

Thư Viện ĐHTL

006.3

NG-N

2016

NGUYỄN HÀ NAM - NGUYỄN TRÍ THÀNH
HÀ QUANG THỦY

GIÁO TRÌNH

KHAI PHÁ DỮ LIỆU



Thư Viện ĐHTL



GT/287380

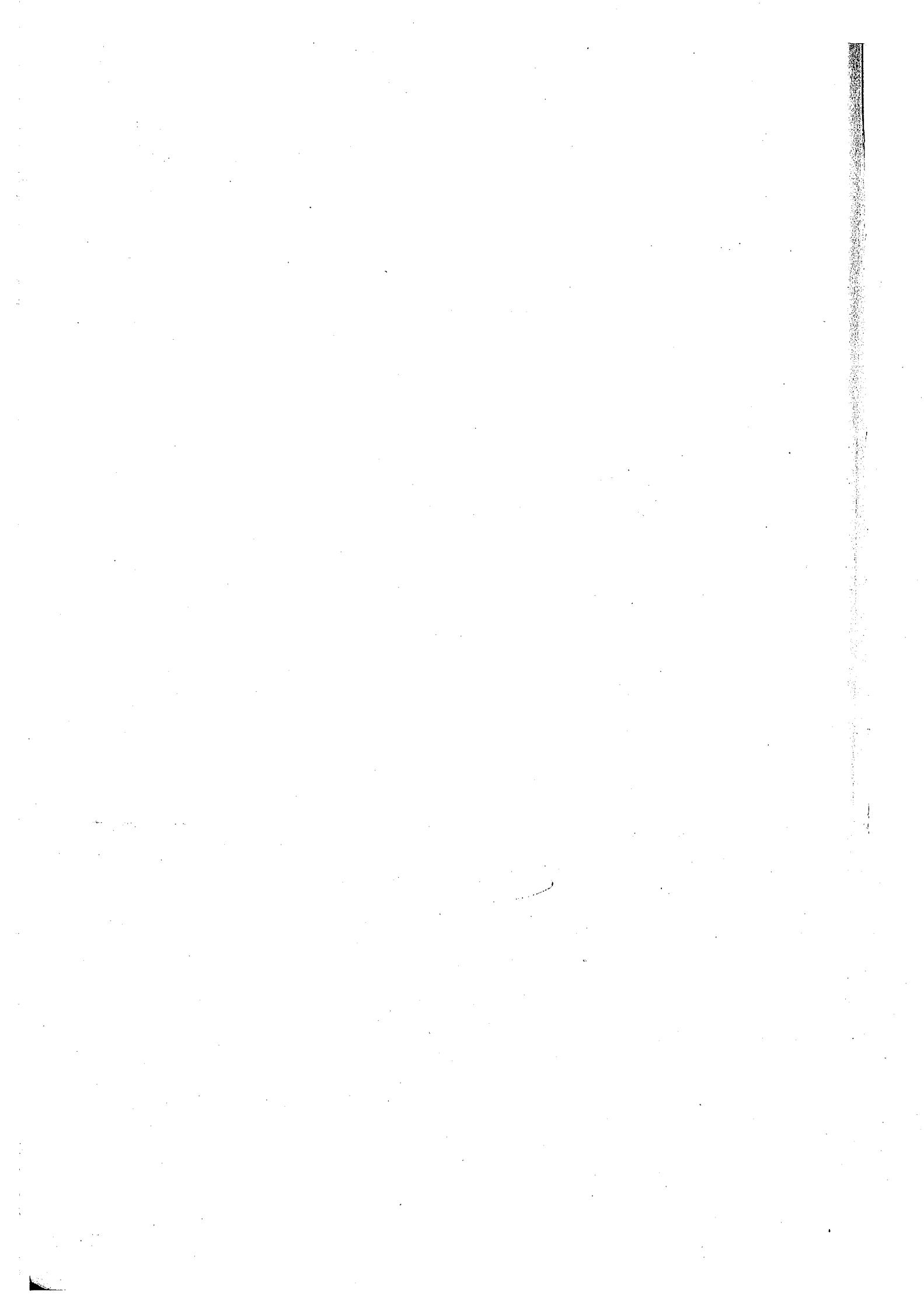
TỦ SÁCH KHOA HỌC
MS: 87-KHTN-2013



NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

GIÁO TRÌNH

KHAI PHÁ DỮ LIỆU



**NGUYỄN HÀ NAM, NGUYỄN TRÍ THÀNH,
HÀ QUANG THỦY**

GIÁO TRÌNH KHAI PHÁ DỮ LIỆU

NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC THỦY LỢI
THƯ VIỆN

Xin vui lòng không
tách rời chip bên dưới
nhãn này. Mức phí khi
làm hỏng hoặc mất
chip là 30.000đ/ chip.

3.4. Tích hợp dữ liệu	127
3.4.1. Nhận diện thực thể	128
3.4.2. Sự dư thừa và phân tích độ tương quan.....	129
3.4.3. Phát hiện các bộ lặp.....	133
3.4.4. Phát hiện xung đột trong dữ liệu và mức độ trừu tượng.	133
3.5. Chuyển đổi dữ liệu.....	134
3.5.1. Các chiến lược chuyển đổi dữ liệu	134
3.5.2. Chuẩn hóa dữ liệu.....	135
3.6. Phương pháp thu gọn dữ liệu.....	137
3.6.1. Giảm chiều dữ liệu	137
3.6.2. Giảm số lượng dữ liệu	140
3.7. Rời rạc hóa dữ liệu và sinh cây khái niệm phân cấp	144
3.7.1. Phương pháp áp dụng cho dữ liệu số.....	145
3.8. Phương pháp áp dụng cho dữ liệu phân loại.....	147
3.9. Tổng kết.....	147
Câu hỏi và bài tập	148

Chương 4. PHÁT HIỆN LUẬT KẾT HỢP	151
4.1. Giới thiệu về luật kết hợp.....	151
4.2. Phương pháp khai phá tập mục phổ biến.....	154
4.3. Thuật toán FP-Growth.....	156
4.3.1. Ý tưởng thuật toán.....	156
4.3.2. Thuật toán FP-growth.....	158
4.4. Một số thuật toán song song	164
4.4.1. Thuật toán phân phối độ hỗ trợ	164
4.4.2. Thuật toán phân phối dữ liệu.....	165
4.4.3. Thuật toán phân phối tập ứng cử viên	168
4.4.4. Thuật toán sinh luật song song.....	170
4.4.5. Một số thuật toán khác	172
4.5. Một số ứng dụng của luật kết hợp	172
Câu hỏi và bài tập	173

Chương 5. PHÂN CỤM DỮ LIỆU.....	175
5.1. Giới thiệu	175
5.1.1. Bài toán phân cụm	175
5.1.2. Các phương pháp phân cụm	176
5.2. Một số độ đo cơ bản dùng trong phân cụm	180
5.2.1. Độ đo tương đồng.....	180
5.2.2. Độ đo khác biệt	181
5.3. Thuật toán phân cụm phẳng	184
5.3.1. Thuật toán k-means	184
5.3.2. Thuật toán k-medoids.....	187
5.3.3. Tìm số lượng cụm thích hợp.....	189
5.4. Thuật toán phân cụm phân cấp.....	191
5.4.1. Phân cụm phân cấp gộp.....	191
5.4.2. Các thuật toán phân cụm phân cấp BIRCH.....	196
5.4.3. Thuật toán phân cụm phân cấp từ trên xuống DIANA.....	201
5.4.4. Thuật toán phân cụm phân cấp ROCK.....	203
5.5. Thuật toán phân cụm dựa trên mật độ.....	208
5.6. Giải thuật phân cụm dựa trên mô hình	211
5.7. Nhận xét sơ bộ các thuật toán phân cụm.....	215
5.8. Đánh giá các giải thuật phân cụm.....	217
5.8.1. Đánh giá dựa trên độ tương tự.....	217
5.8.2. Đánh giá dựa trên dữ liệu gán nhãn.....	218
5.9. Một số ứng dụng của phân cụm	223
Câu hỏi và bài tập	225
Chương 6. PHÂN LỚP DỮ LIỆU	227
6.1. Giới thiệu	227
6.2. Phân lớp bằng cây quyết định	230
6.2.1. Độ lợi thông tin	234
6.2.2. Tỉ số độ lợi	236

6.2.3. Chỉ số Gini	237
6.2.4. Tỉa cây quyết định.....	240
6.3. Thuật toán phân lớp Naive Bayes	240
6.3.1. Định lý Bayes	240
6.3.2. Phân lớp Naive Bayes.....	241
6.4. Thuật toán phân lớp máy vector hỗ trợ SVM	245
6.4.1. Trường hợp dữ liệu có thể phân loại tuyến tính.....	245
6.4.2. Trường hợp dữ liệu không thể phân tách tuyến tính.....	249
6.4.3. Phân lớp đa lớp với SVM.....	253
6.5. Thuật toán phân lớp kNN	254
6.6. Đánh giá các giải thuật phân lớp	258
6.7. Một số ứng dụng của các giải thuật phân lớp.....	261
Câu hỏi và bài tập	262

<i>Chương 7. PHƯƠNG PHÁP HỌC BÁN GIÁM SÁT</i>	265
7.1. Giới thiệu	265
7.2. Thuật toán cực đại kỳ vọng EM	268
7.3. Thuật toán học cộng tác (co-training).....	273
7.3.1. Thuật toán học cộng tác dựa trên nhiều khung nhìn	273
7.3.2. Thuật toán học cộng tác co-EM	277
7.3.3. Thuật toán học cộng tác dựa trên nhiều giải thuật học giám sát	278
7.4. Thuật toán Tri-training	280
7.5. Thuật toán tự huấn luyện (Shelf-training)	283
7.6. Một số ứng dụng của các giải thuật học bán giám sát.....	285
Câu hỏi và bài tập	285

<i>Chương 8. KHAI PHÁ DỮ LIỆU BẢO VỆ TÍNH RIÊNG TỰ.....</i>	287
8.1. Khía cạnh pháp luật bảo vệ tính riêng tư và khai phá dữ liệu	288
8.1.1. Hướng dẫn của OECD về dữ liệu riêng tư và tác động tới hoạt động phát hiện tri thức từ dữ liệu.....	288
8.1.2. Tiếp cận pháp luật bảo vệ tính riêng tư tại nước Mỹ và tác động tới khai phá dữ liệu.....	291
8.2. Phương pháp khai phá dữ liệu bảo vệ tính riêng tư.....	293
8.2.1. Mô hình và phương pháp khai phá dữ liệu bảo vệ tính riêng tư	293

8.2.2. Một số thuật toán khai phá dữ liệu bảo vệ tính riêng tư	296
Câu hỏi và bài tập	304
Chương 9. TẬP MỜ, TẬP THÔ VÀ TẬP MỜ – THÔ	
TRONG KHAI PHÁ DỮ LIỆU	305
9.1. Phương pháp tập mờ trong khai phá dữ liệu.....	305
9.1.1. Một số kiến thức cơ sở của lý thuyết tập mờ.....	306
9.1.2. Phương pháp tập mờ trong khai phá dữ liệu	312
9.2. Phương pháp tập thô trong khai phá dữ liệu	319
9.2.1. Một số kiến thức cơ sở về lý thuyết tập thô.....	320
9.2.2. Phương pháp tập thô rút gọn thuộc tính	326
9.2.3. Phương pháp tập thô rời rạc tập giá trị thuộc tính	330
9.3. Phương pháp tập mờ-thô trong khai phá dữ liệu.....	333
9.3.1. Lựa chọn thuộc tính dựa trên tập mờ - thô	334
9.3.2. Phân lớp k-NN dựa trên tập mờ - thô	335
Câu hỏi và bài tập	335
Chương 10. MỘT SỐ BÀI HỌC VÀ KHUYNH HƯỚNG PHÁT TRIỂN	
TRONG KHAI PHÁ DỮ LIỆU	337
10.1. Một số bài học trong khai phá dữ liệu	338
10.1.1. Bài học về kỹ thuật.....	338
10.1.2. Bài học về triển khai dự án	344
10.1.3. Đặc trưng của chuyên viên khai phá dữ liệu	345
10.2. Một số lỗi thường gặp trong khai phá dữ liệu	347
10.3. Công cụ khai phá dữ liệu	356
10.3.1. Tiêu chí phân loại các công cụ khai phá dữ liệu	357
10.3.2. Các kiểu công cụ khai phá dữ liệu.....	360
10.3.3. Tập ví dụ đánh giá công cụ nghiên cứu.....	365
10.4. Khuynh hướng phát triển của khai phá dữ liệu	366
10.4.1. Khuynh hướng phát triển của khoa học máy tính.....	367
10.4.2. Khuynh hướng phát triển của khai phá dữ liệu.....	368
Câu hỏi và bài tập	379
Tài liệu tham khảo.....	381

MỤC LỤC

Lời giới thiệu	9
Chương 1. GIỚI THIỆU CHUNG VỀ KHAI PHÁ DỮ LIỆU.....	13
1.1. Nhu cầu phát hiện tri thức từ dữ liệu.....	14
1.1.1. Tình trạng “bung nổ dữ liệu”	14
1.1.2. Ngành công nghiệp dựa trên dữ liệu	21
1.2. Khái niệm phát hiện tri thức trong cơ sở dữ liệu	25
1.2.1. Giải thích một số thuật ngữ.....	28
1.2.2. Quá trình phát hiện tri thức trong cơ sở dữ liệu.....	34
1.2.3. Bước khai phá dữ liệu trong quá trình phát hiện tri thức từ dữ liệu... ..	37
1.2.4. Kiến trúc một hệ thống khai phá dữ liệu	37
1.3. Khai phá dữ liệu và xử lý CSDL truyền thống.....	39
1.4. Một số lĩnh vực ứng dụng khai phá dữ liệu điển hình	42
1.5. Kiểu dữ liệu trong khai phá dữ liệu	45
1.5.1. Cơ sở dữ liệu quan hệ	45
1.5.2. Kho dữ liệu.....	46
1.5.3. Cơ sở dữ liệu giao dịch.....	47
1.5.4. Các hệ thống dữ liệu mở rộng	47
1.6. Các bài toán khai phá dữ liệu điển hình	48
1.6.1. Mô tả khái niệm	50
1.6.2. Quan hệ kết hợp	50
1.6.3. Phân lớp.....	51
1.6.4. Phân cụm.....	52
1.6.5. Hồi quy.....	52
1.6.6. Mô hình phụ thuộc	53

