các phân lớp của chúng. Từ các công trình nghiên cứu như thế, Ali K. & Pazzani M. [5] đã tổng quát việc rút gọn lỗi trong học máy mô tả phức có thể được đặc trưng theo ba chiều hướng:

- Loại mô hình đang được học (cây, luật...),
- Phương pháp trong việc tạo ra các mô hình phức,
- Phương pháp tổ hợp sự phân lớp các mô hình để tạo ra phân lớp toàn bộ.

Nghiên cứu của Kwok và Carter (1990) được coi là đã góp phần làm nền tảng về nội dung cho nghiên cứu của Ali K. & Pazzani M. [5] về tác động của việc đa dạng hoá cú pháp trong tỉ lệ lỗi. kết quả cho thấy rằng, học máy theo tập hợp các cây quyết định khác nhau về cú pháp sẽ thu được độ chính xác cao hơn tập hợp với cây có ít sự khác nhau.

Trước công trình của Ali K. & Pazzani M., các nghiên cứu về mặt lý thuyết trong việc học với mô hình phức bao gồm sự hình thành công thức Buntine của lý thuyết học Bayes tổng quát, thuật toán Schapire Boosting (1990) và các kết quả từ Hansen cùng Salamon (1990) và Drobnic cùng Gams (1992, 1993). Các nghiên cứu của Schapire tiến hành trên nguyên tắc (được chứng minh trong Hansen và Salamon, 1990) mà các mô hình tạo ra lỗi độc lập tuyệt đối sẽ tạo ra tập lỗi thấp hơn. Thuật toán Boosting chỉ mới học với thuật toán mà việc hợp nhất với mục đích cực tiểu hoá các lỗi tương quan trong quá trình học. Tuy nhiên, số lượng các ví dụ được đòi hỏi bởi thuật toán đó tăng lên như một hàm theo độ chính xác của các mô hình được học. Phương pháp Schapire có thể không được sử dụng để nghiên cứu nhiều mô hình về các kích cỡ tập dạy vừa phải.

Các kết quả mang tính lý thuyết khác về tác động của việc sử dụng mô hình phức bắt nguồn từ Hansen & Salamon (1990) đã chứng minh rằng, nếu tất cả các mô hình có cùng xác suất tạo lỗi và xác suất này dưới 0,5 và nếu như tất cả các mô hình tạo lỗi một cách độc lập thì sau đó toàn bộ các lỗi toàn cảnh

giảm đơn điệu như là một hàm của số mô hình. Phân tích mang tính lý thuyết về sử dụng các mô hình hồi quy phức cũng được thực hiện bởi Breiman. Tuy nhiên, nghiên cứu này không đề cập về định lượng việc rút gọn lỗi và nghiên cứu của Hansen & Salamon không nói gì đến các lỗi không độc lập tuyệt đối.

Ngoại trừ nghiên cứu của Buntine (1990), hầu hết các nghiên cứu mang tính thực nghiệm đã được tiến hành với một số lượng nhỏ các miền (2 miền: Knok & Carter (1990); 3 miền: Kononenko & Kovacic (1992); 3 miền: Smyth và cộng sự (1990)). Chỉ một số lượng nhỏ các miền được sử dụng đã làm giảm chính xác các điều kiện đặc trưng cần khảo sát theo rút gọn lỗi. Hơn nữa, mặc dù Buntine sử dụng nhiều tập hợp dữ liệu, nhưng ông không làm rõ được sự khác nhau về đặc trưng của các miền trong việc rút gọn lỗi. Bằng việc sử dụng 29 tập hợp dữ liệu của 21 miền, Ali K. & Pazzani M. [5] nghiên cứu xem xét nhằm phát hiện các đặc trưng của miền sẽ là yếu tố cần thiết trong việc rút gọn lỗi.

Xuyên suốt, học máy mô hình phức dữ liệu được đưa ra với mục đích cải tiến để loại trừ tỉ lệ lỗi phân lớp khi đối sánh với tỉ lệ lỗi gặp phải qua nhiều mô hình học máy dữ liệu đơn (*cây quyết định*: Kwok & Carter, 1990; Buntine 1990, Kong & Dietterich, 1995; *luật*: Gams, 1989; Smyth & Goodman 1992; Konen Ko & Kovacic, 1992; Brazdil & Torgo, 1990; *mạng nơron*: Hansen & Salomon, 1990 Baxt, 1992; *lưới Bayes*: Madigan & York, 1993; *hồi quy*: Perrone, 1993 Breiman). Tuy có nhiều nghiên cứu học mô hình phức đã được tiến hành song số lượng các miền dữ liệu được sử dụng lại rất ít. Các công trình như vậy, đã bước đầu cố gắng để biểu diễn việc rút gọn lỗi theo sự biến thiên từ miền này sang miền khác. Thông qua ba tập hợp dữ liệu được sử dụng, nghiên cứu của Ali K. & Pazzani M. [5] đi theo cách tiếp cận này đã cung cấp sự rút gọn lỗi lớn nhất (Tic-tac-toe, DNA, Wine). Với các tập hợp dữ liệu này, cách tiếp cận mô hình phức cho phép rút gọn lỗi phân lớp trên một tập test ví dụ tới 7 lần. Ali K. & Pazzani M. [5] thông qua việc định nghĩa chính xác "lỗi tương quan" để cung cấp quan niệm về sự thay đổi trong việc rút gọn lỗi. Các tác giả cũng trình bày quan điểm

"tăng liên kết" để hiểu được tại sao cách tiếp cận mô hình phức hiệu quả và tại sao nó lại đặc biệt hiệu quả đối với những miền có nhiều thuộc tính không thích hợp.

Nhiều công trình học mô hình phức được tiến hành theo lý thuyết học Bayes (ví dụ, Bernardo & Smith, 1994) đã tiến hành bức chế sự cực đại hoá độ chính xác của dự báo, thay vì tiến hành phân lớp dựa trên một mô hình học đơn cần sử dụng tất cả các giả thiết trong không gian giả thiết. Độ tin cậy của mỗi giả thiết phải được đánh giá bằng xác suất hậu nghiệm của chính giả thiết đó trong dữ liệu dạy. Bởi vì lý thuyết này đòi hỏi độ tin cậy của tất cả các giả thiết hay các mô hình trong không gian giả thiết, nên sự dễ dàng thực hiện vận dụng của lý thuyết này trở thành xấp xỉ. Điều này đặt ra câu hỏi đối với các nghiên cứu về rút gọn lỗi: *Phương pháp sinh-mô hình/tổ hợp-chứng cứ nào* sẽ cho tỉ lệ lỗi thấp nhất trong thực tế? Hoặc là, *làm cách nào để có thể đặc trưng hoá* các miền mà theo đó có được một phương pháp riêng làm việc tốt nhất và *tại sao phương pháp* riêng đó lại làm việc tốt nhất trong các miền như vậy?

Ali K. và Pazzani M. trong [5] trình bày kết quả giải quyết các câu hỏi nói trên mà đặc biệt là câu hỏi tại sao lại hợp lý khi mô hình học với các lỗi không tương quan tại một miền lại nhiều hơn so với các miền khác. Các tác giả đã nghiên cứu ảnh hưởng của sự thay đổi hai đặc trưng miền (mức độ nhiễu và số lượng các thuộc tính không thích hợp) tới tỉ lệ lỗi. Hơn nữa, các ông đã khảo sát ảnh hưởng của việc đa dạng hóa cú pháp tới lỗi toàn cảnh. Đây là sự tiếp nối nghiên cứu của Knok và Carter (1990), khẳng định rằng việc học theo cây quyết định với việc đa dạng hóa cú pháp sẽ đưa đến lỗi toàn cảnh thấp hơn.

Sau khi kiểm tra các vấn đề chính về học với mô hình phức, Ali K. & Pazzani M. [5] đã giới thiệu những kết quả nghiên cứu ở đây đối với thuật toán học HYDRA (Ali và Pazzani, 1992,1993,1994), trong đó đã đề xuất nhiều cải tiến mô hình học phức.

Các công trình nghiên cứu về rút gọn lỗi được tiến hành nhằm trả lời các câu hỏi sau:

1. Phương pháp mô hình phức có ảnh hưởng thế nào tới lỗi phân lớp khi so sánh với lỗi nảy sinh bởi mô hình đơn với cùng tập dữ liệu cần dạy?
2. Mối liên hệ nào giữa lượng rút gọn lỗi quan sát được (E_r) và xu hướng tạo ra các lỗi tương quan của các mô hình học?
3. Số lượng rút gọn lỗi được thấy tại một miền có thể được tiên đoán từ số lượng các mối ràng buộc thu thập kiến thức qua học với thuật toán trong miền đó hay không?
4. Việc tăng số lượng lớp có ảnh hưởng nhiễu như thế nào tới rút gọn lỗi?
5. Việc tăng số lượng các thuộc tính không thích hợp có ảnh hưởng như thế nào tới số lượng rút gọn lỗi?
6. Việc làm tăng tính đa dạng của mô hình có cần thiết hay không để rút gọn lỗi?

III.2. MỘT SỐ NỘI DUNG VỀ RÚT GỌN LỖI TRONG HỌC MÁY MÔ TẢ PHỨC

III.2.1 Sử dụng tập luật phức cho lỗi thấp hơn

Bằng các kết quả thực nghiệm, Ali K. và Pazzani M. [5] đã khẳng định rằng, sử dụng tập luật phức sẽ cho lỗi thấp hơn so với các mô hình chỉ sử dụng các luật đơn trong hầu hết các tập dữ liệu và không làm tăng lỗi đối với các tập dữ liệu còn lại.

Trong thử nghiệm, các tác giả đã sử dụng phương pháp chia nhỏ và ngẫu nhiên để học 11 mô hình (việc lựa chọn số lượng lẻ để tránh các liên kết xảy ra nhằm phân biệt với phương pháp kết hợp đánh giá đồng bộ với miền hai lớp). Mặt khác số lượng 11 mô hình là vừa đủ vùu rằng các thử nghiệm sơ bộ cho thấy với hầu hết các tập hợp số liệu, sử dụng nhiều hơn 11 mô hình sẽ cho kết quả không tăng ích lợi bao nhiêu trong khi đó chi phí thời gian tính toán tăng đáng

kết. Hiện tại chưa có phương pháp để chỉ ra số lượng tối ưu các mô hình để học cho một tập số liệu xác định.

Với phương pháp bất kỳ được sử dụng, các literal cho kết quả tối thiểu là 0,8 (β=0,8) được coi như là literal tốt nhất thì được dùng tiếp cho việc nghiên cứu tác dụng của việc thay đổi kích cỡ số lượng lớn.

Thực nghiệm mà Ali K. và Pazzani M. [5] đã triển khai cho thấy các tập hợp dữ liệu được nhóm lại thành nhóm như sau:

- Nhóm thứ nhất chứa các dữ liệu dạy nhiễu-tự do của các khái niệm nhân tạo (với khái niệm chúng ta biết các mô tả lớp là đúng),

- Nhóm 2 chứa đựng số liệu nhiễu của các khái niệm nhân tạo,

- Nhóm thứ 3 chứa đựng các tập hợp dữ liệu của các miền phân tử sinh học,

- Nhóm cuối cùng có thể chứa dữ liệu nhiễu của các chẩn đoán y học và các miền "thế giới thực" khác. Các miền ở nhóm cuối cùng được phân lớp để các miền này với các độ chính xác mô hình đơn lẻ cao nhất xuất hiện đầu tiên.