1.6.7. Phát hiện biến đổi và độ lệch	53
1.7. Tính liên ngành của khai phá dữ liệu	54
Câu hỏi và bài tập	59
 Chương 2. CÔNG NGHỆ TRI THỨC VÀ PHÁT HIỆN TRI THỨC TỪ DỮ LIỆU ... 61	
2.1. Vai trò của CNTT trong kinh tế tri thức	61
2.1.1. Nghịch lý hiệu quả của CNTT của Robert Solow và luận điểm của N. Carr ..	62
2.1.2. Vai trò của CNTT trong nền kinh tế tri thức	67
2.1.2. Vai trò của giám đốc thông tin trong doanh nghiệp và tổ chức.....	71
2.2. Công nghệ tri thức	74
2.2.1. Khái niệm tri thức	75
2.2.2. Nguồn tri thức cho cá nhân và tổ chức	78
2.2.3. Công nghệ tri thức.....	82
2.3. Bài toán phát hiện tri thức từ dữ liệu	85
2.3.1. Sự tiến hóa của mô hình phát hiện tri thức	85
2.3.2 Về bài toán khai phá dữ liệu.....	97
2.4. Độ đo hấp dẫn trong khai phá dữ liệu	99
Câu hỏi và bài tập	106
 Chương 3. CHUẨN BỊ DỮ LIỆU 109	
3.1. Giới thiệu	109
3.2. Hiểu dữ liệu	110
3.2.1. Đo độ tập trung của dữ liệu	110
3.2.2. Đo độ phân tán của dữ liệu.....	112
3.2.3. Hiển thị dữ liệu tóm tắt	114
3.3. Tiền xử lý dữ liệu	118
3.4. Làm sạch dữ liệu	120
3.4.1. Các giá trị bị thiếu	121
3.4.2. Dữ liệu bị nhiễu.....	123
3.4.3. Làm sạch dữ liệu phải là một quy trình.....	125

LỜI GIỚI THIỆU

Trong thời đại ngày nay, sử dụng tri thức đã trở thành động lực chủ chốt cho tăng trưởng kinh tế quốc gia, cho tăng cường năng lực cạnh tranh của doanh nghiệp. Đồng thời, dung lượng dữ liệu số tăng rất nhanh chóng, đặc biệt loại dữ liệu do người sử dụng tạo ra (User-Generated Content: UGC) chiếm tỷ trọng ngày càng cao, đã trở thành nguồn tài nguyên tiềm ẩn thông tin và tri thức có tiềm năng lớn hữu ích cho phát triển kinh tế và tăng cường năng lực cạnh tranh. Nghiên cứu và triển khai các phương pháp tự động phát hiện các mẫu mới, có giá trị, hữu ích tiềm năng và hiểu được trong khối dữ liệu đồ sộ, khắc phục hiện tượng *giàu về dữ liệu mà nghèo về thông tin*, hướng tới mục tiêu tăng cường tài nguyên tri thức là hết sức cần thiết và có ý nghĩa. Khai phá dữ liệu (Data Mining) và Phát hiện tri thức trong cơ sở dữ liệu (Knowledge Discovery in Data Bases: KDD), thành phần quan trọng của công nghệ tri thức (Knowledge Technology), đang phát triển rất mạnh mẽ.

Khai phá dữ liệu là môn học bắt buộc trong chương trình đào tạo ngành Hệ thống thông tin (HTTT) bậc cử nhân và chuyên ngành HTTT bậc thạc sĩ tại Khoa CNTT, Trường Đại học Công nghệ (ĐHCN), Đại học Quốc gia Hà Nội (ĐHQGHN). Nhu cầu đào tạo, nghiên cứu và phát triển lĩnh vực khai phá dữ liệu trước hết tại Trường ĐHCN, và sau đó tại các cơ sở đào tạo và nghiên cứu trong nước đòi hỏi một giáo trình có nội dung toàn diện về lĩnh vực nghiên cứu và triển khai quan trọng này.

Trước khi giới thiệu nội dung của giáo trình này, chúng tôi muốn nêu lên một vài điểm về cách tiếp cận của chúng tôi. Thứ nhất, giáo trình được viết để phục vụ việc giảng dạy và học tập bậc đại học và bậc sau đại học tại Trường ĐHCN, ĐHQGHN. Nội dung

trong giáo trình được tổng hợp và tóm lược từ một số tài liệu nổi tiếng cũng như những nghiên cứu thời sự nhất về khai phá dữ liệu. Thứ hai, nội dung về kho dữ liệu được viết thành giáo trình "Kho dữ liệu" cho nên sẽ không được đưa vào giáo trình này. Thứ ba, giáo trình này còn có mục tiêu định hướng cho các nghiên cứu chuyên sâu về khai phá dữ liệu, vì vậy, giáo trình bổ sung thêm một số nội dung khác với nhiều cuốn sách hiện có về khai phá dữ liệu. Nội dung đầu tiên được bổ sung là một số kiến thức về tri thức và kinh tế tri thức. Thêm nữa, chúng tôi bổ sung một số nội dung về khai phá dữ liệu dựa trên lý thuyết tập mờ, lý thuyết tập thô và một số bài học thành công cũng như một số lỗi thường gặp trong khai phá dữ liệu. Khuynh hướng nghiên cứu và triển khai khai phá dữ liệu được trình bày với các nội dung cập nhật nhất có thể được.

Giáo trình gồm 10 chương với nội dung sơ bộ như được trình bày dưới đây.

Chương 1. Giới thiệu chung về khai phá dữ liệu trình bày về sự tăng trưởng mạnh mẽ về dung lượng dữ liệu (đặc biệt là dữ liệu nội dung do người dùng sinh ra: generated user content – GUC), về công nghệ dựa trên dữ liệu, về nhu cầu phát hiện tri thức từ dữ liệu, về các khái niệm cơ bản nhất của khai phá dữ liệu và phát hiện tri thức từ dữ liệu. Tính liên ngành của khai phá dữ liệu và sự phân biệt giữa hệ thống khai phá dữ liệu và hệ thống quản lý cơ sở dữ liệu, giữa bài toán khai phá dữ liệu và bài toán thống kê cũng được đề cập.

Chương 2. Công nghệ tri thức và phát hiện tri thức từ dữ liệu cung cấp những kiến thức cơ bản nhất về tri thức và kinh tế tri thức, vai trò của CNTT và công nghệ tri thức cho phát triển kinh tế và tạo lợi thế cạnh tranh. Quá trình tiến hóa của mô hình phát hiện tri thức từ dữ liệu được phân tích. Một số nội dung về độ đo hấp dẫn và tính hấp dẫn của mẫu được trình bày.

Chương 3. Chuẩn bị dữ liệu và kho dữ liệu cung cấp các kiến thức và kỹ năng về hiểu dữ liệu, tiền xử lý dữ liệu, chuyển dạng dữ liệu, lựa chọn thuộc tính.

Chương 4. Phát hiện luật kết hợp trình bày khái niệm luật kết hợp, một số thuật toán khai phá luật kết hợp điển hình (thuật toán Apriori, thuật toán FP-growth và và một số thuật toán khác), khái niệm luật dây và khai phá luật dây. Một số ứng dụng của luật kết hợp cũng được giới thiệu.

Chương 5. Phân cụm dữ liệu và mô tả cung cấp kiến thức về bài toán phân cụm và một số thuật toán phân cụm điển hình (phân cụm phân cấp, phân cụm phẳng K-mean, phân cụm EM, một số thuật toán khác). Phương pháp đánh giá phân cụm và một số ứng dụng phân cụm cũng được giới thiệu.

Chương 6. Phân lớp dữ liệu trình bày về khái niệm bài toán phân lớp, một số thuật toán phân lớp điển hình (C4.5, Naive Bayes, k-NN, SVM và một số thuật toán khác). Phương pháp đánh giá thuật toán phân lớp và một số ứng dụng thuật toán phân lớp cũng được giới thiệu.

Chương 7. Phương pháp học bán giám sát được bắt đầu bằng các nội dung cơ bản của phương pháp học bán giám sát. Một số thuật toán bán giám sát điển hình (Adaboost, Co-training, Shelf-training và một số thuật toán học bán giám sát khác) được trình bày chi tiết. Một số ứng dụng học bán giám sát cũng được giới thiệu.

Chương 8. Khai phá dữ liệu bảo vệ tính riêng tư cung cấp các kiến thức cơ bản về tính riêng tư, một số mô hình và giải pháp khai phá dữ liệu bảo vệ tính riêng tư.

Chương 9. Tập mờ, tập thô và tập mờ-thô trong khai phá dữ liệu trình bày một số kiến thức cơ bản về tập mờ, tập thô, tập mờ-thô và ứng dụng các tập nói trên trong khai phá dữ liệu.

Chương 10. Một số bài học và khuynh hướng phát triển của khai phá dữ liệu trình bày một số bài học và lỗi thường gặp trong khai phá dữ liệu. Phần cuối của chương đề cập tới khuynh hướng phát triển khai phá dữ liệu, tập trung vào, khai phá dữ liệu phương tiện xã hội, học máy hướng miền ứng dụng và học máy không dừng được chọn lựa để giới thiệu chi tiết hơn.

Giáo trình này được sử dụng cho cả bậc đại học và bậc cao học. Một phương án đề nghị cho đào tạo bậc đại học là gói nội dung bao gồm chương 1, chương 2 (không kể mục 2.4), chương 3 (không kể mục 3.4), chương 4, chương 5, chương 6, chương 10 (hai mục 10.1, 10.2). Ôn lại nội dung dành cho bậc đại học và nghiên cứu các nội dung còn lại trong giáo trình là phương án nội dung dạy-học cho bậc sau đại học.

Đối với lĩnh vực khai phá dữ liệu, việc dùng thuật ngữ tiếng Việt là rất khó khăn vì đây là lĩnh vực nghiên cứu còn rất mới không chỉ ở Việt Nam mà còn trên thế giới. Với mỗi thuật ngữ tiếng Anh, thuật ngữ tiếng Việt tương ứng được coi là phổ biến được chọn lựa.

Nhóm tác giả xin bày tỏ lời cảm ơn chân thành tới TS. Nguyễn Lê Minh, TS. Đoàn Sơn, TS. Phan Xuân Hiếu, TS. Nguyễn Cẩm Tú, TS. Nguyễn Việt Cường, TS. Đặng Thanh Hải đã nhiệt tình cộng tác. Nhóm tác giả đánh giá cao và chân thành cảm ơn tập thể cán bộ, sinh viên thuộc Phòng Thí nghiệm Công nghệ Tri thức và Bộ môn HTTT, Khoa CNTT. Giáo trình này cũng là một sản phẩm của quá trình cộng tác nghiên cứu của chúng tôi với Cố Giáo sư Susumu Horiguchi tại Viện Khoa học & Công nghệ tiên tiến và Đại học Tohoku Nhật Bản, GS. Akira Shimazu tại Viện Khoa học & Công nghệ tiên tiến Nhật Bản, TSKH Nguyễn Hùng Sơn tại Đại học Vasava Ba Lan.

Dù nhóm tác giả đã cố gắng thu thập, nghiên cứu và tổng hợp song giáo trình chắc chắn còn không ít khiếm khuyết. Chúng tôi mong muốn nhận được sự cảm thông cũng như các ý kiến đóng góp từ các nhà khoa học, các giảng viên và người học để giáo trình ngày càng hoàn thiện.

Nhóm tác giả xin chân thành cảm ơn các cơ quan hữu quan đã tích cực hỗ trợ để xuất bản giáo trình.

Các tác giả

Chương 1.

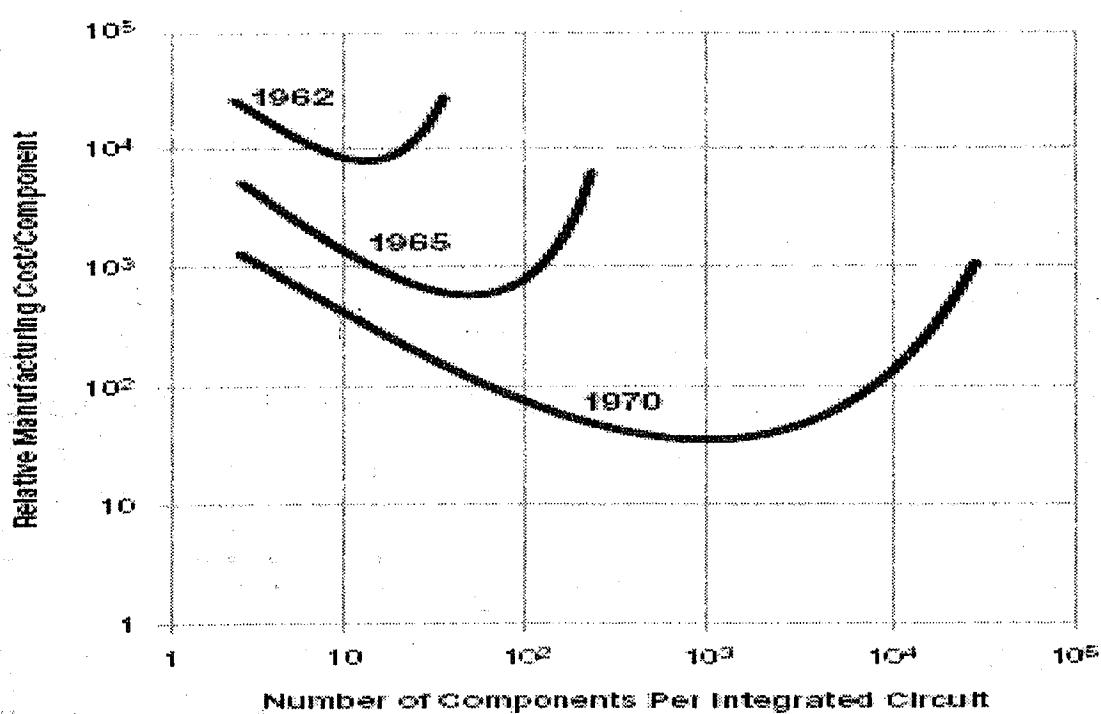
GIỚI THIỆU CHUNG VỀ KHAI PHÁ DỮ LIỆU

Chương mở đầu của giáo trình trình bày một số nét khái quát nhất về khai phá dữ liệu. Mục đầu tiên giới thiệu về tính tự nhiên của tình trạng bùng nổ dữ liệu và phát hiện tri thức từ dữ liệu như một thành phần nền tảng công nghệ của ngành kinh tế định hướng dữ liệu. Mục thứ hai giới thiệu khái niệm phát hiện tri thức trong cơ sở dữ liệu, khái niệm khai phá dữ liệu. Phát hiện tri thức trong cơ sở dữ liệu là một quá trình gồm nhiều bước tìm ra những mẫu có giá trị, mới, hữu ích tiềm năng và hiểu được trong một tập dữ liệu lớn. Khai phá dữ liệu là bước xử lý đặc thù nhất của quá trình này, vì vậy, trong không ít trường hợp hai khái niệm phát hiện tri thức từ dữ liệu và khai phá dữ liệu được dùng thay thế nhau. Mục tiếp theo trình bày một số khía cạnh phân biệt hệ thống khai phá dữ liệu (cung cấp thông tin hỗ trợ quyết định) với hệ thống cơ sở dữ liệu điều hành tác nghiệp truyền thống (phục vụ xử lý giao dịch tác nghiệp). Mục thứ tư giới thiệu một số lĩnh vực ứng dụng khai phá điển hình, trong đó kinh doanh là một trong những lĩnh vực ứng dụng phổ biến nhất. Mục thứ năm cung cấp một số thông tin cho biết tính đa dạng của kiểu dữ liệu đầu vào của bài toán khai phá dữ liệu. Mục thứ sáu giới thiệu các bài toán khai phá dữ liệu điển hình thuộc vào hai lớp bài toán dự báo và mô tả. Mục cuối cùng của chương này trình bày tính đa ngành của lĩnh vực khai phá dữ liệu.

1.1. NHU CẦU PHÁT HIỆN TRI THỨC TỪ DỮ LIỆU

1.1.1. Tình trạng “bùng nổ dữ liệu”

Thời đại ngày nay, mỗi chúng ta đã từng nghe nói và chứng kiến về sự tăng trưởng liên tục với tốc độ vượt bậc về dung lượng dữ liệu do con người khởi tạo, lưu giữ và truyền dẫn; sự tăng trưởng này còn được gọi là “hiện tượng bùng nổ thông tin”. Trước khi xem xét mối liên hệ giữa hiện tượng bùng nổ thông tin với nhu cầu khai phá dữ liệu và phát hiện tri thức từ dữ liệu, chúng ta tìm hiểu về các nguyên nhân tạo nên hiện tượng bùng nổ thông tin đó. Nói một cách khái quát, hiện tượng bùng nổ thông tin có nguyên nhân từ nhu cầu hoạt động mọi mặt của đời sống xã hội, tuy nhiên, những nội dung trình bày dưới đây sẽ làm chi tiết hơn về các khía cạnh công nghệ và xã hội đã góp phần thúc đẩy sự tăng trưởng dữ liệu vượt bậc đó.



Hình 1.1. Xu thế tối ưu chi phí sản xuất mạch bán dẫn: Số lượng thành phần bán dẫn trong một mạch tích hợp tăng và chi phí sản xuất một thành phần bán dẫn giảm (G.E. Moore, 1965 [Moore65]).

1.1.1.1. Về mặt công nghệ

Bảng 1.1. Tổng giao vận IP năm 2009 và dự báo các năm 2010-2014.
 Chú thích: Consumer: Lưu lượng IP cố định do hộ gia đình, cư dân trường đại học, và cà phê Internet tạo ra; Business: Lưu lượng IP hoặc WAN cố định (không bao gồm lưu lượng sao lưu) do doanh nghiệp và chính quyền tạo ra; Mobility: Lưu lượng dữ liệu di động và truy cập Internet từ thiết bị cầm tay, thẻ máy tính xách tay, WiMAX; Internet: toàn bộ lưu lượng IP đi qua đường trực Internet. Nguồn: Sách trắng CISCO 2010

IP Traffic - 2009 - 2014							
	2009	2010	2011	2012	2013	2014	CAGR 2009-2014
By Type (PB per Month)							
Internet	10,942	15,205	21,181	26,232	36,709	47,176	34%
Managed IP	3,652	4,963	6,771	8,651	11,078	13,199	29%
Mobile Data	81	228	538	1,158	2,132	3,528	108%
By Segment (PB per Month)							
Consumer	11,602	16,534	23,750	32,545	43,117	55,801	37%
Business	3,083	3,862	4,740	5,697	6,601	8,103	21%
By Geography (PB per Month)							
North America	5,115	7,091	10,051	12,988	16,136	19,019	30%
Western Europe	3,495	4,818	6,712	9,261	12,417	16,156	36%
Asia Pacific	3,920	5,367	7,295	9,815	12,985	17,421	35%
Japan	1,068	1,539	2,149	2,855	3,591	4,300	32%
Latin America	438	680	1,026	1,527	2,274	3,479	51%
Central Eastern Europe	493	678	938	1,306	1,815	2,510	38%
Middle East and Africa	157	223	319	490	700	1,018	45%
Total (PB per Month)							
Total IP Traffic	14,686	20,396	28,491	38,242	49,919	63,904	34%

Source: Cisco VNI, 2010

Sự tăng trưởng dữ liệu với tốc độ cao như được đề cập được dẫn xuất từ các nguyên nhân công nghệ sau đây:

+ Công nghệ chế tạo các thiết bị xử lý, lưu giữ và truyền dẫn dữ liệu đã và đang phát triển không ngừng, tạo ra các sản phẩm thiết bị có tốc độ hoạt động ngày càng cao và giá thành ngày càng hạ. Sự phát triển công nghệ này được dẫn dắt bởi định luật Moore, một định luật có xuất phát điểm từ nội dung một bài báo được Gordon E. Moore, một đồng sáng lập công ty Intel (INTEGRATED Electronics) công bố vào năm 1965 [Moore65]. Nội dung được coi là quan trọng nhất trong bài báo này của G. E. Moore là dự báo về xu thế tăng số lượng thành phần bán dẫn để đạt được chi phí sản xuất hiệu quả nhất (Hình 1.1).

Sau này, dự báo nói trên của G.E. Moore được phát biểu dưới dạng “phương ngôn 2x” như sau “Số lượng bán dẫn tích hợp trong

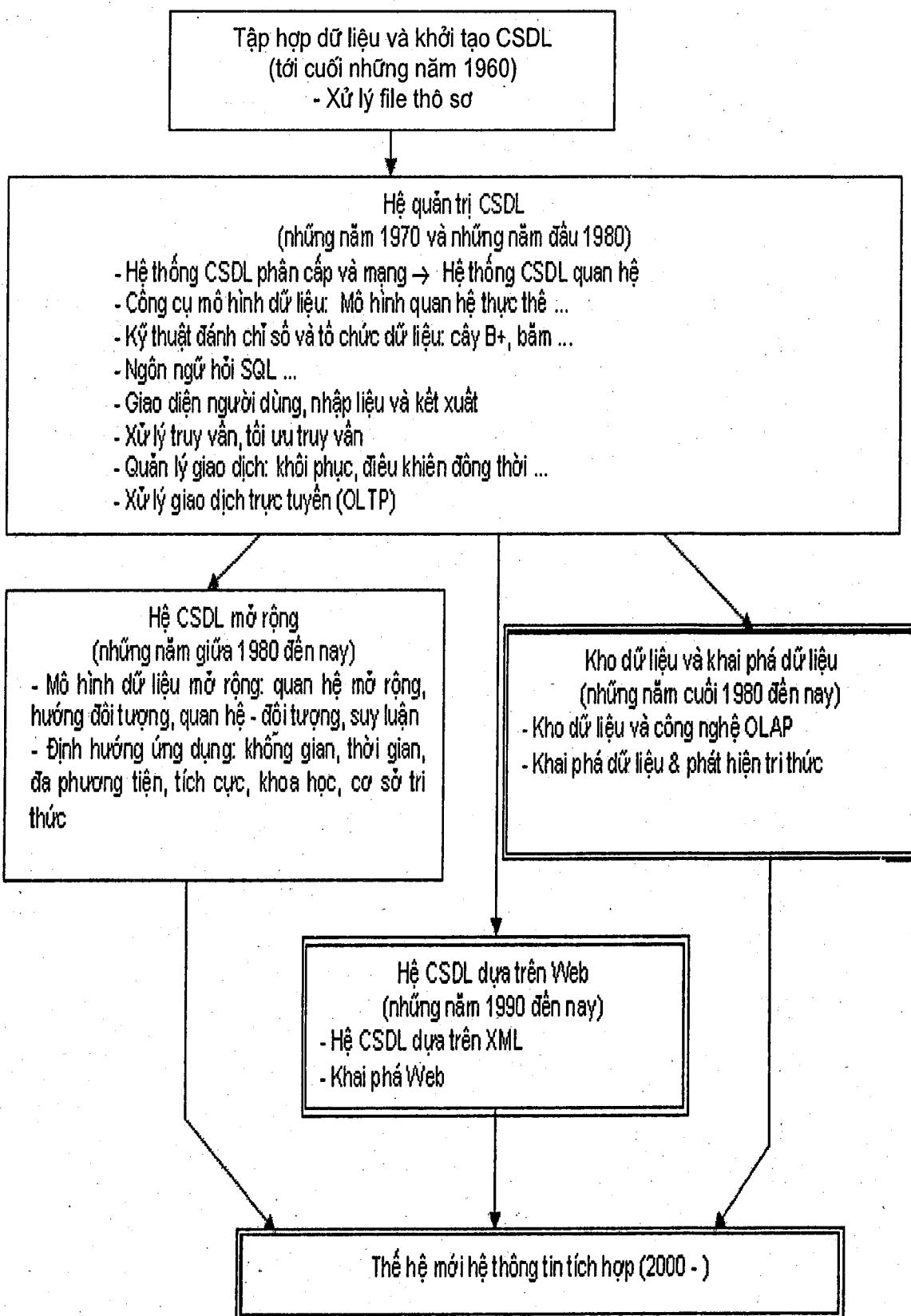
một chíp sẽ tăng gấp đôi sau một chu kỳ khoảng hai năm". Một dạng phát biểu khác của định luật Moore là "chi phí sản xuất mạch bán dẫn với cùng tính năng giảm một nửa sau khoảng hai năm". Phiên bản "18 tháng" của phương ngôn 2x rút ngắn chu kỳ thời gian từ hai năm xuống còn 18 tháng.

*Định luật Moore đã và đang dẫn dắt ngành công nghiệp mạch bán dẫn mà "về bản chất, nó là mô hình cơ bản cho ngành công nghiệp bán dẫn". Theo Paul S. Otellini, Chủ tịch và Giám đốc điều hành Tập đoàn Intel, thì "Định luật Moore vẫn tạo khả năng cơ bản cho sự phát triển của chúng tôi, và nó vẫn còn hiệu lực tốt tại Intel. Nhưng cách chúng tôi và khách hàng xem xét định luật Moore đã có sự thay đổi. Định luật Moore không chỉ là mạch bán dẫn. Nó cũng là cách sử dụng sáng tạo mạch bán dẫn". Theo Daniel Grupp, Giám đốc phát triển công nghệ tiên tiến của Acorn Technologies, Inc. (<http://acorntech.com/>) thì "tất cả các bước chu trình thiết kế, phát triển, sản xuất, phân phối và bán hàng được coi là có tính bền vững khi tuân theo định luật Moore. Nếu đánh bại định luật Moore, thị trường không thể hấp thụ hết các sản phẩm mới, và kỹ sư bị mất việc làm. Nếu bị tụt sau định luật Moore, không có gì để mua, và gánh nặng đè lên vai của chuỗi nhà phân phối sản phẩm"*¹.

Cuộc cách mạng trong công nghiệp mạch bán dẫn (nền tảng của công nghiệp điện tử) tác động mạnh mẽ đối với công nghiệp phần cứng máy tính, tạo ra sự bùng nổ về năng lực xử lý tính toán và dung lượng lưu trữ dữ liệu; kết quả là các thiết bị tạo lập và lưu trữ dữ liệu mang theo sự tiến bộ công nghệ không ngừng được sản xuất và đưa vào sử dụng.

¹ "Intel Silicon Innovation". http://download.intel.com/museum/Moores_Law/Printed_Materials/Intel_Silicon_Brochure.pdf

² <http://www.edavision.com/200.111/feature.pdf>

**Hình 1.2.** Tiến hóa của công nghệ cơ sở dữ liệu [HK0106]

Lịch sử phát triển các bộ xử lý Intel là một minh họa điển hình, thể hiện sự phát triển công nghệ bộ xử lý được dẫn dắt bởi định luật Moore³. Một ví dụ khác, hoạt động thu thập dữ liệu của Sloan Digital Sky Survey (SDSS) - tổ chức hợp tác quốc tế lớn nhất về khảo sát thiên văn bắt đầu làm việc từ năm 2000 – là một minh chứng điển hình về sự phát triển của công nghệ thu thập dữ liệu. Trong vài tuần hoạt động đầu tiên, kính viễn vọng đầu tiên của SDSS tại New Mexico đã thu thập được lượng dữ liệu nhiều hơn dung lượng dữ liệu được tích lũy trong toàn bộ lịch sử thiên văn học trước đó. Hiện tại, sau một thập kỷ, kho tài nguyên dữ liệu của SDSS lên tới 140 TB. Kính viễn vọng kế tiếp của SDSS (Large Synoptic Survey Telescope) đặt tại Chile, được bắt đầu hoạt động vào năm 2016, sẽ thu nhận được khối lượng dữ liệu như vậy (140 TB) chỉ trong năm ngày.