Thuật toán tìm kiếm ngẫu nhiên sử dụng tổ hợp tin cậy thì có thể giảm lỗi một cách tương đối mĩ mãn (95% tin cậy) hoặc duy trì lỗi trên tất cả các miền trừ miền ung thư- vú. Với tập hợp số liệu về ung thư vú, một số phương pháp nghiên cứu có thể cho ta độ chính xác cao hơn đáng kể so với kết quả thu được bằng cách phỏng đoán lớp thường xuyên nhất với giả sử là nó thiếu các thuộc tính thích hợp cho việc phân biệt các lớp. Với gần 1/ 2 các tập hợp dữ liệu, lỗi được giảm xuống bởi các cận đặc trưng về mặt thống kê khi sử dụng các mô hình được học bằng tìm kiếm ngẫu nhiên và kết hợp với phương pháp tổ hợp tin cậy. Các phương pháp tổ hợp bằng chứng khác và các phương pháp học dẫn đến việc giảm lỗi một cách đáng kể mang tính thống kê với các tập dữ liệu. Không có sự thay đổi lớn trong vấn đề lỗi cho hầu hết các tập dữ liệu khác- chỉ trong rất ít trường hợp tiến hành cách tiếp cận các mô hình phức dẫn đến sự tăng lỗi đáng kể.

Mặc dù các tỉ lệ lỗi cho các tập hợp dữ liệu nhiễu thể hiện sự giảm lỗi đáng kể về mặt thống kê, các tỉ lệ không gây ấn tượng như là các tỉ lệ cho các miền nhiễu-tự do, có chứa các thuộc tính không thích hợp. Đối lại, một số tập cung cấp một số dấu hiệu nội tại: các biến với các thuộc tính không thích hợp không có nhiễu và các biến thể nhiễu thì không có các thuộc tính không thích hợp. Từ đó đối với một số tập dữ liệu, tỉ lệ lỗi đạt được qua việc sử dụng các mô tả phức trở nên kém ích lợi khi số lượng nhiễu tăng.

Ali K. và Pazzani M. [5] đã khẳng định rằng việc sử dụng các mô tả phức cho sự giảm đáng kể trong phân lớp lỗi cho khoảng một nửa các tập hợp dữ liệu đã được kiểm tra. Với hầu hết các tập hợp dữ liệu khác, lỗi thay đổi không đáng kể. Do đó, nói chung cách tiếp cận các mô tả phức giúp đỡ đáng kể hoặc không sinh lỗi. Điều này đúng cho cả hai phương pháp đưa ra mô tả và tất cả các phương pháp tổ hợp bằng chứng được thực hiện ở đây.

III.2.2 Mối quan hệ giữa giảm lỗi và các lỗi tương quan

Hansen & Salmon vào năm 1990 đã đưa ra giả định rằng bằng cách sử dụng một tập các mô hình học máy là có ích khi các mô hình thành phần của nó tạo ra các lỗi độc lập với mọi mô hình khác trong tập hợp. Các ông đã chứng minh được rằng khi tất cả các mô hình có lỗi như nhau và lỗi đó thấp hơn 0,5 và chúng tạo ra các lỗi độc lập toàn bộ mà lỗi tập hợp kỳ vọng phải giảm một cách đơn điệu với số lượng các mô hình.

Trong công trình của mình, Ali K. & Pazzani M. [5] đã kiểm nghiệm về mối liên hệ giữa số lượng của giảm lỗi và xu hướng tạo nên lỗi tương quan của của các mô hình được học. Ali K. & Pazzani M. nghiên cứu trường hợp tổng quát hơn, thứ nhất là không cần giả thiết các lỗi tạo ra hoàn toàn độc lập, và thứ hai là giải thích số lượng giảm lỗi liên quan đến vấn đề về các lỗi tương quan. Trước hết xem xét khái niệm về lỗi tương quan trong một toàn cảnh.

Bằng thực nghiệm, Ali K. & Pazzani M. [5] đã kiểm nghiệm về mối liên hệ giữa số lượng của giảm lỗi và xu hướng tạo nên lỗi tương quan của của các mô

hình được học. Theo thực nghiệm cho thấy độ tin cậy của các dữ liệu đã cho là nhỏ hơn 0,01 cho tám tổ hợp của phương pháp học và phương pháp tổ hợp bằng chứng. Do đó, có thể kết luận rằng có sự tương quan tuyến tính đặc trưng giữa tỉ lệ lỗi và xu hướng tạo ra các lỗi tương quan cho tất cả các phương pháp học và các phương pháp tổ hợp bằng chứng trong nghiên cứu này. Các tác giả cũng kết luận rằng dùng phương pháp chia nhỏ cân bằng là có lợi hơn so với phương pháp tổ hợp chứng cứ để giảm lỗi.

III.2.3. Thu thập các mối quan hệ và rút gọn lỗi

Trong nhiều công trình nghiên cứu đã nảy sinh vấn đề về mối qliên hệ giữa số lượng rút gọn lỗi có thể được tiên đoán và số lượng quan hệ thu thập được từ các tri thức đối với tập hợp dữ liệu.

Xuất phát điểm cho giả thiết này là quan sát mỗi lần phương pháp tạo ra việc lựa chọn ngẫu nhiên khi sử dụng cùng một dữ liệu dạy. Tuy nhiên, có thể đưa ra các mô tả khác nhau bởi vì nó lấy một literal bất kỳ mà thông tin thu được của literal thuộc loại cao nhất. Nếu có nhiều literal như vậy thì xác suất cho phép biến thiên cú pháp từ mô tả này tới mô tả khác lớn hơn và như vậy tạo ra sự đa dạng hóa cú pháp. Càng đa dạng hoá cú pháp nhiều hơn thì có thể cho ta sự tương quan thấp hơn của các lỗi và sau đó cho tỉ lệ lỗi thấp hơn. Tổng số các quan hệ đã được học, hiểu biết trong quá trình nghiên cứu sau đó được chia ra theo số lượng các literal trong mô hình, để tạo ra "số lượng trung bình của các liên kết bổ sung" cho tập hợp dữ liệu. Số lượng lớn liên kết như vậy là trở ngại cho người học khi xác định " leo đồi" nhưng tạo ra cơ hội cho người học biết về mô hình phức. Các tác giả đã chỉ ra rằng, một số các rút gọn lỗi lớn nhất tăng thêm cho các tập hợp dữ liệu mà các liên kết như vậy là thường xuyên. Tuy nhiên, cũng chỉ ra rằng giá trị trung bình cao cho các liên kết tăng thêm không là điều kiện cần thiết cho việc giảm đáng kể lỗi. Ví dụ, các mô hình phức có thể tăng thêm các tỷ lệ lỗi thấp với bộ dữ liệu này và các biến thể nhiễu tự do song

với một số bộ dữ liệu khác không có nhiều liên kết thu được với những tập hợp dữ liệu này.

Tóm lại, nếu số lượng các liên kết tăng thêm được hiểu trung bình cho một tập hợp dữ liệu là lớn (từ hai trở lên) thì sau đó tập hợp dữ liệu đó sẽ có tác dụng lớn qua việc sử dụng các mô hình phức (có nghĩa là lỗi của nó được giảm xuống ít nhất 40%). Thực nghiệm cho thấy không có ngoại lệ nào đối với xu hướng này. Tuy nhiên, nếu số lượng của các liên kết tăng thêm là nhỏ thì số lượng của rút gọn lỗi không thể tiên đoán. Những kết quả tăng thêm liên kết này không chỉ đúng cho HYDRA mà cũng đúng cho nhiều mô hình học khác có liên quan đến thuật toán sử dụng cây quyết định chính tắc trong thuật toán học.

III.2.4. Tác động của nhiễu

Các kết quả thực nghiệm chỉ ra rằng, đa số (80%) các tập hợp dữ liệu mà tỉ lệ lỗi không đáng kể (trên 0,8) thì được ghi lại là các tập hợp dữ liệu với số lượng đáng kể của nhiễu. Hơn nữa, các thử nghiệm với một số miền dữ liệu cung cấp các chứng cứ sơ bộ rằng việc bổ sung thêm các thuộc tính nhiễu có thể làm tăng các tỉ lệ lỗi. Ali K. & Pazzani M. [5] đã chứng tỏ bằng thực nghiệm rằng việc tăng số lượng lớp nhiễu làm hạn chế việc giảm lỗi.

Trong các mô hình thực nghiệm đã sử dụng lựa chọn việc học có tác động của lớp nhiễu thay vì nhiễu theo thuộc tính nhiễu bởi vì các thuộc tính trong một số miền có các giá trị hơn là các thuộc tính trong các miền khác và một thuộc tính càng ít giá trị thì dường như có cơ hội làm thông tin tăng thêm lớn. Việc so sánh các mức độ của thuộc tính nhiễu trong toàn thể các miền là rất khó khăn.

Một dạng của nhiễu được thêm vào là cho dữ liệu dạy. Đối với một số tập dữ liệu, trong trường hợp chưa bổ sung nhiễu thì đã khẳng định là cách tiếp cận các mô hình phức có thể rút bớt một số lượng lớn lỗi. Đối với các mô hình này cần khảo sát xem ưu điểm này có bị xói mòn bởi việc bổ sung nhiễu. Bằng thực nghiệm với 4 tập hợp dữ liệu lựa chọn, Ali K. & Pazzani M. cho thấy đối với các mô hình phức ảnh hưởng của nhiễu đối với lỗi thấp hơn so với học máy theo mô

hình đơn. Có thể khẳng định rằng mô hình học máy mô tả phức ổn định hơn so với mô hình đơn. Thực nghiệm cũng chứng tỏ rằng, với các nhiễu gây tác động không tốt cho mô hình phức thì cũng gây tác dụng không tốt cho từng mô hình đơn, ngược lại, một số trường hợp gây nhiễu cho một số mô hình đơn lại không ảnh hưởng đến mô hình phức. Tuy nhiên, mô hình phức không tạo ra sự cải thiện đáng kể so với mô hình đơn khi có tác động của nhiễu. Kết quả thực nghiệm chứng tỏ rằng việc tăng lớp nhiễu đã làm cho cách tiếp cận mô hình phức cũng tạo ra tỉ lệ lỗi tăng lên.

Bằng thực nghiệm cũng đã cho biết dạng phân phối của các lỗi tập hợp. Do đó, thực nghiệm chỉ ra rằng, khi mức độ nhiễu tăng, tất cả các mô hình đều không phân lớp được một ví dụ kiểm tra với một tỉ lệ lớn hơn các ví dụ kiểm tra mà tập hợp lỗi được tạo ra. Điều này chỉ ra rằng, khi lớp nhiễu tăng lên, một số ví dụ kiểm tra trở nên khó hơn cho tất cả các mô hình.

III.2.5. Tác động của thuộc tính không thích hợp

Các thử nghiệm trong [5] cho rằng lợi ích của việc sử dụng cách tiếp cận các mô hình phức tăng lên khi số lượng của các thuộc tính không thích hợp tăng. Có thể khẳng định rằng khi tăng tăng số lượng các thuộc tính không thích hợp thì cũng cho phép làm tăng việc giảm lỗi.

Thông qua việc thêm vào các số lượng khác nhau của các thuộc tính không thích hợp Boolean đối với các mẫu đặc trưng cho các tập hợp dữ liệu có thể quan sát sự giảm lỗi khi học máy. Nguyên nhân của việc chọn các thuộc tính Boolean thay vì chi việc xây dựng các thuộc tính không thích hợp mà các giá trị thuộc vào miền đặc biệt để dễ đối sánh bởi vì các thuộc tính trong một số tập hợp dữ liệu có thể tiếp nhận nhiều giá trị hơn các thuộc tính trong các tập hợp dữ liệu khác làm cho sự so sánh trở nên khó khăn.