Các kết quả của sự phát triển công nghệ phần cứng máy tính đã tạo điều kiện thuận lợi cho sự phát triển công nghệ cơ sở dữ liệu (liên quan tới hoạt động tổ chức và quản lý dữ liệu) và công nghệ mạng (liên quan tới hoạt động truyền dẫn dữ liệu), hợp thành một nền tảng kỹ thuật tổng hợp cho sự bùng nổ thông tin.

+ Công nghệ CSDL đã và đang phát triển không ngừng nhằm đáp ứng nhu cầu quản lý dữ liệu ngày càng nâng cao của xã hội loài người (nói chung) và trong hoạt động quản lý (nói riêng). Hình 1.2 trình bày quá trình tiến hóa công nghệ CSDL theo quan điểm của J. Han và M. Kamber [HK0106].

Trong quá trình tiến hóa của công nghệ CSDL, nhiều hệ quản trị cơ sở dữ liệu được phát triển và năng lực của hệ quản trị cơ sở dữ liệu cũng ngày được nâng cao. Sự tăng trưởng nổi bật về kích thước của cơ sở dữ liệu quản lý đã dẫn đến có nhiều cơ sở dữ liệu với kích thước hàng trăm TB (1TB = 1000 GB) xuất hiện. Chẳng hạn, cơ sở dữ liệu của Trung tâm Tính toán Khoa học Nghiên cứu Năng lượng Quốc gia Mỹ (National Energy Research Scientific Computing Center: NERSC) tới tháng 3/2010 đã đạt

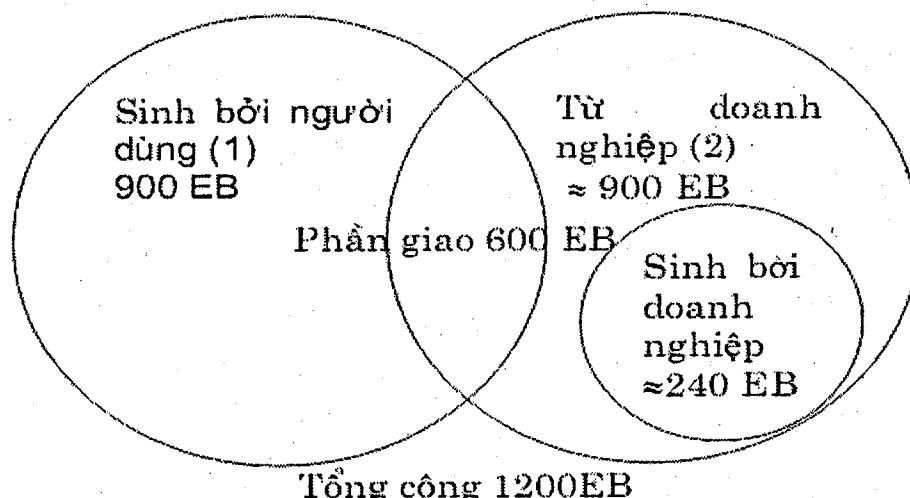
³ “Intel 40th Anniversary backgrounder”.

http://www.intel.com/pressroom/enhanced/40th_Anniversary/40th_anniversary_backgrounder.pdf?iid=pr_smrelease_40th_addlmat1

khoảng 460 TB⁴. Cơ sở dữ liệu của YouTube sau hai năm hoạt động đã có tới hàng trăm triệu video, dung lượng cơ sở dữ liệu của YouTube tăng gấp đôi sau mỗi chu kỳ 5 tháng. Hệ thống siêu thị bán lẻ Wal-Mart, mỗi giờ có hơn 1 M giao dịch khách hàng, cung cấp các cơ sở dữ liệu mà dung lượng chung ước tính lên tới hơn 2,5 PB (1 PB = 1000 TB⁵).

+ Sự phát triển công nghệ mạng cá về quy mô và tốc độ đã tạo ra sự tăng trưởng mạnh mẽ về năng lực truyền dẫn thông tin. Theo báo cáo tổng hợp của CISCO, tổng dung lượng dữ liệu thông qua giao vận IP trong một tháng đã tăng từ 14.686 PB vào năm 2009 lên 20.396 PB vào năm 2010 và dự báo lên tới 63.463 PB vào năm 2014. Theo dự báo, độ tăng trung bình hàng năm về dung lượng dữ liệu qua giao vận IP trong giai đoạn 2009-2014 đạt khoảng 34% (Bảng 1.1).

Đặc biệt, World Wide Web đã trở thành mạng thông tin khổng lồ, trong đó số lượng trang Web được đánh chỉ số đã lên tới con số hàng chục tỷ (theo số liệu công bố vào ngày 23/01/2011 của WorldWideWeb.com, đã có hơn 13 tỷ rưỡi trang Web được đánh chỉ số)⁶.



Hình 1.3. Dung lượng dữ liệu tổng thể năm 2010 đạt khoảng 1.260 EB (1EB = 1tỷ GB) [IDC10]. *Chú thích:* (1) Người dùng và nhân viên tạo, lưu giữ, hoặc sao chép thông tin cá nhân; (2) Doanh nghiệp tạo, vận chuyển, lưu trữ, quản lý, hoặc bảo mật.

⁴ http://www.nersc.gov/news/annual_reports/annrep0809/annrep0809.pdf

⁵ Dãy đơn vị đo dung lượng nhỏ được xếp theo chiều tăng 1000 lần: Byte (B), Kilo bytes (KB), Mega B (MB), Giga B (GB), Texa B (TB), Peta B (PB), Exa B (EB), Zetta B (ZB), Yotta B (YB). Như vậy, 1 EB = 1 tỷ GB và 1 ZB = 1 nghìn tỷ GB.

⁶ <http://www.worldwidewebsize.com/>

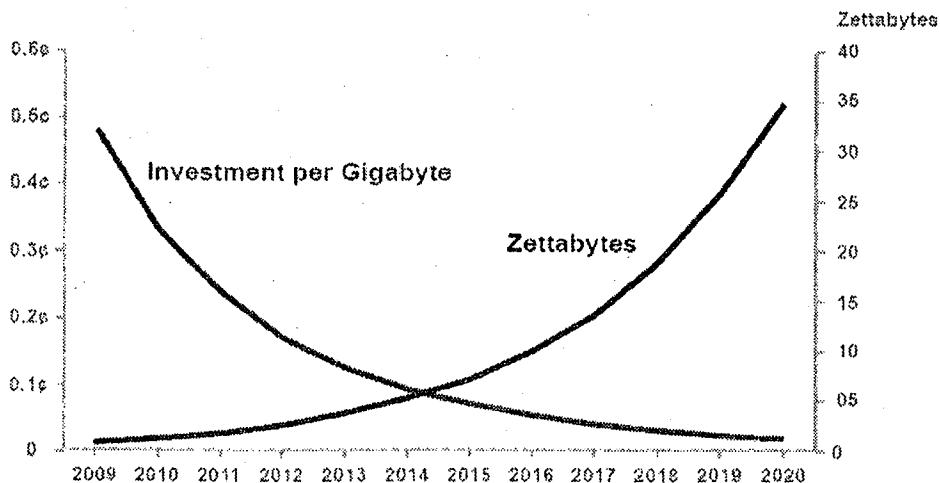
1.1.1.2. Về mặt xã hội

Xu thế phát triển xã hội thông tin đã mở rộng đội ngũ tác nhân tạo lập và sử dụng dữ liệu. Nguồn dữ liệu được tạo lập, khai thác và truyền dẫn không chỉ có trong hoạt động tác nghiệp tại các công ty, mà một lượng dữ liệu khổng lồ khác đã được một lực lượng hùng hậu các cá nhân tạo lập và phổ biến trên Internet trên các trang web cá nhân, các mạng xã hội... Tới tháng 2/2011, mạng xã hội Facebook đã bao gồm khoảng 40 tỷ ảnh⁷.

Tại Hình 1.3, vào năm 2010, dung lượng dữ liệu tổng thể toàn thế giới đã đạt khoảng 1.260 EB, trong đó có tới 900 EB dữ liệu do người sử dụng tạo ra (UGC: User-Generated Content); dung lượng dữ liệu loại này đã gấp gần 4 lần dung lượng dữ liệu được các doanh nghiệp tạo lập ra (khoảng 240 EB).

1.1.1.3. Chi phí tạo lập dữ liệu mới ngày càng giảm

Theo tính toán dự báo của IDC được công bố vào tháng 5/2010, giá thành tạo mới 1 GB dữ liệu là gần 0,5 xu Mỹ vào năm 2009; giá thành này sẽ tiếp tục giảm trong các năm tiếp theo và dự kiến giá tạo mới một GB dữ liệu sẽ vào khoảng 0,02 xu Mỹ vào năm 2020 (Hình 1.4). Điều có lợi này vừa là kết quả của cuộc cách mạng công nghệ vừa là một nguyên nhân góp phần tăng trưởng dung lượng dữ liệu.



Hình 1.4. Dung lượng dữ liệu tổng thể và giá thành tạo lập dữ liệu giai đoạn 2009-2020 [IDC10].

⁷ http://www.economist.com/node/15557443?story_id=15557443, đăng ngày 25/2/2010.

Sau đây là một số ví dụ minh họa về tính phong phú của hiện tượng “bùng nổ dữ liệu”. Dữ liệu tổng thể tiếp tục phát sinh, lưu trữ bao gồm giao dịch thương mại, cuộc gọi điện thoại, dữ liệu khoa học: thiên văn, sinh học, Web, văn bản, ảnh,... Theo tổng hợp của IDC, tuy có bị ảnh hưởng của khủng hoảng kinh tế trong các năm 2008-2009 song dung lượng dữ liệu tổng thể vào năm 2010 đã tăng 62% so với gần 0,8 ZB (800 EB) vào năm 2009 để đạt tới 1,26 ZB (1260 EB).

Cũng theo IDC, nguồn dữ liệu tổng thể được dự báo lên tới 35 ZB vào năm 2020. Độ dốc của đường biểu diễn dung lượng nguồn dữ liệu tổng thể trong Hình 1.4 ngày càng lớn, chứng tỏ độ tăng trưởng dữ liệu ngày càng cao.

1.1.2. Ngành công nghiệp dựa trên dữ liệu

Việc tạo lập, thu thập và lưu trữ dữ liệu với kết quả là xuất hiện các kho chứa dữ liệu khổng lồ được liệt kê trên đây không ngoài mục đích khai phá dữ liệu nhằm phát hiện các tri thức mới giúp ích cho hoạt động của con người trong tập hợp dữ liệu.

Theo Jim Gray, chuyên gia của Microsoft, người được nhận giải thưởng Turing năm 1998, thì “Chúng ta đang ngập trong dữ liệu khoa học, dữ liệu y tế, dữ liệu nhân khẩu học, dữ liệu tài chính, và các dữ liệu tiếp thị. Con người không có đủ thời gian để xem xét dữ liệu như vậy... Vì vậy, chúng ta phải tìm cách tự động phân tích dữ liệu, tự động phân loại nó, tự động tóm tắt nó, tự động phát hiện và mô tả các xu hướng trong nó, và tự động chỉ dẫn các dị thường. Đây là một trong những lĩnh vực năng động và thú vị nhất của cộng đồng nghiên cứu cơ sở dữ liệu. Các nhà nghiên cứu về thống kê, trực quan hóa, trí tuệ nhân tạo, và học máy đang đóng góp cho lĩnh vực này. Tính rộng lớn của lĩnh vực đã làm cho nó trở nên khó khăn để nắm bắt những tiến bộ phi thường trong vài thập kỷ gần đây” [HK0106].

Kenneth Cukier đưa ra nhận định tương tự “*Thông tin từ khan hiếm tới dư dật. Điều đó mang lại lợi ích mới to lớn... tạo nên*

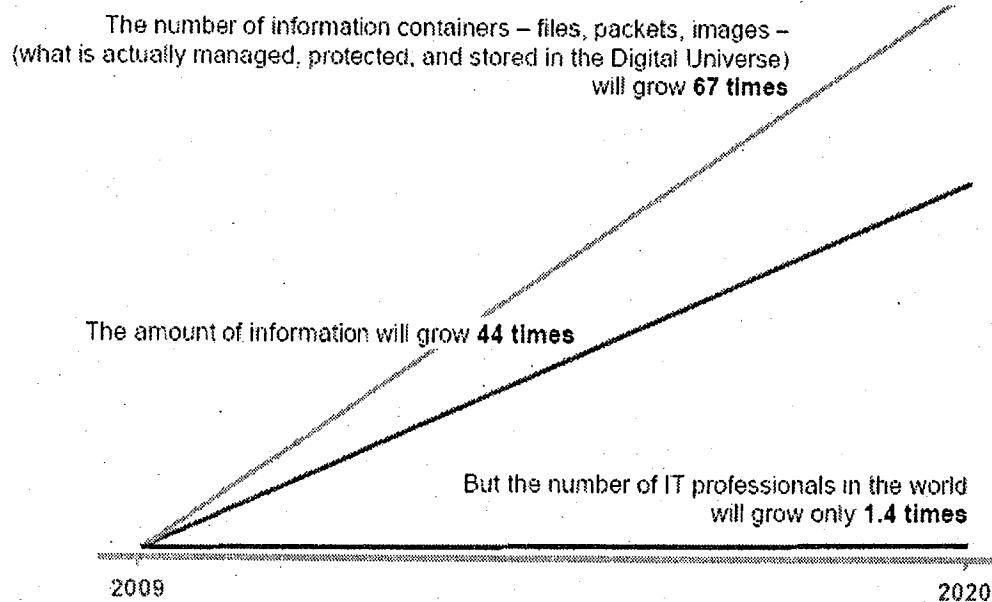
khả năng làm được nhiều việc mà trước đây không thể thực hiện được: nhận ra các xu hướng kinh doanh, ngăn ngừa bệnh tật, chống tội phạm... Được quản lý tốt, dữ liệu như vậy có thể được sử dụng để mở khóa các nguồn mới có giá trị kinh tế, cung cấp những hiểu biết mới vào khoa học và tạo ra lợi ích từ quản lý..." (xem chỉ dẫn 7).

Như đã được trình bày, nhiều tri thức có ích đang tiềm ẩn trong tập dữ liệu đồ sộ được thu thập và lưu giữ. Tuy nhiên, dung lượng khổng lồ của dữ liệu được tạo lập, thu thập và lưu trữ lại tạo nên các thách thức mới cho con người trong việc hiểu và xử lý dữ liệu, dẫn đến tình trạng con người "*ngập trong dữ liệu*". Cũng theo Kenneth Cukier (xem chỉ dẫn 7) thì thông tin từ khan hiếm tới dư dật "*cũng là một nỗi đau đần lớn... Con người đã từ lâu phàn nàn rằng họ đã phải bơi trong thông tin (dữ liệu)*". "Nỗi đau đần" mà Kenneth Cukier muốn nói đến là hiện tượng con người ngày càng khó tiếp cận được nguồn dữ liệu to lớn đang ngày càng gia tăng cũng như khó tiếp cận được cách thức để quản lý tốt được nguồn dữ liệu khổng lồ đó nhằm mang lại lợi ích to lớn trong việc nhận ra các xu hướng kinh doanh, ngăn ngừa bệnh tật, chống tội phạm...

Hình 1.5 cung cấp một dự báo IDC Digital Universe Study về độ tăng dữ liệu tổng thể trong "vũ trụ số" giai đoạn 2009-2020. So với năm 2009, vào năm 2020, số lượng đối tượng chứa tin tăng 67 lần, dung lượng dữ liệu ước đạt 35 ZB tăng 44 lần, tuy nhiên lực lượng lao động về CNTT chỉ tăng 1,4 lần. Sự chênh lệch giữa tốc độ tăng dung lượng dữ liệu so với tốc độ tăng lực lượng lao động CNTT cũng trở thành một thách thức lớn trong xử lý và sử dụng dữ liệu.

Hơn nữa, thế giới ngày nay đang trong thời kỳ quá độ chuyển từ kinh tế hàng hóa (good economy, hay kinh tế hướng hàng hóa: good-dominant economy) sang kinh tế dịch vụ (service economy, hay kinh tế hướng dịch vụ: service-dominant economy), mà tri thức đã trở thành động lực chủ chốt cho tăng trưởng kinh tế (Chương 2). Trong xu thế chuyển sang nền kinh tế hướng dịch vụ, yêu cầu quản lý tốt dữ liệu lớn càng trở nên cấp bách đối với con người trong việc giải quyết tình trạng "*ngập trong dữ liệu mà khát tri thức*".

Như vậy, một yêu cầu cấp thiết đặt ra là phải xây dựng được các phương pháp mới xử lý (tự động) dữ liệu để phù hợp với hoàn cảnh khối lượng dữ liệu đã rất lớn và đang tăng trưởng với tốc độ ngày càng cao. Các nhà nghiên cứu và triển khai đã đề cập tới cuộc cách mạng công nghiệp dữ liệu (“the industrial revolution of data”). Một lĩnh vực khoa học mới mang tên “khoa học dữ liệu lớn” (xem chỉ dẫn 7) (science of big data) đã được hình thành. Từ nguồn dữ liệu khổng lồ được quản lý tốt, chúng ta sẽ thu nhận được các tri thức về xu hướng kinh doanh, về ngăn ngừa bệnh tật, về chống tội phạm.



Hình 1.5. Độ tăng của dữ liệu tổng thể và lực lượng lao động CNTT giai đoạn 2009-2020 [IDC10].

Cũng theo Kenneth Cukier, công nghiệp quản lý và phân tích dữ liệu để nhận được tri thức tiềm ẩn từ dữ liệu (công nghiệp dựa trên dữ liệu) được định giá lên tới hơn 100 tỷ đô la Mỹ tại thời điểm năm 2010 và có tốc độ tăng trưởng khoảng 10% hàng năm (gần gấp đôi so với tốc độ tăng trưởng của kinh doanh phần mềm nói chung). Trong một vài năm cuối của thập niên 2000, các tập đoàn CNTT hàng đầu thế giới như Oracle, IBM, Microsoft và SAP đã chi tới hơn 15 tỷ đô la Mỹ để mua lại các công ty phần mềm chuyên về quản lý và phân tích dữ liệu.

Định nghĩa công nghệ dữ liệu lớn của IDC vào năm 2011 [GR11] cung cấp một cách hiểu về nội dung của công nghệ mới

này: Công nghệ dữ liệu lớn mô tả một thế hệ mới của công nghệ và kiến trúc hạ tầng, được thiết kế tiết kiệm nhất để thu được giá trị từ khối lượng rất lớn của dữ liệu đa dạng, bằng cách cho phép chụp tốc độ cao, phát hiện và/hoặc phân tích⁸.

Song hành với xu hướng hoạt động quản lý, phát hiện và phân tích dữ liệu ngày càng được tăng cường, giám đốc thông tin (Chief information officer: CIO) có vai trò ngày càng nổi bật trong bộ máy điều hành của tổ chức. Họ là các nhà khoa học dữ liệu (data scientist), những người tích hợp được các kỹ năng của lập trình viên, nhà thống kê và nghệ nhân nhằm “đào được vàng cối ẩn trong núi dữ liệu”. Đặc điểm “nghệ nhân” của nhà khoa học dữ liệu còn được chỉ dẫn như là “người kể chuyện” (storyteller). Điều này có thể được giải thích là nhà khoa học dữ liệu có năng lực “kể lại được câu chuyện của dữ liệu”, để từ đó cho phép nhận ra được các tri thức hữu ích, cần thiết từ “núi dữ liệu đồ sộ”. Theo Quỹ Khoa học Quốc gia Mỹ (NSF), nhà khoa học dữ liệu có các chức năng sau đây “thi hành sáng tạo hoạt động khảo sát và phân tích, tăng cường tư vấn, hợp tác, và phối hợp năng lực của những người khác để tiến hành nghiên cứu và giáo dục bằng các bộ dữ liệu số; đi tiên phong trong việc phát triển sáng tạo trong lĩnh vực công nghệ cơ sở dữ liệu và khoa học thông tin, bao gồm phương pháp trực quan hóa dữ liệu và phát hiện tri thức để áp dụng vào các lĩnh vực khoa học và giáo dục liên quan đến các bộ dữ liệu; thi hành một cách tốt nhất cả theo khía cạnh thực tiễn lẫn khía cạnh công nghệ; đóng vai trò cố vấn để khởi tạo hoặc chuyển đổi dữ liệu cho các nhà điều tra, sinh viên và những người khác có quan tâm tới khoa học dữ liệu; thiết kế và thi hành các chương trình giáo dục và tiếp cận cộng đồng làm cho lợi ích của các bộ dữ liệu và thông tin khoa học kỹ thuật số tới các nghiên cứu viên, giảng viên, sinh viên và công chúng trong một phạm vi rộng nhất có thể được” [NSF05]. A. Swan và S. Brown [SB08] quan niệm rằng nhà khoa học dữ liệu là những người nghiên cứu và thực hiện toàn bộ hoặc

⁸ Nguyên văn: "Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large volumes of a wide variety of data, by enabling high-velocity capture, discovery, and/or analysis".

bộ phận tập hợp các chức năng như định nghĩa trên đây của NSF. Hai tác giả phân biệt nhà khoa học dữ liệu với nhà tạo lập dữ liệu (tác giả dữ liệu), người quản lý dữ liệu, và chuyên viên thư viện. Một nhà khoa học dữ liệu hoặc là nhà khoa học máy tính có kỹ năng đáng kể miền ứng dụng hoặc là nhà khoa học miền ứng dụng có kỹ năng đáng kể về tính toán.

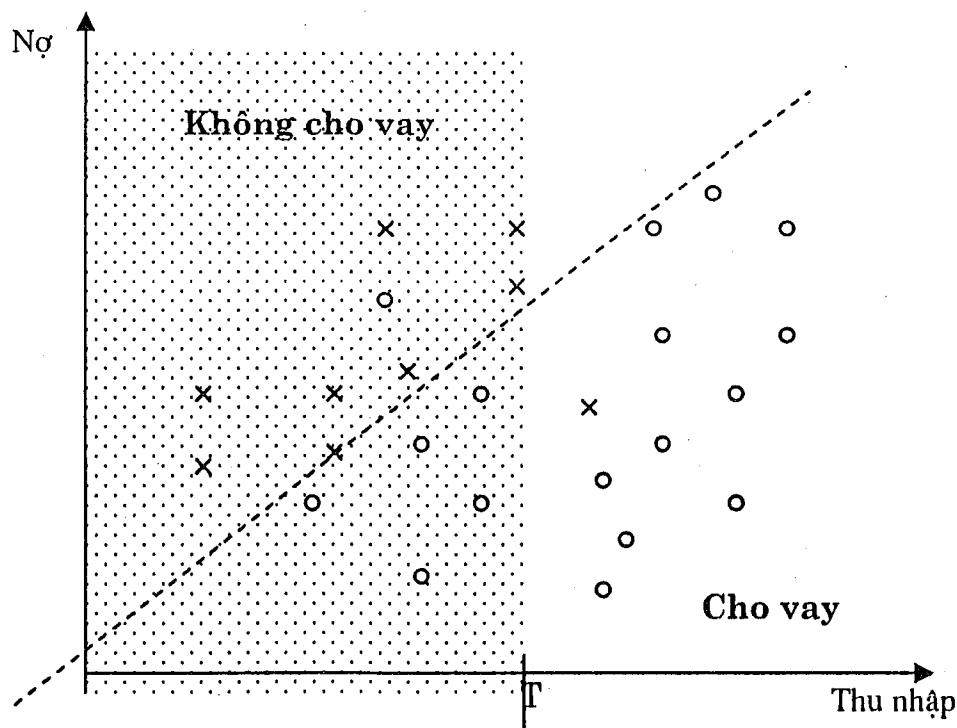
Thích ứng với hoàn cảnh dữ liệu lớn, hỗ trợ đắc lực cho nhà khoa học dữ liệu là các phương pháp xử lý dữ liệu mới và các bộ công cụ tiện ích thi hành với các phương pháp này để phát hiện ra các tri thức mới, có giá trị, hữu dụng đang tiềm ẩn trong dữ liệu lớn đó. Xây dựng và phát triển phương pháp và công cụ xử lý dữ liệu lớn nhằm mục đích phát hiện tri thức tiềm ẩn là nội dung của lĩnh vực phát hiện tri thức trong cơ sở dữ liệu (Knowledge Discovery in Databases: KDD). Khai phá dữ liệu (Data Mining) là bài toán xử lý dữ liệu cơ bản nhất trong quá trình phát hiện tri thức trong cơ sở dữ liệu. Trong nhiều trường hợp, hai khái niệm Khai phá dữ liệu và Phát hiện tri thức trong cơ sở dữ liệu còn mang cùng một nội dung.

Như được mô tả trong Hình 1.2, J. Han và M. Kamber [HK0106] cho rằng quá trình tiến hóa của lĩnh vực công nghệ cơ sở dữ liệu (CSDL), trong đó công nghệ khai phá dữ liệu (Data Mining) được coi là giai đoạn tiến hóa mới của công nghệ CSDL. Quá trình tiến hóa này được bắt đầu từ cuối những năm 1980 và không ngừng được phát triển về bề rộng và chiều sâu.

1.2. KHÁI NIỆM PHÁT HIỆN TRI THỨC TRONG CƠ SỞ DỮ LIỆU

Lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL là một lĩnh vực rộng lớn, đã cuốn hút các phương pháp, thuật toán và kỹ thuật từ nhiều chuyên ngành nghiên cứu khác nhau như học máy, thu nhận mẫu, CSDL, thống kê, trí tuệ nhân tạo, thu nhận tri thức trong hệ chuyên gia cùng hướng tới một mục tiêu thống nhất là trích lọc ra được các "tri thức" từ dữ liệu trong các kho chứa khổng lồ. Tính phong phú và đa dạng của lĩnh vực khai phá dữ liệu dẫn đến một thực trạng là tồn tại các quan niệm khác

nhau về các chuyên ngành khoa học - công nghệ gần gũi nhất với lĩnh vực đó.

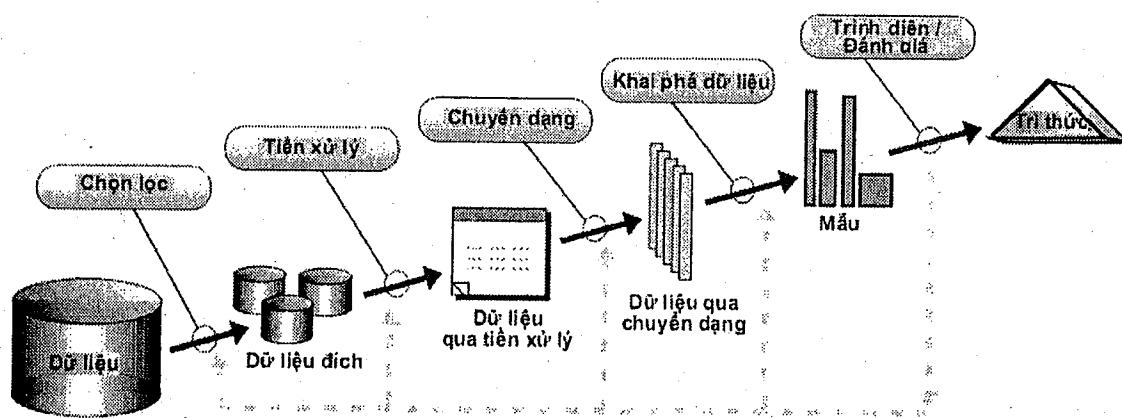


Hình 1.6. Ngưỡng đơn T theo thu nhập để phân lớp cho vay: Vùng bị phủ bởi ngưỡng T (vùng các dấu chấm) tương ứng quyết định không cho vay [FPS96]
(Lưu ý, vùng phía trên đường nghiêng rời nét cho quyết định tốt hơn).

Giáo trình này tán thành quan niệm của J. Han và M. Kamber coi lĩnh vực khai phá dữ liệu là giai đoạn phát triển mới của công nghệ CSDL và có liên quan mật thiết với nhiều ngành. Như vậy, có thể gắn lĩnh vực này với chuyên ngành hệ thống thông tin.