Ali K. & Pazzani M. đã chứng tỏ được rằng cách tiếp cận mô hình phức có thể thu được giảm lỗi đáng kể đặc biệt khi nhiều thuộc tính không thích hợp có mặt trong dữ liệu. Hơn nữa tỉ lệ lỗi tăng như là một hàm của việc tăng số lượng

các thuộc tính không thích hợp. Với cách tiếp cận các mô hình phức lỗi tạo ra do nhiều thuộc tính không thích hợp thì ít nhất một nửa số các mô hình được học cần liên quan tới thuộc tính không thích hợp để phân lớp lỗi. Nếu số lượng các thuộc tính không thích hợp không quá lớn thì dường như là ít nhất một nửa các mô hình sẽ bị tác động trong cách này. Do đó cách tiếp cận các mô hình phức sẽ không tạo ra lỗi trong trường hợp này. Nhưng việc tiếp cận mô hình đơn cần duy nhất khi mắc lỗi vì học các luật liên quan thuộc tính không thích hợp sớm trong chiến lược chia nhỏ và chế ngự của nó cho hầu hết các luật tiếp theo để rút lui khỏi phương pháp. Do vậy cách tiếp cận mô hình đơn dường như phải chịu đựng các thuộc tính không thích hợp. Hầu hết các tập hợp dữ liệu bổ sung thêm các thuộc tính không thích hợp thì sẽ cho các tỉ lệ lỗi nhỏ hơn.

Cụ thể hơn, số lượng trung bình của các liên kết tăng lên khi số lượng các thuộc tính không thích hợp tăng. Để áp dụng cho tỉ lệ lỗi thấp hơn thì có thể thu được cho các tập hợp dữ liệu mà ở đó thuật toán học tìm tòi các liên kết tăng lên nhiều hơn.

Tuy nhiên, trong trường hợp một số lượng lớn bất kỳ của các thuộc tính không thích hợp được thêm vào một tập hợp dữ liệu thì kết quả như trên không còn đảm bảo. Như vậy, việc bổ sung các thuộc tính không thích hợp không phải với một số lượng tùy ý được và phải tuân theo một hạn định. Đáng tiếc, hiện chưa có một quan hệ tường minh về số lượng thuộc tính không thích hợp được bổ sung là tốt.

Nói tóm lại, câu trả lời cho câu hỏi được đưa ra với phần này (việc tăng số lượng các thuộc tính không thích hợp có tác động như thế nào tới tổng số lỗi) là toàn bộ các tỷ lệ lỗi giảm khi các thuộc tính không thích hợp được bổ sung do vậy đưa ra cơ hội cho cách tiếp cận các mô hình bội số.

Tuy nhiên, việc bổ sung thêm một số thuộc tính không thích hợp sẽ bắt đầu làm khó khăn cho cách tiếp cận các mô hình bội số và các tỷ lệ lỗi bắt đầu

tăng lên tới một. Trong giới hạn, hoặc cách tiếp cận mô hình đơn hoặc cách tiếp cận các mô hình bội số sẽ được sử dụng nhiều và tỷ lệ lỗi sẽ là một.

III.2.6. Tác động của việc đa dạng hoá

Một loạt các công trình nghiên cứu đưa tới khẳng định là khi tăng sự đa dạng hoá của các mô hình sẽ mang lại sự giảm lỗi nhiều hơn.

Kwok và Carter (1990) khi xem xét trên 2 miền, đã chỉ ra rằng các tập hợp chứa đựng càng nhiều sự đa dạng hoá đối với các cây quyết định về cú pháp thì cho phép thu được các tỉ lệ lỗi càng thấp hơn các tập hợp có chứa ít sự đa dạng hoá các cây quyết định.

Ali K. & Pazzani M. [5] đã cải tiến thuật toán "leo đồi ngẫu nhiên" bằng cách cho phép người sử dụng chỉ rõ kích thước của đơn vị xử lý (được gọi là thùng: bucket). Kích thước này càng lớn thì các tập hợp mà các thành phần trong đó càng đa dạng về cú pháp. Khi lựa chọn sự đa dạng của các miền cho mô tả này bằng cách tạo nên sự khác biệt về nhiễu đối với các tập dữ liệu khác nhau. Tuy nhiên, việc tăng kích cỡ của bucket không phải luôn luôn cho sự tăng lên độ chính xác. Để thu được độ chính xác cao hơn, các mô hình trên được đa dạng hoá và mỗi mô hình phải là rất chính xác.

Các tác giả [5] đã nhận định rằng tất cả các thuộc tính thích hợp được lựa chọn đối với việc học các thuộc tính. Trong thí nghiệm, với tất cả các tập hợp dữ liệu này, tất cả các mô hình thích hợp mà có thể được học thì tương tự về mặt cú pháp do vậy, việc tăng sự đa dạng hoá về mặt cú pháp thì không là một ý tưởng tốt cho loại tập hợp dữ liệu này. Thử nghiệm này cho thấy rằng, mặc dù lý thuyết yêu cầu việc tổ hợp bằng chứng của tất cả các mô hình trong không gian mô hình và giả thiết (Buntine, 1990) nhưng trong thực tế, chỉ một số lượng nhỏ các mô hình được học và do đó nó có thể cần thiết sàng lọc ra các mô hình kém thích hợp để nhằm tối đa hoá độ thích hợp tổng thể.

Các thử nghiệm chỉ ra rằng tối thiểu hoá lỗi tập hợp là cần thiết cho việc cân bằng sự đa dạng được tăng lên với khả năng để đảm bảo các thành phần khác nhau của tập hợp đều thích hợp.

Các thực nghiệm mà Ali K. & Pazzani M. [5] tiến hành đã sử dụng mẫu các tập hợp dữ liệu lớn của kho chứa UCI cho thấy 3 tập hợp dữ liệu mà cách tiếp cận các mô hình phức đưa ra các tỉ lệ lỗi nổi bật: 1/7 cho tập dữ liệu rượu, 1/5 cho tập dữ liệu tic-tac-toe, 1/ 2,5 cho tập dữ liệu DNA.

Đặc biệt, các mô hình phức sẽ cho phép giảm lỗi tốt hơn trong trường hợp mô hình mà dữ liệu đã tương đối chính xác (rút gọn lỗi cho dữ liệu Tic-tac-toe từ 1% xuống còn 0,2%) so với việc rút gọn lỗi trong các miền dữ liệu chứa đựng nhiễu. Tuy nhiên, khi nhân tố hữu hạn không là nhiễu hay sự cản trở của dữ liệu, cách tiếp cận các mô hình phức cho kết quả tốt trong việc thu được giảm lỗi lớn.

Với các tập hợp dữ liệu còn chứa đựng nhiễu dữ liệu, cách tiếp cận các mô hình phức thực sự tốt hơn mô hình đơn khi số lượng của các thuộc tính không thích hợp tăng lên. Như vậy, cách tiếp cận mô hình phức thực sự tốt khi có nhiều liên kết tăng lên. Ali K. & Pazzani M. [5] kết luận rằng, nhân tố cơ bản trong việc giải thích phương sai của giảm lỗi là xu hướng của các mô hình được học để tạo ra các lỗi tương quan. Mặt khác, các tác giả đã cố gắng tăng số lượng của các liên kết đối với mỗi tập hợp dữ liệu bằng cách bổ sung thêm một số thuộc tính không thích hợp cho với mỗi tập hợp dữ liệu. Điều này làm tăng số lượng của các liên kết tăng lên được biết và cũng tạo ra sự giảm lỗi mạnh hơn.

CHƯƠNG 4. THUẬT TOÁN TÌM KIẾM VÀ PHÂN LỚP TRONG CƠ SỞ DỮ LIỆU FULL-TEXT

IV.1. CƠ SỞ DỮ LIỆU FULL-TEXT

IV.1.1 Khái niệm về cơ sở dữ liệu full-text

Các mô hình dữ liệu điển hình là mô hình phân cấp, mô hình mạng và mô hình quan hệ được biết như những mô hình dữ liệu có cấu trúc: dữ liệu trong hệ thống được liên kết nhau theo mỗi cấu trúc tương ứng. Cấu trúc hệ thống cho biết mối liên hệ lẫn nhau giữa các đối tượng trong phạm vi xem xét. Chẳng hạn, trong mô hình quan hệ, thông tin về các đối tượng dữ liệu được trình bày dưới dạng bảng, các đối tượng dữ liệu có sự tương ứng đồng nhất nhau theo các trường thông tin chung cho bảng. Tuy nhiên, đối với nhiều hệ thống thông tin, không phải luôn luôn biểu diễn được toàn bộ dữ liệu theo những cấu trúc đã có. Chẳng hạn, không thể xây dựng được một cấu trúc nhằm biểu diễn đối tượng thông tin văn bản, hình ảnh. Dạng thông tin như vậy được gọi là thông tin phi cấu trúc; nó có thể là một hình vẽ, một văn bản, dữ liệu multimedia v.v. Cơ sở dữ liệu quản lý thông tin phi cấu trúc được gọi là cơ sở dữ liệu phi cấu trúc. Thông thường, thông tin trong một cơ sở dữ liệu phi cấu trúc bao gồm hai phần: phần có cấu trúc chứa đựng thông tin chung nhất về toàn bộ đối tượng (do tìm được cấu trúc biểu diễn chúng) và phần phi cấu trúc chứa đựng thông tin riêng về từng đối tượng.

Dữ liệu dạng full-text là một dạng dữ liệu phi cấu trúc với thông tin văn bản dạng text. Mỗi văn bản chứa thông tin về một vấn đề nào đó được thể hiện qua nội dung của tất cả các từ cấu thành văn bản đó và cách liên kết các từ này. Ý nghĩa của một từ trong văn bản không cố định mà tùy thuộc vào từng ngữ cảnh: ngữ cảnh khác nhau sẽ cho ý nghĩa khác nhau. Các từ trong văn bản lại được liên kết với nhau bởi ngôn ngữ trình bày đó. Để có thể hiểu được nghĩa của văn bản không những cần phải hiểu được nội dung của các từ mà còn phải nắm

được ngữ pháp của ngôn ngữ đó. Điều này đối với con người là khá đơn giản bởi chính họ tạo ra ngôn ngữ và các văn bản đó để trình bày vấn đề. Tuy nhiên việc lập trình để làm cho máy tính hiểu được nội dung của một văn bản khi tiếp nhận văn bản đó thì lại là một vấn đề hết sức phức tạp.

Hệ cơ sở dữ liệu Full-text quản lý các văn bản Full-text và tổ chức ***tìm kiếm theo nội dung*** (liên quan đến phần phi cấu trúc của hệ thống). Người sử dụng đưa ra nội dung cần tìm và hệ thống sẽ trả lại các văn bản có nội dung liên quan. Như vậy hệ thống phải nắm bắt được nội dung của mọi văn bản, hay tương ứng, hệ cơ sở dữ liệu Full-text phải có một hệ thống đoán nhận được ngôn ngữ diễn đạt văn bản đó. Hệ thống này là riêng biệt đối với mỗi ngôn ngữ và thực hiện được điều này là hết sức khó khăn và tốn kém.

Một cách làm khác để nắm bắt nội dung của văn bản là thông qua việc nắm bắt các từ và cụm từ trong văn bản đó theo hướng suy nghĩ là các từ trong văn bản cũng phản ánh phần nào ý nghĩa của nội dung văn bản. Cách này kém chính xác hơn nhưng có độ phức tạp vừa phải để có thể giải quyết được mà ít tốn kém hơn về thời gian cũng như về công sức. Trong mỗi ngôn ngữ, số các từ có nghĩa là hữu hạn, vì vậy, thay vì phải quản lý toàn bộ nội dung văn bản, người ta chỉ quản lý các từ có nghĩa được dùng để thể hiện nội dung văn bản đó. Đặc biệt, nếu như hệ thống quản lý các văn bản liên quan đến một lĩnh vực hoạt động riêng biệt thì danh sách các từ cần quản lý được giới hạn theo các từ có nghĩa thuộc lĩnh vực hoạt động riêng biệt đó và vì vậy, kích thước thông tin quản lý sẽ nhỏ. Nội dung cần tìm cũng được thể hiện thông qua các từ hoặc cụm từ. Về mặt thực tế, do tính dễ thực hiện, người ta hay dùng cách này để xây dựng hệ CSDL full-text và đang tìm cách cải tiến chúng để đạt được kết quả tốt hơn.