Ví dụ 1.1. (Frawley, Piatetski-Shapiro và Matheus [FPS96])

Hình 1.6 trình bày một tập dữ liệu giả định về vay nợ ngân hàng gồm 23 trường hợp được biểu diễn trong không gian hai chiều. Mỗi điểm trên đồ thị biểu diễn một trường hợp vay nợ ở ngân hàng trong quá khứ. Trục hoành biểu diễn thu nhập còn trực tung biểu diễn tổng nợ cá nhân của người đi vay (tiền thế chấp, tiền chi trả ô tô...). Dữ liệu được phân thành hai lớp: lớp x gồm những người thiếu khả năng trả nợ ngân hàng và lớp o gồm những người có tình trạng tốt.



Hình 1.7. Quá trình phát hiện tri thức trong cơ sở dữ liệu [FPS96]

Khái niệm 1.1. [FPS96]

Phát hiện tri thức trong cơ sở dữ liệu (đôi khi còn được gọi là khai phá dữ liệu) là một quá trình không tầm thường tìm ra những mẫu có giá trị, mới, hữu ích tiềm năng và hiểu được trong dữ liệu.

Là lĩnh vực nghiên cứu và triển khai được phát triển rất nhanh chóng và có phạm vi rất rộng lớn, lại được rất nhiều nhóm nghiên cứu tại nhiều trường đại học, viện nghiên cứu, công ty ở nhiều quốc gia trên thế giới quan tâm, cho nên tồn tại rất nhiều cách tiếp cận khác nhau đối với lĩnh vực phát hiện tri thức trong CSDL. Chính vì lý do đó mà trong nhiều tài liệu, như đã nói ở trên, các nhà khoa học trên thế giới đã dùng nhiều thuật ngữ khác nhau, mà các thuật ngữ này được coi là mang cùng nghĩa với KDD như chiết lọc tri thức (knowledge extraction), phát hiện thông tin (information discovery), thu hoạch thông tin (information harvesting), khai quật dữ liệu (data archaeology), xử lý mẫu dữ liệu (data pattern processing)... Hơn nữa, trong nhiều trường hợp, hai khái niệm "*Phát hiện tri thức trong cơ sở dữ liệu*" và "*khai phá dữ liệu*" còn được dùng thay thế nhau [FPS96]. Hai khái niệm *khai phá dữ liệu* và *phát hiện tri thức trong các CSDL* thường cặp đôi với nhau.

1.2.1. Giải thích một số thuật ngữ

Một số thuật ngữ có trong định nghĩa 1.1 trên đây cần được giải thích là "mẫu", "có giá trị", "mới", "hữu ích" và "hiểu được". Dưới đây trình bày một số giải thích sơ bộ về các khái niệm này nhằm làm tường minh thêm ngữ nghĩa của khái niệm KDD trong định nghĩa 1.1.

- *Dữ liệu* (chính xác hơn là *tập dữ liệu*) được hiểu như là một tập F gồm hữu hạn các *trường hợp* (*sự kiện*). Theo nội dung của phát hiện tri thức trong các CSDL, dữ liệu phải bao gồm nhiều trường hợp. Trong ví dụ 1.1, F là tập hợp gồm 23 trường hợp (bản ghi) với 3 trường thông tin (thuộc tính) tương ứng chứa các giá trị về *số nợ*, *thu nhập* và *tình trạng vay nợ*. Trong bài toán khai phá văn bản, tập dữ liệu F chính là tập hợp các văn bản có thể có trong miền ứng dụng. Trong bài toán khai phá luật kết hợp giao dịch, tập F bao gồm tất cả các giao dịch có thể có được xem xét trong miền áp dụng của bài toán.

- *Mẫu*: Trong quá trình KDD, người ta sử dụng một ngôn ngữ L để biểu diễn các tập con các sự kiện (dữ liệu) thuộc vào tập sự kiện F, theo đó mỗi biểu thức E trong ngôn ngữ L sẽ biểu diễn một tập con FE tương ứng các sự kiện trong F. E được gọi là *mẫu* nếu nó đơn giản hơn (theo một ngữ cảnh nào đó) so với việc liệt kê các sự kiện thuộc FE. Chẳng hạn, biểu thức "THUNHẬP < \$t" (mô hình chứa một biến THUNHẬP) trong mệnh đề "Nếu THUNHẬP < \$t thì người vay nợ rơi vào tình trạng không thể chi trả" sẽ là một mẫu khi cho biến t nhận một giá trị thích hợp. Như trình bày bằng đồ thị tại Hình 1.6, khi biến t nhận một giá trị cụ thể T mẫu này (biểu diễn mọi trường hợp có THUNHẬP < T) hiển nhiên là gọn hơn so với việc liệt kê 14 trường hợp cụ thể. Tương tự, nếu F là tập các trang Web trong kho lưu trữ của một máy tìm kiếm (chẳng hạn Google) thì mẫu "tài liệu có chứa từ cụm từ "Search Engine"" sẽ biểu diễn một tập bao gồm một số lượng rất lớn các tài liệu Web có chứa cụm từ "Search Engine" đó.

• Quá trình KDD thường bao gồm *nhiều bước là chuẩn bị dữ liệu, tìm kiếm mẫu, ước lượng tri thức, tinh chế sự tương tác nội* tại sau khi chuyển dạng dữ liệu. Quá trình được thừa nhận là ***không*** ***tầm thường*** theo nghĩa là quá trình đó không chỉ nhiều bước mà còn được thực hiện lặp đi lặp lại, và quan trọng hơn, quá trình đó bao hàm một mức độ tìm kiếm tự động. Chẳng hạn trong *Ví dụ 1.1*, khi tính toán ý nghĩa về thu nhập của một người, nếu chỉ thông qua các tác động đơn giản mà chúng ta thu nhận được một kết luận nào đó có thể là hữu ích về mối quan hệ giữa thu nhập và tình trạng vay ngân hàng, chẳng hạn như "người có thu nhập cao thì có khả năng cao ở tình trạng vay nợ tốt", thì đừng vội cho rằng đó đã là một khám phá (hoặc đừng cho rằng một tri thức đã được phát hiện).

• ***Có giá trị:*** Mẫu được phát hiện cần phải có ***giá trị*** đối với các dữ liệu mới (xuất hiện trong tương lai) theo một mức độ chân thực nào đấy. Tính chất "có giá trị" được hiểu theo nghĩa liên quan tới một ***độ đo tính có giá trị (chân thực)*** là một hàm C ánh xạ một biểu thức thuộc ngôn ngữ biểu diễn mẫu L tới một không gian đo được (bộ phận hoặc toàn bộ) M_C . Một biểu thức E trong L biểu diễn một tập con $F_E \subset F$ có thể được gán một độ đo chân thực $c = C(E, F)$.

Chẳng hạn, nếu đường biên xác định mẫu " $THUNHẬP < \$t$ " như chỉ dẫn trong Hình 1.6 được dịch sang phải (biến $THUNHẬP$ nhận giá trị lớn hơn) thì độ chân thực của mẫu mới sẽ bị giảm xuống bởi vì nó đã bao gói thêm các tình huống vay tốt lại bị đưa vào vùng không cho vay nợ.

Tương tự, mẫu "Nếu $a*THUNHẬP + b*Nợ < 0$ (thuộc mô hình tuyến tính hai biến $THUNHẬP$ và $Nợ$ trong $a*THUNHẬP + b*Nợ$) thì người vay nợ rơi vào tình trạng không thể chi trả" biểu diễn một nửa mặt phẳng phía trên của đường rời nét trong Hình 1.6 sẽ cho độ chân thực cao hơn (hay được coi là "có giá trị hơn") so với mọi mẫu thuộc mô hình một biến " $THUNHẬP < \$t$ ".

• *Tính mới:* Mẫu phải là mới trong một miền xem xét nào đó, ít nhất là hệ thống đang được xem xét. *Tính mới có thể đo được* khi quan tâm tới sự thay đổi trong dữ liệu (bằng việc so sánh giá trị hiện tại với giá trị quá khứ hoặc giá trị kỳ vọng) hoặc tri thức (tri thức mới quan hệ như thế nào với các tri thức đã có). Tổng quát, điều này có thể được đo bằng một hàm $N(E,F)$ hoặc là độ đo về tính mới hoặc là độ đo kỳ vọng.

• *Hữu ích tiềm năng:* Mẫu cần có khả năng chỉ dẫn tới các tác động hữu dụng và *được đo bởi một hàm tiện ích*. Chẳng hạn, hàm U ánh xạ các biểu thức trong L tới một không gian đo có thứ tự (bộ phận hoặc toàn bộ) M_U , theo đó $u = U(E,F)$. Ví dụ, trong tập dữ liệu vay nợ, hàm này có thể là *sự tăng hy vọng theo sự tăng lãi của nhà băng* (tính theo đơn vị tiền tệ) kết hợp với quy tắc quyết định được trình bày trong Hình 1.6.

• *Có thể hiểu được:* Một mục tiêu của KDD là tạo ra *các mẫu mà con người hiểu chúng dễ dàng hơn* các dữ liệu nền (dữ liệu sẵn có trong hệ thống). Chính vì lý do tiêu chí này là khó mà đo được một cách chính xác cho nên thường tính chất "có thể hiểu được" được thay bằng một độ đo về sự dễ hiểu. Tồn tại một số độ đo về sự dễ hiểu, các độ đo như vậy được sắp xếp từ cú pháp (tức là cỡ của mẫu theo bit) tới ngữ nghĩa (tức là dễ dàng để con người nhận thức được theo một tác động nào đó). Bởi lý do đó, chúng ta giả định rằng tính hiểu được là *đo được* bằng một hàm S ánh xạ biểu thức E trong L tới một không gian đo được có thứ tự (bộ phận hoặc toàn bộ) M_S ; theo đó, $s = S(E,F)$.

• *Độ hấp dẫn:* Một tiêu chí quan trọng, được gọi là *độ hấp dẫn* (interestingness), thường được coi như một *độ đo tổng thể về mẫu* là sự kết hợp của các tiêu chí *giá trị, mới, hữu ích và có thể hiểu được*. Một số hệ thống KDD thường sử dụng một hàm hấp dẫn dưới dạng hiển $i = I(E, F, C, N, U, S)$ thực hiện ánh xạ một biểu thức trong L vào một không gian đo được M_i . Một số hệ thống KDD khác lại có thể xác định giá trị hấp dẫn của mẫu một cách trực tiếp thông qua thứ tự của các mẫu được phát hiện.

Trong thực tiễn giải quyết các bài toán khai phá dữ liệu, người ta thường chỉ quan tâm đến độ hấp dẫn, còn các độ đo khác được mặc định coi là thành phần của độ hấp dẫn. Cụ thể là, khi thi hành một loại bài toán phát hiện tri thức cụ thể, một số độ đo tương ứng được tính toán nhằm xác định độ hấp dẫn của tri thức ("mẫu", "luật") đang được xem xét. Chẳng hạn, trong bài toán khai phá luật kết hợp, hai độ đo được xem xét, đó là *độ hỗ trợ* (xác định phạm vi ảnh hưởng của luật) và *độ tin cậy* (xác định tính tin cậy của luật) hợp thành độ hấp dẫn của luật kết hợp đã được khai phá. Tương tự, trong bài toán phân lớp, người ta sử dụng hai độ đo cơ bản là *độ hồi tưởng* (recall - khả năng bao gói ví dụ đúng) và *độ chính xác* (precision - khả năng chính xác khi xác định ví dụ đúng); đồng thời, một số độ đo mang ý nghĩa kết hợp từ hai độ đo này cũng được sử dụng.

- *Tri thức*: Một mẫu $E \in L$ được gọi là *tri thức* nếu như đối với một lớp người sử dụng nào đó, chỉ ra được một ngưỡng $i \in M_i$ mà độ hấp dẫn $I(E, F, C, N, U, S) > i$.

Chú ý rằng định nghĩa trên đây về khái niệm "tri thức" không mang một nghĩa tuyệt đối mà phụ thuộc vào quan điểm của người sử dụng hệ thống KDD ("một lớp người sử dụng nào đó"). Như một nội dung của sự kiện, nó chỉ là một định hướng cho người sử dụng và được xác định bằng bất kỳ hàm và ngưỡng nào được người sử dụng chọn. Chẳng hạn, trong bài toán khai phá luật kết hợp, chúng ta chỉ quan tâm tới các "*tập phổ biến*" là những tập có độ hỗ trợ vượt qua một ngưỡng *minsup* nào đó. Hơn nữa, chỉ các luật kết hợp có độ tin cậy vượt quá ngưỡng *minconf* mới được khai phá để cung cấp tri thức tới người sử dụng. Các ngưỡng *minsup* và *minconf* có thể được thay đổi theo lựa chọn của người sử dụng.

Theo cách hình thức hóa, thuyết minh chính xác cho định nghĩa trên đây về "tri thức" là chọn ngưỡng nào đó $c \in M_C$ (về tính "*có giá trị*"), $s \in M_S$ (về tính "*có thể hiểu được*") và $u \in M_U$ (về tính "*hữu ích*") và khi đó gọi mẫu E là *tri thức* nếu và chỉ nếu:

$$C(E, F) > c \text{ và } S(E, F) > s \text{ và } U(E, F) > u$$

Thông qua việc đặt các ngưỡng thích hợp với mục đích phát hiện tri thức, người sử dụng có thể nhấn mạnh một dự báo chính xác hoặc các mẫu hữu ích (vượt qua một ngưỡng độ đo đánh giá nào đó) qua những độ đo liên quan. Rõ ràng là tồn tại một không gian vô hạn cho phép ánh xạ I xác định "tri thức cần phát hiện". Quyết định như vậy là tự do đối với người sử dụng và được đặc trưng đối với từng miền ứng dụng.

Nghiên cứu về tính hấp dẫn của mẫu và tri thức (được gọi là độ đo hấp dẫn: interestingness measures) là một nội dung nghiên cứu quan trọng trong khai phá dữ liệu và phát hiện tri thức từ dữ liệu. Nhiều công trình nghiên cứu khái quát và chuyên sâu về nội dung này đã được công bố, chẳng hạn [Garry05, Grube09, HGEK07, Yao03, HZ10, GH06, ZZNS09]. Chương 2 sẽ giới thiệu chi tiết hơn về độ đo hấp dẫn.