Ngoài chức năng quản lý và tìm kiếm các văn bản theo nội dung, hệ CSDL full-text cần có chức năng phân lớp các văn bản theo một số mẫu đã định sẵn (được gọi là catalog văn bản, mỗi lớp được gọi là một catalog). Dựa trên một số lớp văn bản đã được định sẵn theo các văn bản mẫu của từng lớp văn bản, thuật

toán phân lớp sẽ được thực hiện khi hệ thống có thêm một văn bản mới. Kết hợp hai thuật toán phân lớp và tìm kiếm sẽ cho tác dụng thu hẹp lĩnh vực tìm kiếm văn bản và do đó, việc tìm kiếm sẽ chính xác hơn. Người dùng sẽ nhận được các văn bản trong phạm vi catalog đã chọn ra và như vậy tránh được tình trạng việc tìm kiếm phải xảy ra trên toàn bộ dữ liệu của hệ thống.

IV.1.2. Các nội dung cơ bản của một cơ sở dữ liệu full-text

a/ Lưu trữ văn bản

Văn bản là đối tượng quản lý chính của hệ thống. Bản thân các văn bản chắc chắn phải được lưu giữ tại một nơi nào đó ở bên trong hoặc bên ngoài hệ thống và các thông tin về chúng được quản lý tuỳ thuộc vào thuật toán lưu trữ và tìm kiếm. Có hai cách lưu trữ văn bản sau đây:

Lưu trữ trực tiếp

Các văn bản được lưu trữ trực tiếp trong cột của một bảng như là một trường của bảng đó. Khi đó một văn bản gồm các thuộc tính sau:

- Khoá (mã) văn bản
- Nội dung văn bản: lưu trực tiếp nội dung của văn bản. Theo cách này, văn bản thực chất là một trường có kiểu TEXT trong một bảng. Chẳng hạn, nội dung văn bản được chỉ dẫn từ trường memo trong file dữ liệu của FOXPRO là một biến dạng của cách này.
- Các thông tin khác về văn bản như ngày truy nhập, độ dài của văn bản. Các thông tin này không có tác dụng trong quá trình tìm kiếm theo nội dung mà chỉ bổ sung cho người dùng một số thông tin về một văn bản xác định.

Lưu trữ gián tiếp

Văn bản không được lưu trữ trực tiếp bên trong CSDL mà được lưu tại một nơi nào đó ở bên ngoài CSDL. Việc quản lý các văn bản sẽ thông qua địa chỉ của nó. Địa chỉ có thể là:

- Đường dẫn đến một file

- Địa chỉ URL trỏ đến một trang Web
- Đường dẫn đến một trường dạng Text trong một CSDL khác.

Nếu dữ liệu được lưu vào một file thì phải chú ý đến khả năng truy cập đến file đó, địa chỉ sẽ gồm:

- Tên file
- Nơi đặt file: phải đặt tại nơi mà chương trình thực hiện có thể truy cập đến được.

Thông thường người ta hay sử dụng theo phương pháp lưu trữ ngoài vì khi đó, một mặt, không bị hạn chế về độ dài văn bản lưu trữ, và mặt khác, cho phép mở rộng phạm vi lưu trữ văn bản trên mạng tùy ý.

Trong cách thức này, thông tin về một văn bản gồm:

- Mã văn bản,
- Con trỏ trỏ đến nơi chứa văn bản hay thông tin về địa chỉ của một file,
- Các thông tin về nội dung văn bản để có thể tiến hành tìm kiếm chúng,
- Ngoài ra còn có một số thông tin phụ về văn bản đó như ngày truy cập gần nhất, độ dài văn bản, loại văn bản .. .

b/ Câu hỏi tìm kiếm

Câu hỏi tìm kiếm là bất kỳ câu hỏi nào mà hệ thống cho phép người sử dụng đưa ra nhằm thể hiện yêu cầu tìm kiếm văn bản và điều này là khá rộng rãi. Có nhiều cách đưa ra câu hỏi tuỳ thuộc vào cách tìm kiếm của hệ thống và yêu cầu của người dùng.

Tìm kiếm theo các từ hoặc cụm từ

Câu hỏi tìm kiếm được trình bày qua các từ hoặc cụm từ được liên kết bởi các phép toán logic nhằm diễn tả được nội dung của văn bản. Đơn giản nhất có thể là đưa ra một từ hoặc cụm từ và yêu cầu trả lại các văn bản có chứa từ hoặc cụm từ đó cùng với số lần xuất hiện chúng trong văn bản. Phức tạp hơn, dùng các phép toán lôgic để biểu diễn các từ có thể liên quan đến nhau và có ảnh hưởng khác nhau đến nội dung cần tìm.

Chẳng hạn, có thể đặt ra yêu cầu tìm kiếm như sau: Tìm văn bản có chứa các từ 'ruộng đất', 'luật','đất đai','Điều' nhưng không chứa từ 'thuế' và từ 'cá nhân'.

Ngoài ra có thể đưa ra các yêu cầu bổ sung khác như:

- Khoảng cách giữa các từ cần tìm có thể liên quan đến nội dung văn bản.

Chẳng hạn, đưa ra yêu cầu: Tìm văn bản có chứa các từ 'thu' và từ 'nhập', nếu hai từ này gần nhau thì mức độ liên quan của văn bản tìm được là càng cao. Khi đó văn bản có chứa câu 'thu nhập' được đánh giá là có độ chính xác cao hơn văn bản có chứa câu 'thu và nhập'.

- Câu hỏi đưa vào có thể ở dạng mở rộng hoặc được thể hiện bằng cách dùng một ký tự đặc biệt thay cho một từ hoặc một nhóm từ.

Chẳng hạn, đưa ra yêu cầu: Tìm văn bản có chứa nhóm ký tự là an* hoặc thuy*. Có thể cho phép đưa ra câu hỏi dạng này, tuy nhiên nếu như việc tìm kiếm là theo nội dung thì câu hỏi trên hoàn toàn không có ý nghĩa gì cả. Đối với trường hợp này, hệ thống phải tham khảo và thao tác trên các từ khoá thoả mãn yêu cầu trên.

Tìm kiếm theo chủ đề

Câu hỏi tìm kiếm theo nội dung của văn bản cần tìm. Yêu cầu trước đó các văn bản phải được xác đinh bởi một nội dung theo một cách nào đó (nhập bằng tay hoặc phân tích cú pháp..). Câu hỏi đưa vào cũng có thể được phân tích cú pháp đưa ra một chủ đề tương ứng. Phép tìm kiếm sẽ được tiến hành trên các bảng index của các chủ đề văn bản và chủ đề câu hỏi.

Các câu hỏi đưa vào được lưu trữ trong một bảng. Hệ thống sẽ quản lý một dãy các yêu cầu này trong bảng và giải quyết chúng theo một cách tuần tự.

c/ Bảng kết quả

Bảng kết quả là bảng lưu trữ toàn bộ các kết quả tìm được trong quá trình tìm kiếm và phân lớp văn bản. Kết quả trong bảng không phải là cố định mà chỉ có giá trị đối với từng câu hỏi, tại một thời điểm nhất định. Tuỳ thuộc vào mỗi phương pháp xử lý mà thông tin trong bảng kết quả là khác nhau.

V.1.3. Các mô hình quản lý và lưu trữ thông tin văn bản

a/ Mô hình logic

Theo mô hình này, các từ có nghĩa trong văn bản được gán chỉ số (index), và người ta quản lý nội dung của văn bản theo các chỉ số đó.

Các luật lưu trữ và tìm kiếm

Việc xây dựng hệ thống theo mô hình này được thực hiện như sau:

- Mỗi văn bản đều được chỉ số hóa theo luật:

- Thống kê các từ có nghĩa trong các văn bản, đó là những từ mang thông tin chính về các văn bản lưu giữ.

- Chỉ số hoá các văn bản đưa vào theo danh sách các từ khoá nói trên. Ứng với mỗi từ khoá trong danh sách sẽ lưu vị trí xuất hiện nó trong từng văn bản và tên văn bản mà tồn tại từ khoá đó.

- Câu hỏi tìm kiếm được trình bày dưới dạng logic tức là gồm một dãy các phép toán lôgic (AND, OR, NOT ...) thực hiện trên các từ hoặc cụm từ.

Ví dụ về câu hỏi: Tìm văn bản trong đó có chứa từ "hệ thống" *và* từ "CSDL" nhưng *không* chứa từ "quan hệ"

Việc tìm kiếm sẽ dựa vào bảng chỉ số đã tạo ra và trả lại kết quả là các văn bản thoả mãn toàn bộ các điều kiện trên.

Ưu và nhược điểm

- Ưu điểm

- Tìm kiếm nhanh và đơn giản.

Thực vậy giả sử cần tìm kiếm từ "mẹ". Hệ thống sẽ duyệt trên bảng chỉ số để trỏ đến chỉ số tương ứng nếu như từ "mẹ" tồn tại trong hệ thống. Việc tìm

kiếm này khá nhanh và đơn giản khi trước đó ta đã sắp xếp bảng chỉ số theo vần chữ cái. Phép tìm kiếm trên sẽ có độ phức tạp cấp $n\log_2 n$ (với n là số từ được chỉ số hoá trong bảng chỉ số). Tương ứng với chỉ số trên sẽ cho ta biết các văn bản chứa nó. Như vậy nếu việc tìm kiếm liên quan đến k từ thì số các phép toán cần thực hiện sẽ là $k*n*\log_2 n$ với n là số văn bản đang có.

- Câu hỏi về các từ tìm kiếm là linh hoạt.

Có thể dùng các ký tự đặc biệt trong câu hỏi tìm kiếm mà không làm ảnh hưởng đến độ phức tạp của phép tìm kiếm. Ví dụ từ "bố%" sẽ trả lại tất cả các văn bản có chứa những từ như "bố", "bốn", "bống", "bốt".. là các từ được bắt đầu bằng từ "bố". Ký tự "%" được gọi là kí tự thay thế.

Ngoài ra bằng các phép toán logic các từ cần tìm có thể được tổ chức thành các câu hỏi một cách linh hoạt. Ví dụ từ cần tìm là [vợ, vợi, vương], dấu [.] sẽ thể hiện việc tìm kiếm trên một trong số nhiều từ trong nhóm. Đây thực ra là một cách thể hiện linh hoạt phép toán OR trong đại số logic thay vì phải viết là tìm văn bản có chứa hoặc từ "vợ" hoặc từ "vợi" hoặc từ "vương"..

• Nhược điểm

- Người tìm kiếm phải có chuyên môn trong lĩnh vực mình tìm kiếm.

Thực vậy, do câu hỏi đưa vào dưới dạng logic nên kết quả trả lại cũng có giá trị Boolean, một số văn bản sẽ được trả lại khi thoả mãn mọi điều kiện đưa vào. Như vậy, muốn tìm được văn bản theo nội dung thì người tìm kiếm phải biết chính xác về các từ ngữ có trong văn bản đó và mối quan hệ giữa chúng, gây khó khăn cho việc tìm kiếm.

- Việc chỉ số hóa văn bản là phức tạp và tốn nhiều thời gian.

- Tốn nhiều không gian lưu trữ các bảng chỉ số.

- Các văn bản được tìm không thể sắp xếp theo độ chính xác của chúng.

Do giá trị trả lại là toàn bộ các văn bản thoả mãn điều kiện nên số văn bản trả lại không biết trước với số lượng có thể là rất nhiều. Mặt khác lại không có một tiêu chuẩn nào để đánh giá chất lượng độ liên quan của các văn bản trả

lại với câu hỏi đặt ra trong khi những người tìm kiếm thường xuyên muốn trả lại số các văn bản có giá trị là ít nhất với độ liên quan là lớn nhất. Điều này vẫn có thể giải quyết được bằng cách lưu các chỉ số trong quá trình chỉ số hóa nhưng rất phức tạp trong việc truy xuất và tính toán số liệu.

- Các bảng chỉ số không linh hoạt. Khi các từ khoá trong bảng thay đổi (thêm, xoá..) thì chỉ số của các văn bản cũng phải thay đổi theo.

b/ Mô hình phân tích cú pháp

Thuật toán lưu trữ và tìm kiếm

Việc xây dựng hệ thống theo mô hình này phải tuân theo các luật sau:

- Mỗi văn bản đều phải được phân tích cú pháp và trả lại thông tin chi tiết về chủ đề của văn bản đó. Chủ đề của các văn bản được tạo ra ở nhiều mức trừu tượng khác nhau phụ thuộc vào mức độ chi tiết nội dung văn bản của yêu cầu người dùng.
- Các văn bản được quản lý thông qua các chủ đề này để có thể tìm kiếm được khi có yêu cầu.
- Câu hỏi tìm kiếm sẽ dựa trên các chủ đề trên, như vậy trước đó sẽ tiến hành chỉ số hóa theo chủ đề.
- Cách chỉ số hóa theo chủ đề giống như khi chỉ số hoá theo văn bản nhưng chỉ số hóa trên toàn bộ các từ có trong chủ đề đó.
- Câu hỏi đưa vào cũng có thể được phân tích cú pháp để trả lại một chủ đề và tìm kiếm trên chủ đề đó.

Như vậy bộ phận xử lý chính đối với một hệ CSDL xây dựng theo mô hình này chính là hệ thống phân tích cú pháp và đoán nhận nội dung văn bản.

Đánh giá chung về phương pháp

Chất lượng của hệ thống theo phương pháp này hoàn toàn phụ thuộc vào chất lượng của hệ thống phân tích cú pháp và đoán nhận nội dung văn bản. Trên

thực tế việc xây dựng hệ thống này là rất phức tạp, phụ thuộc vào đặc điểm của từng ngôn ngữ, và đa số vẫn chưa đạt được độ chính xác cao.

Tuy nhiên, khi đã có chủ đề thì việc tìm kiếm theo phương pháp này lại khá hiệu quả do tìm kiếm nhanh và chính xác. Mặt khác, đối với những ngôn ngữ đơn giản về mặt ngữ pháp thì việc phân tích trên là có thể đạt được mức độ chính xác cao và chấp nhận được.

c/ Mô hình vector

Mô hình vector sẽ được trình bày chi tiết trong phần sau (mục IV.2.1). Trong mô hình này, việc lưu trữ và tìm kiếm dựa theo sự xuất hiện các từ có nghĩa trong câu hỏi và các tài liệu tương ứng. Mô hình sử dụng một vector về sự xuất hiện các từ có nghĩa trong văn bản để biểu diễn văn bản và hệ thống tìm kiếm dựa trên việc xem xét các vector này. Các văn bản được lưu trữ ngoài, việc hỏi và tìm kiếm theo nội dung được thực hiện theo các từ.

IV.2. THUẬT TOÁN TÌM KIẾM VÀ PHÂN LỚP TRONG CƠ SỞ DỮ LIỆU FULL-TEXT THEO MÔ HÌNH VECTOR CẢI TIẾN

Mô hình vector là mô hình tổ chức quản lý, tìm kiếm các tài liệu Full-text theo các từ và cụm từ. Nó cho phép người dùng tìm ra được những tài liệu cần thiết khi nhập vào một số từ hoặc cụm từ. Hệ thống sẽ đưa ra danh sách các tài liệu có chứa các từ đó. Đặc điểm nổi bật của mô hình này là các tài liệu lưu trữ và câu hỏi tìm kiếm đều biểu diễn dưới dạng một vector. Việc tìm kiếm tài liệu được thực hiện trên hệ thống các vector.

Trong phần này chúng ta trình bày chi tiết về mô hình vector với một số cải tiến nhỏ của chúng ta, thuật toán tìm kiếm và một số thuật toán phân lớp dựa trên mô hình đã cải tiến đó.

IV.2.1. Mô hình vector cải tiến và thuật toán tìm kiếm

a/ Thuật toán lưu trữ

Giả sử D= {d_1, d_2, d_3, ... , d_n ...} là tập hợp các văn bản mà hệ thống cần quản lý, trong đó d_i là các văn bản mang thông tin thuộc lĩnh vực cần quan tâm.

T = {t_1, t_2, t_3 , ... , t_m} là tập hợp hữu hạn các từ có nghĩa trong lĩnh vực đang được quan tâm có trong các văn bản và được gọi là các từ khoá.

Hệ thống biểu diễn nội dung của văn bản thông qua việc kiểm tra sự có mặt của mỗi từ khoá trong văn bản bằng cách thực hiện một ánh xạ F từ D vào V trong đó V là tập hợp các vector có m thành phần (m là số lượng từ khóa trong hệ thống).

$$F: D \rightarrow V$$

Với mỗi văn bản d thuộc tập hợp D, F(d) sẽ xác định một vector v: v(d) = (v_1, v_2, ..., v_m) trong đó v_i giá trị phản ánh sự xuất hiện của từ khóa t_i trong văn bản d, v_i là 0 - khi không xuất hiện hoặc 1 - khi có xuất hiện (không kể là xuất hiện bao nhiêu lần). Như vậy, thành phần trong vector v được xác định theo luật sau:

- Nếu t_i có mặt trong d ta có $v_i = 1$,
- Nếu t_i không có mặt trong d ta có $v_i = 0$

Ví dụ, với D = {d_1,d_2} trong đó:

d_1 là "Cộng hoà, xã hội, chủ nghĩa, Việt Nam"

d_2 là "Độc lập, tự do, hạnh phúc"

T= {Cộng hoà, độc lập, tự do, chủ nghĩa, bảo thủ, đảng}

ta có $v(d_1) = (1,0,0,1,0,0)$ và $v(d_2) = (0,1,1,0,0,0)$

Câu hỏi tìm kiếm Q thuộc dạng tìm kiếm theo từ và được đưa vào dưới dạng liệt kê sự xuất hiện các từ trong ngôn ngữ đã cho (không nhất thiết là từ khóa) có liên quan đến nội dung văn bản cần tìm, Q = {q_1, q_2, ..., q_k}.

Ta cũng lấy F(Q) giống như đã làm với văn bản d, kết quả được một vector

$$P = (p_1, p_2, \dots , p_m)$$

Với mỗi văn bản d, đặt

$$A(d) = P * v(d) = \sum_{i=1}^{m} p_i * v_i \qquad (4.1)$$

Khi đó, hệ số A(d) được coi là mức độ liên quan của văn bản d đối với nội dung cần tìm.

Ví dụ, giả sử Q = {Đảng, bảo thủ, chiếm, đa số, phiếu bầu} thì P = (0, 0, 0, 0, 1, 1).

Ta có một số nhận xét sau đây:

- A=0 khi:

- Với mọi i, p_i=0 hay các từ trong câu hỏi không có mặt trong tập từ khóa.

- Với mọi j, v_j = 0 hay các từ khoá không có mặt trong văn bản cần lưu trữ,

- Với mọi j, v_j*p_j=0 hay mọi từ có trong câu hỏi tìm kiếm đều không có trong văn bản d.

- A>0 khi tồn tại j thoả mãn v_j*$p_j \neq 0$ hay cũng vậy, tồn tại ít nhất một từ khoá vừa có mặt trong văn bản d vừa có mặt trong câu hỏi tìm kiếm.

- A = m khi mọi từ khóa trong từ điển đều xuất hiện trong văn bản d.

Theo cách tính trên khi đưa vào một câu hỏi q, với mỗi văn bản d, sẽ cho tương ứng một giá trị A(d). Nếu A(d) càng lớn thì có thể quan niệm rằng độ liên quan của văn bản với câu hỏi càng nhiều. Trong các hệ thống thực tế, không làm giảm tổng quát ta luôn giả thiết là trong văn bản hay câu hỏi có ít nhất một từ khóa.

Thuật toán tìm kiếm theo câu hỏi Q (câu hỏi Q được trình bày qua vector P) được mô tả như sau:

- Với mọi d $\in$ D, tính giá trị A(d) = P * v(d) ,

- Sắp xếp các giá trị A(d) theo thứ tự giảm dần,

- Hiện nội dung các văn bản d theo thứ tự giảm dần đã có cho đến khi thỏa mãn yêu cầu người dùng.

b/ Mô hình vector cải tiến

Việc cải tiến mô hình vector trong phần này là sự phát triển một số cải tiến đã được đề cập trong [4].

Cải tiến theo hướng từ đồng nghĩa và số lần xuất hiện

Một điều có tính phổ biến là trong mọi ngôn ngữ tự nhiên luôn tồn tại những từ đồng nghĩa, chẳng hạn, trong tiếng Việt các từ "an dưỡng", "an trí" có cùng một nghĩa là với từ "nghỉ". Như vậy, cùng một vấn đề có thể dùng nhiều từ khác nhau để biểu đạt ý nghĩa của nó. Trong khi đó nội dung trong các câu hỏi thông thường chỉ được diễn đạt bằng một từ duy nhất. Do vậy, trong quá trình tìm kiếm nếu như hệ thống được tổ chức không tốt thì sẽ chỉ tìm kiếm được các tài liệu có chứa các từ được đưa ra trong câu hỏi mà không tìm được các tài liệu có cùng nội dung với các cách thể hiện khác.

Các từ thuộc nhóm từ đồng nghĩa đều thuộc tập các từ có nghĩa đã được liệt kê khi thiết kế hệ thống. Trong một nhóm các từ đồng nghĩa, mặc dù cùng biểu đạt một nội dung nhưng vai trò của các từ có thể sẽ khác nhau do các lý do sau: Với nội dung đó, từ này hay được sử dụng hơn từ kia. Chẳng hạn, các từ trong nhóm đồng nghĩa (nghỉ, an dưỡng, an trí) thì từ "nghỉ" được sử dụng nhiều hơn là "an dưỡng" hay "an trí". Chúng ta chọn trong nhóm từ đồng nghĩa một từ được sử dụng nhiều nhất làm từ đại diện và chỉ từ đại diện xuất hiện trong bảng từ khóa T. Một bảng tra cứu về các từ đồng nghĩa liên quan đến một từ khoá được bổ sung trong thuật toán. Một từ khóa với *nhóm đồng nghĩa* chỉ có từ khóa đó gọi là từ *đơn nghĩa*. Sau khi đã phân tích như trên, ta có thể biểu diễn hệ số của các từ trong nhóm từ đồng nghĩa trên như sau (mọi từ đại diện có hệ số 10):

Từ "nghỉ" có hệ số = 10

Từ "an dưỡng" có hệ số = 9

Từ "an trí" có hệ số = 8.

Việc thống kê các từ đồng nghĩa và đánh giá về hệ số của các từ đồng nghĩa trong nhóm là một việc khá phức tạp đòi hỏi phải có kiến thức về ngữ nghĩa cuả từ trong ngôn ngữ. Vì vậy các nhóm từ đồng nghĩa trong hệ thống cần

phải thông qua việc đánh giá từ các nhà ngôn ngữ học. Khi hệ thống cho phép nhập lại hoặc bổ sung các nhóm từ đồng nghĩa để có thể làm tăng độ chính xác thì hiệu quả hệ thống sẽ tăng.

Một nội dung cần quan tâm là số lần xuất hiện của từ khoá (cùng các từ đồng nghĩa với nó) trong văn bản. Một cách đặt vấn đề khá hợp lý là nếu văn bản D gặp nhiều từ khoá nào đó hơn văn bản D' thì D có thể mang nghĩa của từ khóa đó nhiều hơn văn bản D'. Chính vì lẽ đó, cùng với giải pháp từ đồng nghĩa, chúng ta nhấn mạnh thêm số lần xuất hiện từ khóa trong văn bản. Những nội dung trình bày dưới đây thể hiện các cách đặt vấn đề trên.