Những điều trình bày trên đây cho thấy vai trò của hệ thống KDD cũng như vai trò của người sử dụng trong một phiên làm việc của mình, tạo nên sự cộng tác giữa người sử dụng và hệ thống KDD. Trong sự cộng tác đó, hệ thống KDD tạo thuận tiện cho người sử dụng có cách thức linh hoạt dùng các ngưỡng để được cung cấp "tri thức" từ hệ thống phù hợp với những dự đoán chủ quan của mình. Như vậy, có thể thấy rằng, cùng dùng một phần mềm KDD song mỗi người sử dụng lại có thể khai thác nó theo cách thức riêng của mình.

Khi phân tích nội dung ba cuốn sách hàng đầu về khai phá dữ liệu vào năm 2003, Z.H Zhou [Zhou03] cho biết sự khác biệt không nhỏ về nội dung khái niệm phát hiện tri thức từ dữ liệu của ba nhóm tác giả (J. Han và M. Kamber; IH Witten và E. Frank; D. Hand, H. Mannila và P. Smyth) đều là các chuyên gia hàng đầu về khai phá dữ liệu và phát hiện tri thức trong CSDL. Tài liệu này tiếp nhận quan niệm của Fayyad, Piatetsky-Shapiro, Smyth (được Z.H Zhou gọi là quan niệm truyền thống) coi KDD là một quá trình bao gồm nhiều bước thực hiện (xem **Khái niệm 1.1**), trong đó, khai phá dữ liệu là một bước thực hiện chính yếu. Cách hiểu như vậy đã quy định có sự phân biệt giữa hai khái niệm *khai phá dữ liệu* và *KDD*.

Khai niệm 1.2. (Frawley, Piatetski-Shapiro và Matheus [FPS96])

Khai phá dữ liệu là một bước trong quá trình Phát hiện tri thức trong cơ sở dữ liệu, thi hành một thuật toán khai phá dữ liệu để tìm ra các mẫu từ dữ liệu theo khuôn dạng thích hợp.

Cũng về khái niệm khai phá dữ liệu, theo B.Kovalerchuk và E.Vityaev [KV01], Friedman đã tổng hợp một số quan niệm liên quan sau đây:

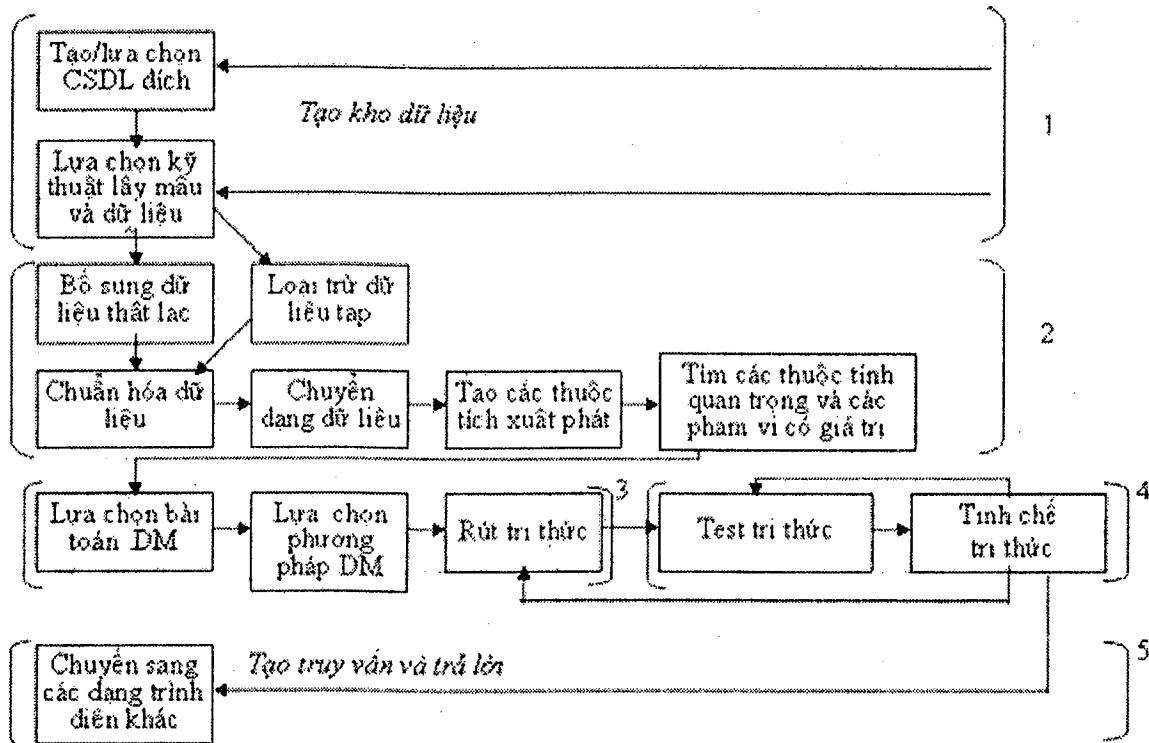
- Quá trình không tầm thường để nhận biết từ dữ liệu ra các mẫu có giá trị, mới, hữu dụng và hiểu được (Fayyad),
- Quá trình trích lọc các thông tin chưa biết trước, có thể nhận thức được, có thể tác động được từ CSDL lớn và sử dụng chúng để tạo ra quyết định công tác (Zekulin),
- Tập các phương pháp được dùng trong quá trình phát hiện tri thức nhằm tường minh các quan hệ và các mẫu chưa biết trước chứa trong dữ liệu (Ferruzza),
- Quá trình hỗ trợ quyết định khi tìm kiếm những mẫu thông tin chưa biết và hữu ích từ CSDL lớn (Parsaye).

Z.H Zhou [Zhou03] giới thiệu ba tiếp cận sau đây về nội dung khái niệm khai phá dữ liệu qua phân tích nội dung ba cuốn sách nêu trên:

- Quá trình khám phá tri thức thú vị từ lượng lớn dữ liệu được lưu trữ trong CSDL, hoặc kho dữ liệu, hoặc các kho thông tin khác (J. Han và M. Kamber),
- Sự khai thác thông tin tiềm ẩn, trước đó chưa biết, và có khả năng hữu ích từ dữ liệu (H. Witten và E. Frank),
- Phân tích tập dữ liệu quan sát (thường lớn) để tìm ra các mối quan hệ tường minh và tóm tắt dữ liệu theo cách mới để chúng vừa dễ hiểu vừa hữu ích cho chủ sở hữu dữ liệu (D. Hand, H. Mannila, P. Smyth).

1.2.2. Quá trình phát hiện tri thức trong cơ sở dữ liệu

Quá trình phát hiện tri thức trong cơ sở dữ liệu được mô tả trong Hình 1.7 và trình bày chi tiết hơn trong Hình 1.8. Tương ứng với sơ đồ mô tả chi tiết quá trình KDD (Hình 1.8), các nhóm bước thực hiện sau đây được tiến hành trong quá trình phát hiện tri thức trong CSDL:



Hình 1.8. Một mô tả chi tiết quá trình KDD

(1) *Mở rộng hiểu biết* về miền ứng dụng, về các tri thức với độ ưu tiên thích hợp và về mục đích của người dùng cuối. Có thể coi nội dung công việc này tương ứng với nội dung khảo sát bài toán trong quá trình xây dựng một hệ thống thông tin nói chung.

Một nhiệm vụ quan trọng của bước này là *xác định bài toán khai phá dữ liệu*. Mục 1.6 sẽ giới thiệu hai lớp bài toán khai phá dữ liệu điển hình nhất là mô tả và dự báo và các bài toán khai phá dữ liệu điển hình thuộc vào hai lớp này.

Khởi tạo tập dữ liệu đích, tạo kho dữ liệu: chọn tập dữ liệu và/hoặc hướng trọng tâm tới tập con các biến hoặc mẫu dữ liệu mà

trên đó công việc phát hiện tri thức được tiến hành. Tri thức miền ứng dụng có được thông qua việc mở rộng hiểu biết về miền ứng dụng nói trên đóng vai trò là nền tảng tri thức để khởi tạo tập dữ liệu đích, kho dữ liệu.

Chương 2 sẽ thảo luận chi tiết về vai trò của tri thức và bài toán phát hiện tri thức trong một miền ứng dụng.

(2) *Tiền xử lý dữ liệu*: thực hiện các thao tác cơ sở như giải quyết thiếu vắng giá trị, loại bỏ nhiễu hoặc yếu tố ngoại lai, kết nối các thông tin cần thiết tới mô hình hoặc loại bỏ nhiễu, quyết định chiến lược nhằm nắm bắt các trường dữ liệu (các thuộc tính), tính toán dãy thông tin thời gian và sự biến đổi được định trước.

Chất lượng của hệ thống khai phá dữ liệu phụ thuộc vào chất lượng của dữ liệu đầu vào. Mục tiêu của *làm sạch dữ liệu* nhằm đảm bảo dữ liệu đầu vào có chất lượng tốt.

Thu gọn và trình diễn dữ liệu có mục tiêu tìm được các đặc trưng hữu ích nhằm trình bày mỗi phụ thuộc dữ liệu theo mục đích của bài toán. Thu gọn dữ liệu được thi hành về chiều ngang (giảm số lượng đối tượng), chiều dọc (giảm số lượng trường dữ liệu) hoặc cả hai nhằm làm cho kích thước dữ liệu được xử lý, tăng tốc độ hoạt động của hệ thống. Sử dụng các phương pháp thu gọn hoặc biến đổi chiều nhằm rút gọn số lượng các biến cần quan tâm hoặc để tìm ra các mô tả bất biến đối với dữ liệu nhằm trình diễn dữ liệu phù hợp nhất. Do khối lượng dữ liệu trong bài toán KDD là rất lớn cho nên việc thi hành bước này là rất cần thiết. Khi thu gọn theo chiều ngang cần lưu ý là tập dữ liệu được chọn lựa sau khi thu gọn phải có *tính đại diện cho tập toàn bộ dữ liệu của miền ứng dụng*. Việc chọn lựa dữ liệu vào xây dựng mô hình khai phá dữ liệu (xây dựng nhà kho dữ liệu) thông thường cần được tiến hành theo một phương pháp đảm bảo tính "ngẫu nhiên" khi chọn lựa dữ liệu trong miền ứng dụng. Tương tự, khi thu gọn theo chiều dọc cần lưu ý các thuộc tính còn lại đảm bảo tính đại diện cho đối tượng trong bài toán khai phá dữ liệu đang xem xét. Trong không ít bài toán khai phá dữ liệu, khi thu gọn theo chiều dọc lại nhận được kết quả tốt hơn không chỉ về thời gian và không gian mà còn

cả về chất lượng của bài toán khai phá dữ liệu khi đạt được độ chính xác cao hơn vì đã loại bỏ được một số thuộc tính gây nhiễu. Phương pháp *phân tử chính* (Principal Component Analysis: PCA, xem chương 3) thường được sử dụng trong bài toán thu gọn theo chiều dọc.

Chương 3 sẽ thảo luận các nội dung chi tiết về các bài toán tiền xử lý dữ liệu và một số phương pháp điển hình giải quyết các bài toán này.

(3) *Khai phá dữ liệu* bao gồm ba nội dung là lựa chọn bài toán, phương pháp khai phá dữ liệu thích hợp và thi hành thuật toán khai phá dữ liệu.

Lựa chọn bài toán khai phá dữ liệu quyết định mục tiêu của quá trình KDD là loại bài toán khai phá dữ liệu cụ thể nào, chẳng hạn như bài toán phân lớp, hồi quy, phân đoạn... Tri thức miền ứng dụng thu nhận thêm được từ bước mở rộng hiểu biết về miền ứng dụng rất cần thiết cho việc lựa chọn bài toán khai phá dữ liệu.

Chọn lựa thuật toán khai phá dữ liệu: lựa chọn phương pháp và thuật toán được dùng để tìm mẫu trong dữ liệu. Nội dung này bao gồm cả việc quyết định các mô hình và tham số có thể được chấp nhận và thuật toán khai phá dữ liệu phù hợp với tiêu chuẩn tổng thể của quá trình KDD.

Thi hành thuật toán khai phá dữ liệu: tiến hành việc dò tìm các mẫu cần quan tâm dưới dạng trình bày riêng biệt hoặc một tập các trình bày như quy tắc phân lớp, cây, hồi quy, phân đoạn... Trong bước này, sự hỗ trợ của người dùng vẫn đóng một vai trò quan trọng. Các chương 4-7 của giáo trình này trình bày các phương pháp cho các khai phá dữ liệu cơ bản nhất.

Mục 1.2.3 trình bày một số nội dung chi tiết hơn về bước khai phá dữ liệu trong quá trình phát hiện tri thức từ dữ liệu.

(4) *Giải thích mẫu* đối với các mẫu được khám phá, có thể quay về một cách hợp lý tới bất kỳ bước nào từ bước đầu tiên tới bước thi hành thuật toán khai phá dữ liệu để thực hiện lặp.

(5) *Hợp nhất các tri thức* đã được khám phá, kết hợp các tri thức này thành một hệ thống trình diễn hoặc được biên soạn dễ dàng và kết xuất thành những thành phần hấp dẫn. Kiểm tra và giải quyết xung đột đối với tri thức được trích chọn.

Trong quá trình phát hiện tri thức trong các CSDL như được mô tả ở trên, chúng ta nhận thấy có sự tham gia của các kho dữ liệu (Data Warehouse).

Mô hình khai phá dữ liệu ngày càng được tiến hóa theo định hướng hỗ trợ chiến lược phát triển của tổ chức, nói riêng trong các doanh nghiệp, mô hình khai phá dữ liệu được tích hợp với mô hình kinh doanh. Chương 2 sẽ trình bày quá trình tiến hóa của mô hình khai phá dữ liệu.