Đặt tập mọi từ có nghĩa $T^* = T \cup$ {từ đồng nghĩa} và hàm $h: T^* \rightarrow N$ (số nguyên dương) trong đó, $\forall t \in T^*$: h(t) chính là hệ số của từ t.

Việc tìm kiếm và mã hoá sẽ được tiến hành trên hệ thống các từ khóa. Trong bảng vector mã hoá, mỗi từ đồng nghĩa sẽ được đại diện bởi một mã nhóm duy nhất, như vậy sẽ giảm được độ dài vector mã hoá đồng thời giảm được các phép tính cần thiết trong quá trình tìm kiếm. Khi mã hoá tài liệu (nhờ ánh xạ F), cách tính giá trị vector của một từ đồng nghĩa cũng khác so với cách tính giá trị vector của một từ đơn nghĩa và được tính theo cách dưới đây.

Với văn bản d, $F(d) = v = (v_1, v_2, ..., v_m)$ được tính như sau:

$$v_i = \sum_{t \in T^* \cap d} h(t)$$

Với câu hỏi $Q = \{q_1, q_2, ..., q_k\}$ thuật toán thiết lập vector $P = (p_1, p_2, ... , p_m)$ theo ánh xạ từ q_i tới từ khóa tương ứng (nếu có). Như vậy nếu q_i là từ khóa t_j (hoặc từ đồng nghĩa với t_j) thì $p_j = 1$ còn ngược lại $p_j = 0$.

Và như vậy, hệ số liên quan $A = P * v$ theo công thức (4.1) cho phép làm tăng độ chính xác của hệ thống.

Một vấn đề đáng quan tâm song không xem xét trong hệ thống của chúng ta là vần đề đa nghĩa của hệ thống từ có nghĩa. Đây là một nội dung rất lý thú đòi hỏi nhiều công sức nghiên cứu.

Cải tiến theo hướng gán trọng số cho các từ thuộc câu hỏi

Câu hỏi đưa vào dưới dạng Q = {q_1, q_2, ..., q_k} trong đó mỗi từ q_i lại có một hệ số c_i thể hiện tầm quan trọng khác nhau của các từ trong câu hỏi. Ánh xạ Q thành vector P (p_1, p_2, ..., p_m) thì giá trị của p_i được tính theo các trọng số này, tức là nếu q_i là từ khóa t_j (hoặc từ đồng nghĩa với t_j) thì $p_j = c_i$ (trọng số của q_i) còn ngược lại $p_j = 0$.

Hệ số liên quan A = P * v theo công thức (4.1) cũng thể hiện được trọng số của các từ trong câu hỏi Q.

c/ Đánh giá về các ưu nhược điểm của thuật toán

- Ưu điểm

- Các văn bản trả lại có thể được sắp xếp theo mức độ liên quan đến nội dung yêu cầu do trong phép thử mỗi văn bản đều trả lại chỉ số đánh giá độ liên quan của nó đến nội dung yêu cầu.
- Việc đưa ra các câu hỏi tìm kiếm là dễ dàng và không yêu cầu người tìm kiếm có trình độ chuyên môn cao về vấn đề đó.
- Tiến hành lưu trữ và tìm kiếm đơn giản hơn phương pháp logic.
- Người tìm kiếm có thể tự đưa ra số các văn bản trả lại có mức độ chính xác cao nhất.
- Mô hình lưu trữ và tìm kiếm vector là phù hợp với thuật toán phân lớp văn bản (được trình bày chi tiết trong phần IV.2.2-IV.2.4) vừa có tác dụng trong việc phân loại văn bản lại có ý nghĩa hỗi trợ cho bài toán tìm kiếm văn bản.

- Nhược điểm

- Việc tìm kiếm tiến hành khá chậm khi hệ thống các từ khoá là lớn do phải tính toán trên toàn bộ các vector của các văn bản.

Khi biểu diễn các vector với các hệ số là số tự nhiên làm tăng mức độ chính xác của phép tìm kiếm nhưng sẽ làm giảm tốc độ tính toán đi rất nhiều do các phép nhân vector phải tiến hành trên các số tự nhiên hoặc số thực, hơn nữa việc lưu trữ các vector sẽ phức tạp và tốn kém.

- Hệ thống không linh hoạt khi lưu trữ các từ khoá. Chỉ cần một thay đổi rất nhỏ trong bảng từ khoá sẽ kéo theo hoặc là vector hoá lại toàn bộ các văn bản lưu trữ hoặc sẽ bỏ qua các từ có nghĩa bổ sung trong các văn bản được mã hoá trước đó.

Tuy nhiên với những ưu điểm nhất định, sự sai số nhỏ này có thể được bỏ qua do hiện tại số các từ có nghĩa được mã hoá đã khá đầy đủ trước khi tiến hành mã hoá các văn bản. Vì vậy, phương pháp vector vẫn được quan tâm và sử dụng.

d/ Về tốc độ thực hiện chương trình

Tốc độ thực hiện chương trình được đánh giá qua tốc độ trong quá trình chế biến, lưu trữ tài liệu và tốc độ tìm kiếm tài liệu.

Tài liệu trước khi lưu trữ trong hệ thống sẽ được mã hoá thành các vector bằng cách duyệt từng từ trong tài liệu. Với một số lượng lớn các từ khoá đồng thời số các từ ngữ trong một tài liệu là nhiều thì quá trình này diễn ra khá chậm.

Việc tìm kiếm tài liệu diễn ra gồm hai quá trình: quá trình mã hoá câu hỏi và quá trình thao tác trên các vector. Do số lượng từ trong câu hỏi đưa vào thông thường là ít nên thời gian mã hoá câu hỏi tương đối nhỏ. Trong khi đó thời gian dành cho các thao tác trên các vector phụ thuộc hoàn toàn vào độ dài các vector hay số lượng các phép tính giữa câu hỏi với các vector mã hoá của tài liệu.

Hệ thống xây dựng theo phương pháp vector phải quản lý các từ có nghĩa trong tài liệu. Số lượng các từ có nghĩa được quản lý trong hệ thống phụ thuộc vào yêu cầu đối với hệ thống đó.

Để hạn chế các phép tính trong giai đoạn này ta có thể giảm độ dài của vector mã hoá các tài liệu bằng việc từ khóa đại diện cho mỗi nhóm từ đồng nghĩa.

IV.2.2. Thuật toán phân lớp Bayes thứ nhất

Tương ứng với cách thức mà Dunja Mladenic' đã trình bày trong [11], kết hợp với mô hình vector trên đây, chúng ta sử dụng thuật toán phân lớp Bayes để phân lớp một tài liệu theo n nhóm tài liệu đã định trước. Thuật ngữ "catalog" tương ứng với thuật ngữ "nhóm" trong bài toán phân lớp. Khi áp dụng thuật toán phân lớp Bayes, việc xác định xác suất tiên nghiệm và xác suất hậu nghiệm hợp lý là một vấn đề có tính cốt lõi. Tồn tại hai bộ phân lớp Bayes để giải quyết bài toán phân lớp tài liệu nói trên. Trong phần này, chúng ta nghiên cứu bộ phân lớp Bayes thứ nhất.

Trong các thuật toán phân lớp dưới đây, sử dụng mô hình vector để biểu diễn các tài liệu.

Ngoài các tham số cơ bản của mô hình vector, hệ thống còn cần thêm các tham số sau đây:

- Tập hợp n catalog $\{C_1, C_2, ..., C_n\}$ trong đó mỗi catalog C_i có một số tài liệu mẫu chỉ dẫn về catalog đó. Trong mô hình này, thông tin về một catalog chỉ có được từ nhóm các tài liệu mẫu tương ứng với nó. Việc chọn tài liệu mẫu vào mẫu catalog dựa theo kinh nghiệm của các chuyên gia.
- Xác suất $P(C_i)$: xác suất được gán cho các catalog C_i. Giá trị này mang ý nghĩa là tài liệu được phân không đều vào các catalog. Điều này bắt nguồn từ thực tế là số lượng các tài liệu trong các catalog là không đều nhau. Do vậy, từng catalog sẽ được gán một giá trị $P(C_i)$. Catalog nào có giá trị $P(C_i)$ lớn chứng tỏ số lượng các tài liệu trong nó là nhiều. Đây là một trong các giá trị có ảnh hưởng đến độ chính xác của bộ phân lớp.
- Ngưỡng $CtgTsh_i$ để kiểm tra xem tài liệu Doc đã cho có đúng thuộc vào catalog C_i đó hay không, mỗi catalog có một ngưỡng riêng.

Với mỗi lớp catalog C, sau khi thực hiện phân lớp, hệ thống sẽ sinh ra một giá trị tương ứng P(C/Doc) - đây là xác suất để phân tài liệu Doc thuộc vào C.

Tài liệu Doc sẽ được coi là thuộc vào C nếu P(C/Doc) ≥ CtgTsh tương ứng, ngược lại, khi (P(C/Doc) < CtgTsh) thì tài liệu Doc không thuộc vào catalog C.

Kết quả là hệ thống sinh ra các xác suất $P(C_i/Doc)$ và cuối cùng sẽ quyết định xem tài liệu Doc đã cho thuộc vào catalog nào. Các xác suất $P(C_i/Doc)$ được gọi là xác suất hậu nghiệm.

Giá trị P(C/Doc) - xác suất tài liệu Doc thuộc vào catalog C được tính toán dựa vào công thức sau:

$$P(C|Doc) = \frac{P(C) \times \prod_{F_j \in T} P(F_j|C)^{TF(F_j, Doc)}}{\sum_{i=1}^{n} P(C_i) \times \prod_{F_l \in T} P(F_l|C_i)^{TF(F_l, Doc)}} \qquad (4.2)$$

và

$$P(F_j|C) = \frac{1 + TF(F_j, C)}{|T| + \sum_{i=1}^{n} TF(F_i, C)} \qquad (4.3)$$

Trong đó:

- F_j là từ thứ j trong tập từ khóa.
- $TF(F_j, C)$ là tần suất của từ F_j trong tài liệu Doc.
- $TF(F_j, C)$ là tần suất của từ F_j trong Catalog C.
- $|T|$ là số lượng các từ có trong tập từ khóa T.
- $P(F_j, C)$ là xác suất có điều kiện để từ F_j có mặt trong tài liệu của Catalog C.
- n là số lượng Catalog có trong hệ thống.

Trong công thức (4.3) xác suất $P(F_j/C)$ được tính sử dụng ước lượng xác suất Laplace. Để tránh trường hợp tần suất của từ F_j trong Catalog C bằng 0 - tức là từ F_j không có trong Catalog C thì tử số được cộng thêm 1.

Tuy nhiên có một điều quan trọng để làm giảm sự phức tạp trong tính toán và làm giảm bớt thời gian tính toán trong công thức (4.2) ta để ý thấy rằng: Không phải tài liệu Doc đã cho đều chứa tất cả các từ trong tập từ khóa T. Do đó, $TF(F_j , Doc) = 0$ khi từ khóa F_j thuộc T nhưng không thuộc tài liệu Doc. Và kết quả là kéo theo là $P(F_j|C)^{TF(F_j,Doc)}=1$ tại từ khóa F_j đó. Và như vậy ta có thể bỏ qua từ khóa F_j này mà không làm ảnh hưởng đến công thức (4.2). Và công thức cuối cùng của công thức (4.2) được viết lại như sau:

$$P(C|Doc) = \frac{P(C) \times \prod_{F_j \in Doc} P(F_j|C)^{TF(F_j,Doc)}}{\sum_{i=1}^{n} P(C_i) \times \prod_{F_l \in Doc} P(F_l|C_i)^{TF(F_l,Doc)}} \qquad (4.2')$$

và tương tự (4.3) được viết lại như sau:

$$P(F_j|C) = \frac{1 + TF(F_j, C)}{|T| + \sum_{i=1}^{n} TF(F_i, C)} \qquad (4.3')$$

với Fj ∈Doc.

Như vậy, bộ phân lớp không phải duyệt toàn bộ tập từ khóa T mà chỉ phải duyệt trên Vector của tài liệu Doc.

Một điều đáng chú ý nữa là: Các giá trị P(Ci) và các ngưỡng $CtgTsh_i$ là các giá trị được xác định trước thông qua những phân tích từ thực tế. Việc xác định các tham số này càng chính xác thì càng làm tăng độ tin cậy của bộ phân lớp.

Để hiểu rõ hơn sự hoạt động của bộ phân lớp, ta xem xét ví dụ 4.1 sau đây.

Ví dụ 4.1

Giả sử có hai Catalog C1 và C2 có các tham số như sau:

Tham số	C1	C2
P(C)	0.5	0.5
Ngưỡng	0.75	0.6

Số lượng các từ khóa trong tập từ khóa T (tức | T |) là 75.

Hệ thống từ khóa riêng của các Catalog (tức là các từ xuất hiện của các từ trong các Catalog) C1 và C2 là như sau:

Catalog C1		Catalog C2	
Từ khóa	Tần suất	Từ khóa	Tần suất
Xã hội	10	Xã hội	15
Chủ nghĩa	20	Tư bản	30
Cộng hoà	15		
Việt Nam	30		

Tài liệu Doc có nội dung: "Xã hội chủ nghĩa".

Vector của tài liệu này là: ((Xã hội,1), (Chủ nghĩa, 1));

Như vậy ta có các giá trị tính toán như sau:

Với Catalog C1:

P(Xã hội | C1)= 11/110;

P(Chủ nghĩa | C1)= 21/110;

Với Catalog C2:

P(Xã hội | C2)= 16/90;

P(Chủ nghĩa | C2)= 1/90;

$$\sum_{i=1}^{n} P(C_i) \times \prod_{F_l \in Doc} P\left(F_l \middle| C_i\right)^{TF(F_l, Doc)} = 0.6464;$$

P(C1 | Doc) = 0.914;

P(C2 | Doc) = 0.156;

Như vậy P(C1 | Doc)= 0.914 > 0.75; do đó nó được phân vào trong Catalog C1.

Còn P(C2 | Doc) = 0.156 < 0.6 do đó nó không được phân vào trong Catalog C2.

Bộ phân lớp đã phân tài liệu Doc đã cho vào trong catalog C1 với độ chính xác là 0.914%. Như vậy các tài liệu tuy được phân vào cùng một catalog nhưng

có thể có các giá trị P(C |Doc) hoàn toàn khác nhau, giá trị này của chúng càng cao thì độ chính xác của nó càng cao.

IV.2.3. Thuật toán phân lớp Bayes thứ hai

Bộ phân lớp thứ hai cũng có quá trình hoạt động hoàn toàn giống bộ phân lớp Bayes thứ nhất. Tuy nhiên, cách biểu diễn tài liệu Doc và các tính giá trị P(C | Doc) là khác với bộ phân lớp thứ nhất. Các thông tin sau cũng được đòi hỏi:

- Tập từ khóa T – Có ý nghĩa giống như bộ phân lớp Bayes thứ nhất.
- Xác suất P(Ci) – Có ý nghĩa giống như trên.
- Các ngưỡng $CtgTsh_i$ - Có ý nghĩa giống như trên.

Doc cũng biểu diễn dưới dạng một vector, nhưng kích thước của vector này bằng kích thước của tập từ khóa T. Mỗi thành phần của vector sẽ gồm từ khóa Fi và giá trị 1 hoặc 0 thể hiện từ khóa Fi đó xuất hiện hay không xuất hiện trong tài liệu Doc. Điều này có nghĩa là để tính toán giá trị P(C | Doc) thì bộ phân lớp này phải hoạt động trên toàn bộ tập từ khóa T.

Công thức tính toán giá trị P(C | Doc) được mô tả dưới đây:

$$P(C|Doc) = \frac{P(C) \times \prod_{F_j \in T} P\left(Doc(F_j)|C\right)}{\sum_{i=1}^{n} P(C_i) \times \prod_{F_l \in T} P\left(Doc(F_l)|C_i\right)} \qquad (4.4)$$

và

$$P\left(Doc(F_j)|C\right) = \frac{1 + N\left(Doc(F_j)|C\right)}{2 + |Dc|} \qquad (4.5)$$

Trong đó:

- P(Doc(Fj) | C) là xác suất có điều kiện để từ khóa Fj trong lớp C có cùng giá trị như trong tài liệu Doc, và được tính toán sử dụng công thức ước lượng xác suất Laplace như trong công thức (4.5).

- N(Doc(Fj) | C) là số lượng các tài liệu thuộc catalog C có cùng giá trị của từ Fj với tài liệu Doc. Tức là, số lượng các tài liệu thuộc catalog C cùng có hoặc cùng không có từ khóa Fj.
- | Dc | là tổng số các tài liệu có trong catalog C.

Trong công thức (4.5) sở dĩ có số 1 ở trên tử số là để tránh trường hợp N(Doc(Fj)|C)=0, còn số 2 ở mẫu là hai trạng thái giá trị của từ (xuất hiện hoặc không xuất hiện trong tài liệu).

Quá trình còn lại hoàn toàn tương tự như bộ phân lớp Bayes thứ nhất. Tức là, so sánh P(C | Doc) với ngưỡng CtgTsh để quyết định tài liệu Doc có thuộc vào Catalog C hay không.

Ví dụ 4.2

Giả sử có 2 Catalog C1 và C2 như sau:

Catalog	C1	C2
Xác suất	0.5	0.5
Ngưỡng	0.7	0.6

Và Catalog C1 có các tài liệu:

1. Học tốt phải học và học.
2. Và học mãi mãi mãi.
3. Đã học phải học.

Còn Catalog C2 có các tài liệu:

1. Sống đẹp sống mãi.
2. Đẹp nhất là sống đẹp.
3. Cuộc sống là đẹp.

Tài liệu Doc cần phân lớp là: "Mãi phải học".

Tập từ khóa T là: [Học, Tốt, Phải, Và, Mãi, Đã, Sống, Đẹp, Là, Nhất].

Tính toán các giá trị cho Catalog C1 như sau:

Các giá trị của catalog C1	Các giá trị của catalog C2
$\| Dc_1 \|=3$	$\| Dc_2 \|=3$

N(Học | C1)= 3; (Số các tài liệu trong Catalog C1 có chứa từ Học- giống như trong tài liệu Doc).	N(Học | C2)= 0; (Số các tài liệu trong Catalog C2 có chứa từ Học- giống như trong tài liệu Doc).
N(Và | C1)=1; (Số các tài liệu trong Catalog C1 không chứa từ Và- giống như trong tài liệu Doc).	N(Và | C2)= 3; (Số các tài liệu trong Catalog C2 không chứa từ Và- giống như trong tài liệu Doc).
N(Phải | C1)= 2	N(Phải | C2)= 0
N(Tốt | C1)= 2	N(Tốt | C2)= 3
N(Mãi | C1)=1	N(Mãi | C2)=1
N(Đã | C1)=2	N(Đã | C2)=3
N(Sống | C1)=3	N(Sống | C2)=0
N(Đẹp | C1)=3	N(Đẹp | C2)=0
N(Là | C1)=3	N(Là | C2)=1
N(Nhất | C1)=3	N(Nhất | C1)=2

$$P(C_1)\times \prod_{F_j\in T} P\left(Doc(F_j)\middle|C_1\right)= 55296/5^{10};$$

$$P(C_2)\times \prod_{F_j\in T} P\left(Doc(F_j)\middle|C_2\right)=384/5^{10};$$

$$\sum_{i=1}^{n} P(C_i)\times \prod_{F_l\in T} P\left(Doc(F_l)\middle|C_i\right)=55680/5^{10};$$

Vậy P(C1 | Doc)=0.993;

P(C2 | Doc)=0.016;

Do P(C1 | Doc)= 0.993 > 0.7 nên tài liệu Doc được phân vào trong Catalog C1; P(C2 | Doc)= 0.016 < 0.6 nên tài liệu Doc không được phân vào trong Catalog C2.

Việc chọn ngưỡng phân lớp

Như đã trình bày trong chương 1, cách chọn các ngưỡng $CtgTsh_i$ là phải phù hợp. Về mặt lí thuyết thì chọn ngưỡng $CtgTsh_i$ càng cao thì khi phân tài liệu

vào lớp C đòi hỏi phải có xác suất P(C|Doc) lớn hơn ngưỡng và khi đó xác suất đó phải cao và vì vậy, độ chính xác cũng cao theo. Tuy nhiên, nếu chọn các ngưỡng quá cao, thì có thể xảy ra trường hợp mọi giá trị P(Ci | Doc) đều không vượt qua được các ngưỡng $CtgTsh_i$. Điều đó có nghĩa rằng tài liệu Doc không được phân vào Catalog nào cả.

Ví dụ 4.3

Giả sử có 3 Catalog C1, C2 và C3 lần lượt có các ngưỡng là 0.7; 0.6; 0.5. Một tài liệu Doc sau khi được tính toán cho kết quả như sau:

P(C1 | Doc)=0.6; P(C2 | Doc)=0.3; P(C3 | Doc)=0.1; Khi đó tài liệu Doc sẽ không được phân vào Catalog nào trong 3 Catalog C1, C2, C3.

Ngược lại, nếu chọn ngưỡng nhỏ quá thì dẫn đến tình huống sau:

- Thứ nhất là về độ chính xác của tài liệu được phân là nhỏ.
- Thứ hai có khả năng xảy ra một tài liệu có thể được phân vào nhiều nhóm cùng một lúc.

Ví dụ 4.4

Có 3 Catalog như trên nhưng có các ngưỡng là 0.4; 0.5; 0.3; Và có P(C1 | Doc)=0.5; P(C2 | Doc)=0.1; P(C3 | Doc)=0.4;

Như vậy, tài liệu cùng một lúc được phân vào trong hai Catalog C1 và C3.

Để chọn được ngưỡng phù hợp thì cách tốt nhất là sự kết hợp giữa người và máy. Điều này được thực hiện bằng cách ta lấy một tài liệu đã biết trước được nội dung của nó nằm ở trong Catalog nào. Sau đó cho máy tự tìm ra các P(Ci | Doc) và phân các tài liệu đó vào các Catalog dựa trên các ngưỡng cũ đã có. Nếu ta thấy chúng được phân đúng theo như kết quả như đã biết trước thì giữ nguyên lại các ngưỡng cũ. Ngược lại nếu hệ thống phân không đúng với dự kiến thì tuỳ theo tình huống cụ thể mà có thể tăng hoặc giảm các ngưỡng cũ nếu thấy cần thiết.

IV.2.4. Thuật toán phân lớp "k_người láng giềng gần nhất"

Như đã trình bày trong chương 1, đối với cơ sở dữ liệu full-text theo mô hình vector trên đây, sự hoạt động của thuật toán không phụ thuộc vào tập từ khóa. Nói cách khác, nó không dựa vào tập từ khóa. Tuy nhiên, thuật toán vẫn sử dụng các ngưỡng $Ctgtsh_i$, và nó cũng tuần tự đi theo các bước như đã nói ở trên. Sự hoạt động của nó là dựa vào k tài liệu (lấy ngẫu nhiên) trong hệ thống, và tính P(C/Doc) dựa vào sự giống nhau của tài liệu Doc đã cho với k tài liệu được chọn. Trong thuật ngữ "k người láng giềng gần nhất", chính là chỉ k tài liệu được chọn. Cụ thể công thức tính P(C/ Doc) như sau:

$$P(C|Doc) = \frac{\sum_{l=1}^{k} Sm(Doc, D_l) \times P(C|D_l)}{\sum_{i=1}^{n}\sum_{l=1}^{k} Sm(Doc, D_l) \times P(C_i|D_l)} \qquad (4.6)$$

Trong đó:

- k là số lượng tài liệu được chọn để so sánh
- n là số catalog
- P (C_i | D_l) có giá trị 1 hoặc 0, chỉ ra tài liệu D_l có thuộc vào catalog C_i hay không, sở dĩ có giá trị này là bởi tại một tài liệu có thể được phân vào nhiều hơn một catalog .
- Sm (Doc, D_l) xác định mức độ giống nhau của tài liệu đã cho Doc với tài liệu được chọn D_l. Nó được tính bằng cos của góc giữa hai vectơ biểu diễn tài liệu Doc và tài liệu D_l theo công thức sau đây:

$$Sm(Doc, D_l) = Cos(Doc, D_l) = \frac{\sum_i X_i * Y_i}{Sqrt(\sum_j X_j^2 \sum_l Y_l^2)} \qquad (4.7)$$

Trong đó các biểu diễn tài liệu hoàn toàn tương tự với cách biểu diễn của bộ phận lớp Bayes thứ nhất. Tức là nó gồm các từ khóa F_i và các tần số X_i tương ứng.

Trong công thức (4.7):

- X_i là tần suất của các từ trong tài liệu Doc.

- Y_i là tần suất của các từ trong tài liệu D_l.
- $\Sigma_i X_i * Y_i$ là tổng của các tích các tần suất của các từ giống nhau giữa hai tài liệu Doc và D_l.
- $\Sigma_j X^2_j$ là tổng bình phương tần suất các từ có trong tài liệu Doc.
- $\Sigma_l Y^2_l$ là tổng bình phương tần suất các từ có trong tài liệu D_l.
- $Sqrt(\Sigma_j X^2_j \Sigma_l Y^2_l)$ là căn bậc hai của $\Sigma_j X^2_j \Sigma_l Y^2_l$.

Chẳng hạn, tài liệu Doc là

((Hà Nội , 2) , (Việt nam , 3) , (cộng hoà ,1) , (chủ nghĩa,4))

Còn tài liệu D_l là:

((Cộng hoà ,2) , (Xã hội , 5) , (Chủ nghĩa , 3) , (Việt nam, 1))

Khi đó:

$Cos(Doc,D_l)=(2*1+3*4+3*1)/ sqrt((2^2+3^2+1^2+4^2)*(2^2+5^2+3^2+1^2)) =0.497$

Quá trình còn lại hoàn toàn tương tự như các bộ phận lớp ở trên. Tức là nó cũng so sánh giá trị P(C | Doc) với ngưỡng CtgTsh của Catalog C để xem nó có thuộc vào trong Catalog đó không. Ví dụ 4.5 như được trình bày dưới đây mô tả hoạt động của thuật toán.

Ví dụ 4.5

Giả sử có 2 catalog C1và C2 với các ngưỡng tương ứng là 0.67 và 0.6.

Tài liệu Doc cần được phân lớp là: "Chủ nghĩa xã hội".

Các tài liệu được chọn ra để so sánh là:

1. " Cộng hoà xã hội chủ nghĩa Việt Nam" thuộc catalog C1.
2. " Xã hội xã hội chủ nghĩa" thuộc C1.
3. " Chủ nghĩa đế quốc, xã hội tư bản" thuộc catalog C2.

Quá trình tính toán diễn ra như sau:

Các vector biểu diễn các tài liệu là:

- D_1 = ((cộng hoà,1), (xã hội, 1), (chủ nghĩa, 1), (Việt Nam, 1)).
- D_2 = ((Xã hội, 2), (chủ nghĩa, 1)).

- D_3 = ((xã hội, 1), (chủ nghĩa, 1), (Đế quốc, 1), (tư bản, 1)).
- Doc = ((xã hội, 1), (chủ nghĩa, 1)).
- Cos (Doc, D_1)= 0.716;
- Cos (Doc, D_2) = 0.949;
- Cos (Doc, D_3) = 0.716;
- $\sum_{i=1}^{n}\sum_{l=1}^{k} Sm(Doc, D_l) \times P\left(C_i \middle| D_l\right)$ = 2.362;

Và:

- P(C_1 | Doc) = 0.70;
- P(C_2 | Doc) = 0.30;

Kết quả cuối cùng là: P(C_1 | Doc)=0.7 > 0.67 nên tài liệu Doc được phân vào trong Catalog C_1. Ngược lại, P(C_2 | Doc)=0.3 < 0.6 nên tài liệu Doc không được phân vào trong Catalog C_2.

Chú ý về nâng cao chất lượng thuật toán

Việc xác định k số lượng tài liệu mẫu và cách chọn những tài liệu mẫu nào để tính toán các khoảng cách nói trên có ý nghĩa quan trọng đối với chất lượng của thuật toán. Một trong những cách hiệu quả là dựa theo kinh nghiệm và sự kết hợp giữa người và máy.

PHẦN KẾT LUẬN

Luận văn đã xem xét một số nội dung trong mô hình học máy có giám sát. Học máy là một lĩnh vực được coi là liên quan mật thiết đến công nghệ tri thức. Tùy thuộc vào lượng thông tin đã có để phân loại học máy thành học máy không giám sát và học máy có giám sát. Bài toán học máy có giám sát đã có nhiều kết quả trong khi đó bài toán học máy không giám sát lại còn rất ít kết quả. Trong học máy có giám sát, đã có nhiều thuật toán giải quyết công việc phân lớp đối tượng trong đó điển hình nhất có thể kể đến các thuật toán Bayes, thuật toán k-người láng giềng gần nhất, thuật toán cây quyết định v.v. Trong mô hình học máy mô tả phức, mỗi khái niệm tương ứng một tập các luật và dữ liệu được xem xét không chỉ từng tập hợp dữ liệu đơn lẻ mà còn được xem xét theo nhiều tập hợp dữ liệu. Nhiều công trình nghiên cứu cho thấy, mô hình học máy mô tả phức cho kết quả học máy chính xác hơn so với mô hình đơn tương ứng. Bài toán học máy được gặp trong nhiều lĩnh vực khác nhau trong công nghệ tri thức và một số dạng của học máy cũng được tìm thấy trong cơ sở dữ liệu full-text. Bài toán phân lớp tài liệu trong cơ sở dữ liệu full-text là một bài toán khá phổ biến: Có thể sử dụng một số thuật toán học máy có giám sát theo mô hình vector của cơ sở dữ liệu full-text.

Luận văn đã thực hiện được một số nội dung chính như sau:

- Trình bày được một cách nhìn nhận tổng quan về bài toán học máy, phân loại bài toán học máy và một số thuật toán chính. Nội dung tổng quan này được tập hợp từ nhiều nguồn tài liệu khác nhau, ở trong nước cũng như ngoài nước.

- Trình bày được những nội dung cơ bản về học máy mô tả phức. Luận văn đã trình bày những nét cơ bản nhất về các mô hình học máy mô tả phức như FOIL, FOCL, HYDRA, HYDRA-MM. Kết quả của nội dung này được

tập hợp chủ yếu từ nhiều công trình nghiên cứu của nhóm học máy tại trường Đại học Tổng hợp California, Ivrin.

- Trình bày nội dung cơ bản về cơ sở dữ liệu full-text. Luận văn phát triển đề xuất cải tiến mô hình vector, bao gồm việc xem xét tần suất xuất hiện các từ khóa trong tài liệu cũng như vấn đề từ đồng nghĩa. Luận văn cũng trình bày một số thuật toán phân lớp tài liệu đối với cơ sở dữ liệu full-text. Một số kết quả cải tiến ở đây tuy có giá trị chưa cao song thực sự được phát triển bởi chính luận văn trên cơ sở của [5, 13].

Do còn có hạn chế về điều kiện, về khả năng triển khai trên máy tính nên luận văn còn có khiếm khuyết là chưa thể hiện được một cài đặt cụ thể cả về bài toán học máy mô tả phức lẫn bài toán tìm kiếm và phân lớp trong cơ sở dữ liệu full-text.

Các thuật toán học máy được gặp khá phổ biến trong các quá trình khám phá trí thức trong các cơ sở dữ liệu (KDD: Knowledge Discovery in Databases) và đây là lĩnh vực định hướng nghiên cứu tiếp của luận văn.

TÀI LIỆU THAM KHẢO

Tài liệu tiếng Việt

1. Hồ Tú Bảo. *Một số kết quả nghiên cứu về công nghệ tri thức*. Báo cáo Hội nghị Khoa học Viện Công nghệ Thông tin. Hà Nội 5&6-12-1996, trang 18-25.
2. Hồ Tú Bảo. *Học tự động không giám sát trên dàn Galois với dữ liệu thay đổi*. Báo cáo Hội nghị Khoa học Viện Công nghệ Thông tin. Hà Nội 5&6-12-1996, trang 27-36.
3. Hà Quang Thụy. *Tập thô trong bảng quyết định*. Tạp chí Khoa học Đại học Quốc gia Hà Nội. Tập 12. Số 4-1996, trang 9-14.
4. Nguyễn Thị Vân. *Xây dựng cơ sở dữ liệu Full-Text*. Luận văn tốt nghiệp Đại học, Khoa CNTT, 1998.

Tài liệu tiếng Anh

5. Ali K. & Pazzani M.. *Error Reduction through Learning Multiple Descriptions* Machine Learning, 24:3, 1996.
6. Ali K., Brunk C. & Pazzani M.. *Learning Multiple Relational Rule-based Models*. In "Preliminary Papers of the 5th International Workshop on Artificial Intelligence and Statistics". Fort Lauderdale, FL, 1995.
7. Ali K. & Pazzani M.. *HYDRA-MM: Learning Multiple Descriptions to Improve Classification Accuracy*. International Journal on Artificial Intelligence Tools, 4, 1995.
8. Ali K., Brunk C. & Pazzani M. *On Learning Multiple Descriptions of a Concept*. In Proceedings of the Sixth International Conference on Tools with Artificial Intelligence. New Orleans, LA: IEEE Press, 1994.

9. Bay S. D. *Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets*. Proceedings of the International Conference on Machine Learning. Morgan Kaufmann Publishers. Madison, Wisc., 1998.
10. Billsus D. & Pazzani M. *Learning probabilistic user models*. In workshop notes of Machine Learning for User Modeling, Sixth International Conference on User Modeling, Chia Laguna, Sardinia, 2-5 June 1997.
11. Bruce Moxon. *Defining Data mining*. DBMS Data Warehouse Supplement, August 1996.
12. Domingos P. *Knowledge Acquisition from Examples Via Multiple Models*. Proceedings of the Fourteenth International Conference on Machine Learning, 1997. Nashville, TN: Morgan Kaufmann.
13. Dunja Mladenic'. *Machine Learning on non-homogeneous, distbuted text data* (*Chapter 3. Document representation and learning algorithms*). Doctoral dissertation. University of Ljubljana, Slovenia. 1998.
14. Hume T. & Pzzani M. *Learning Sets of Related Concepts: A Shared Task Model*. Proceedings of the Sixteen Annual Conference of the Cognitive Science Society. Pittsburgh, PA: Lawrence Erlbaum, 1995.
15. Merz C. & Pazzani M. *Handling Redundancy in Ensembles of Learned Models Using Principal Components*. AAAI Workshop on Integrating Multiple Models, 1997.
16. Pazzani M. & Billsus D. *Learning and Revising User Profiles: The identification of interesting web sites*. Machine Learning 27, 313-331, 1997.
17. Shankle W. S., Datta P., Pazzani M. & Michael D. *Improving dementia screening tests with machine learning methods*. Alzheimer's Research, June, 1996, vol. 2 no. 3.

19. Peter Cheeseman, John Stutz. *Bayesian Classification (AutoClass): Theory and Results*. Advances in Knowledge Discovery and Data Mining. AAAI Press / The MIT Press 1996. 153-180.
20. Pazzani M., Kibler D. *The Utility of Knowledge in Inductive Learning*. Machine Learning, 9 , 54-97, 1992.