1.2.3. Bước khai phá dữ liệu trong quá trình phát hiện tri thức từ dữ liệu

Trong quá trình phát hiện tri thức từ dữ liệu, khai phá dữ liệu là bước thực hiện chính yếu có nhiệm vụ tạo ra các mẫu mới từ dữ liệu đã được tiền xử lý và chuyển dạng.

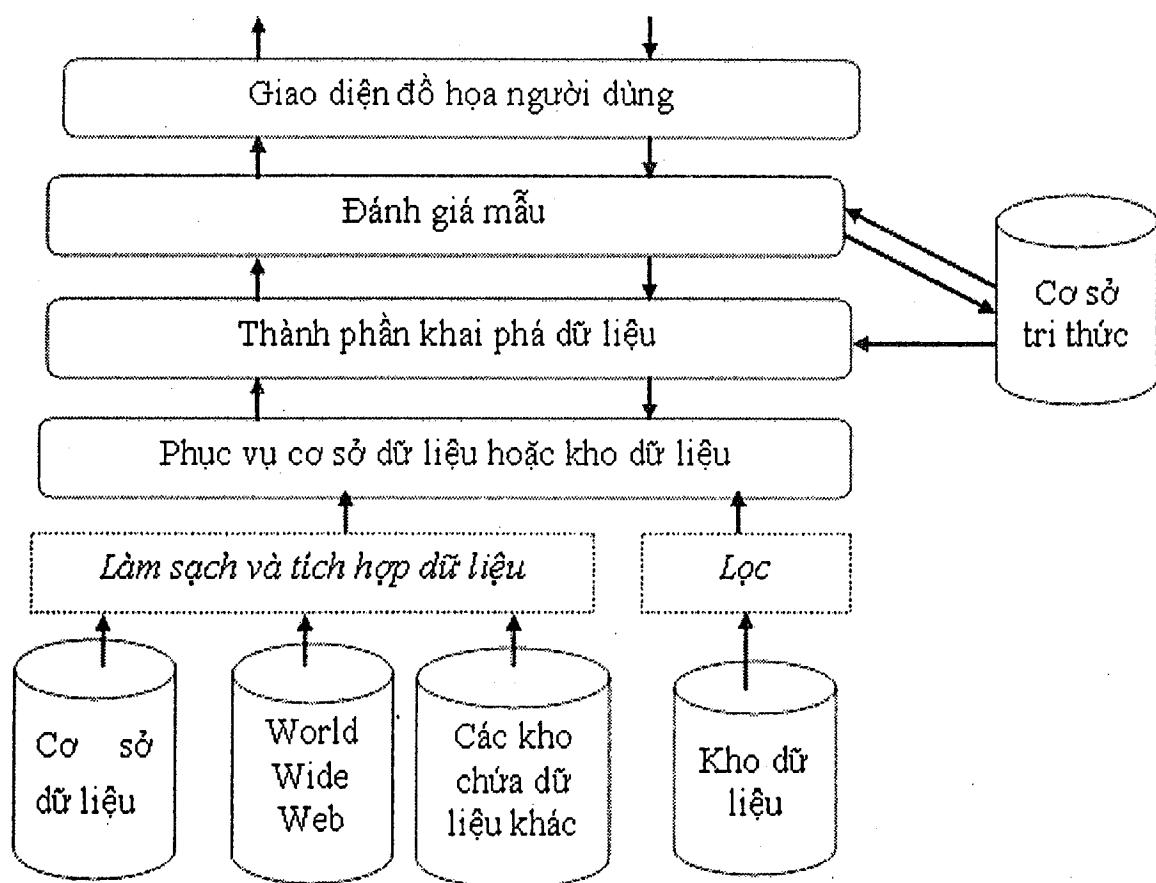
Việc chọn lựa bài toán khai phá dữ liệu nào đã được xác định chủ yếu từ bước mở rộng hiểu biết về miền ứng dụng. Kết quả tiền xử lý dữ liệu cung cấp thêm thông tin để làm rõ về bài toán khai phá dữ liệu đã được xác định.

Đối với bài toán khai phá dữ liệu đã được chọn, tồn tại nhiều thuật toán giải quyết. Về cơ bản, hiểu biết miền ứng dụng và tiền xử lý dữ liệu đã cơ bản định hình được thuật toán (hoặc sự kết hợp của một nhóm thuật toán) được tiến hành, trong đó việc chuyển dạng dữ liệu là hướng tới thuật toán hay nhóm thuật toán này. Các chương 5-8 sẽ trình bày một số thuật toán điển hình cho từng loại bài toán khai phá dữ liệu. Chương 10 giới thiệu một vài chỉ dẫn liên quan tới cách sử dụng các thuật toán trong bài toán khai phá dữ liệu.

1.2.4. Kiến trúc một hệ thống khai phá dữ liệu

Kiến trúc điển hình của một hệ thống khai phá dữ liệu được trình bày trong Hình 1.9 [HK0106]. Trong kiến trúc hệ thống này,

các nguồn dữ liệu cho các hệ thống khai phá dữ liệu bao gồm hoặc cơ sở dữ liệu, hoặc kho dữ liệu, hoặc World Wide Web, hoặc kho chứa dữ liệu kiểu bất kỳ khác, hoặc tổ hợp các kiểu đã liệt kê nói trên.



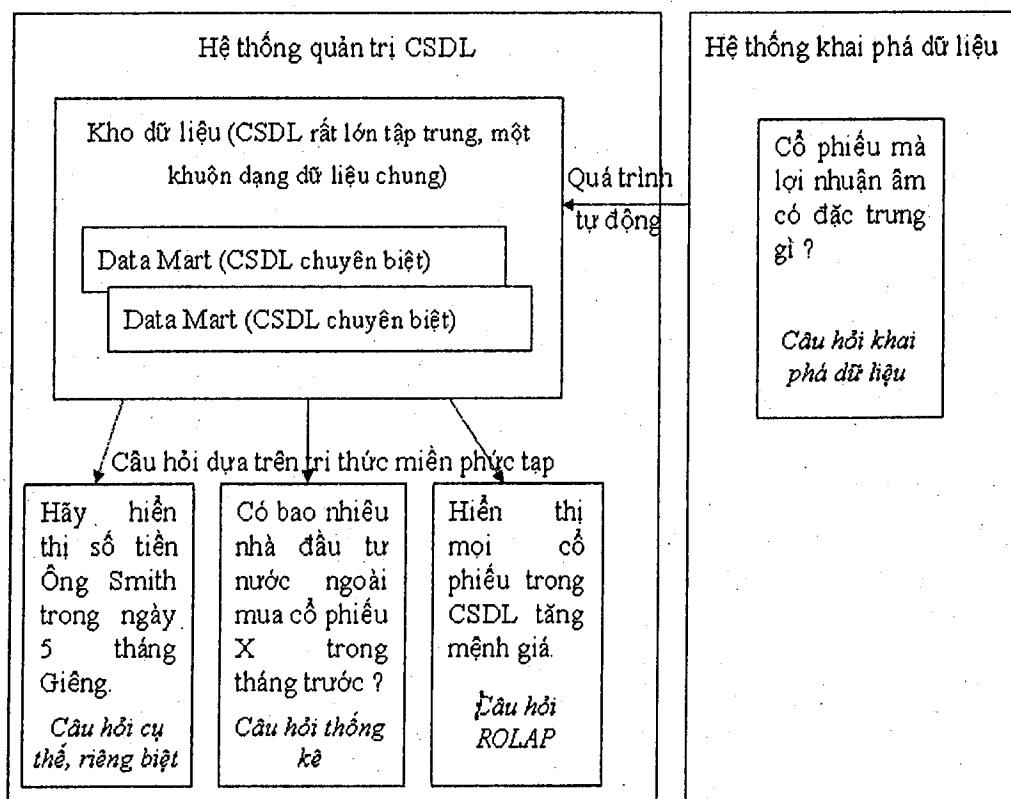
Hình 1.9. Kiến trúc điển hình hệ thống khai phá dữ liệu [HK0106]

Cơ sở tri thức, bao chứa các tri thức miền ứng dụng hiện có, được sử dụng trong thành phần hệ thống khai phá dữ liệu để làm tăng tính hiệu quả của thành phần này. Một số tham số của thuật toán khai phá dữ liệu tương ứng sẽ được tinh chỉnh theo tri thức miền sẵn có từ cơ sở tri thức trong hệ thống. Cơ sở tri thức còn được sử dụng trong việc đánh giá các mẫu đã khai phá được xem chúng có thực sự hấp dẫn hay không, trong đó có việc đối chứng mẫu mới với các tri thức đã có trong cơ sở tri thức. Nếu mẫu khai phá được là thực sự hấp dẫn thì chúng được bổ sung vào cơ sở tri thức để phục vụ cho hoạt động tiếp theo của hệ thống. Như vậy, nguồn tri thức bổ sung vào cơ sở tri thức ở đây không chỉ từ lập luận lôgic theo các hệ toán lôgic để có tri thức mới, không chỉ do con người hiểu biết thêm về thế giới khách

quan để bổ sung vào mà còn là tri thức được phát hiện một cách tự động từ nguồn dữ liệu.

1.3. KHAI PHÁ DỮ LIỆU VÀ XỬ LÝ CSDL TRUYỀN THỐNG

Như đã giới thiệu, khai phá dữ liệu là một thế hệ phát triển mới trong thời gian gần đây của công nghệ CSDL. Điều đó có nghĩa là có mối quan hệ gần gũi giữa bài toán khai phá dữ liệu và bài toán xử lý (tác nghiệp) CSDL truyền thống trong mối liên quan tới một đối tượng chung là CSDL. Tuy nhiên, hai bài toán này cũng có sự phân biệt. Điểm hiệu phân biệt đầu tiên giữa khai phá dữ liệu và xử lý CSDL truyền thống là đối tượng tác động của bài toán khai phá dữ liệu phải là các CSDL, các kho dữ liệu có dung lượng rất lớn, trong khi đó bài toán tác nghiệp CSDL truyền thống liên quan tới các CSDL với mọi kích thước. Thêm nữa, những nội dung dưới đây cung cấp thêm các thông tin bổ sung về bài toán khai phá dữ liệu [KV01]. Mối quan hệ giữa hệ thống quản trị CSDL với hệ thống khai phá dữ liệu được mô tả trong Hình 1.10 [KV01].



Hình 1.10. Mối quan hệ giữa hệ thống CSDL và hệ thống khai phá dữ liệu [KV01]

Hệ quản trị CSDL truyền thống được định hướng việc tìm kiếm tới:

- *Ghi nhận riêng lẻ*, chẳng hạn như cần tìm kiếm câu trả lời cho truy vấn "Hãy hiển thị số tiền của Ông Nguyễn Văn A có trong ngày 5 tháng Giêng năm nay". Việc tìm kiếm các ghi nhận riêng lẻ thường được chỉ dẫn là xử lý giao dịch trực tuyến (on-line transaction processing - OLTP).

- *Ghi nhận thống kê*, chẳng hạn như để trả lời câu hỏi "Có bao nhiêu nhà đầu tư nước ngoài mua cổ phiếu X trong tháng trước?". Việc tìm kiếm ghi nhận thống kê thường được chỉ dẫn là hệ thống hỗ trợ quyết định thống kê (stastical decision support system - DSS).

- *Ghi nhận về dữ liệu đa chiều*, chẳng hạn như để đáp ứng yêu cầu "Hiển thị mọi cổ phiếu trong CSDL với mệnh giá tăng". Việc tìm kiếm các ghi nhận dữ liệu đa chiều thường được hiểu là cung cấp xử lý phân tích trực tuyến (on-line analytic processing - OLAP) và xử lý phân tích trực tuyến quan hệ (relational OLAP - ROLAP).

Để các loại truy vấn (như những truy vấn nói trên) đặt ra được vấn đề cần giải quyết một cách đúng đắn, và qua đó tạo ra được các quyết định hữu ích thì cần phải công nhận đã tồn tại **một giả thiết về tri thức miền phức hợp "đầy đủ"** (sophisticated domain knowledge) mà các loại truy vấn nói trên được đưa ra dựa trên cơ sở tri thức miền đó. Trong CSDL quan hệ thì tập ràng buộc, điển hình là tập phụ thuộc hàm cùng các luật suy diễn Armstrong là một bộ phận của tri thức miền ứng dụng nói trên. Tuy nhiên, với các CSDL lớn có dung lượng tới hàng trăm Gigabytes (GB) thì rất khó khăn để công nhận một tri thức miền phức hợp đầy đủ.

Về mục tiêu của hệ thống, phương pháp khai phá dữ liệu hỗ trợ việc mở rộng mục tiêu của CSDL truyền thống bằng cách cho phép tìm kiếm các câu trả lời cho các truy vấn tuy thô sơ song lại quan trọng, có tác dụng cải tiến miền tri thức (trong trường hợp này tri thức miền phức hợp được coi là *chưa đầy đủ*) như:

- Các cổ phiếu tăng giá có đặc trưng gì?
- Tỷ giá USD - DMark có đặc trưng gì?
- Hy vọng gì về cổ phiếu X trong tuần tiếp theo?
- Trong tháng tiếp theo, sẽ có bao nhiêu đoàn viên công đoàn không trả được nợ của họ?
- Những người mua sản phẩm Y có đặc trưng gì?
- Tôi nên mua loại ô tô nào?
- Tôi nên vào trường đại học nào?
- Những bài báo nền tảng về chủ đề nghiên cứu sinh của tôi là những bài báo nào?
- v.v.

Trả lời các truy vấn này dường như là chúng ta đã khám phá ra được các quy tắc (luật) tiềm ẩn trong dữ liệu và trên cơ sở các quy tắc đó mà đưa ra được các dự báo. Như vậy mục tiêu của khai phá dữ liệu là cung cấp thông tin, tri thức hỗ trợ quyết định thông qua các mẫu, các luật được khám phá. Các mẫu (luật) được khám phá là *không tuyệt đối, không mang tính "bất di bất dịch"* mà có tính chất "*đa số trường hợp là đúng*" và có thể thay đổi từ thời điểm này đến thời điểm khác. Chẳng hạn như luật kết hợp "có đến 80% người nếu đã mua bia thì cũng mua thêm tã trẻ em" được phát hiện cho thấy tại thời điểm đang xem xét phần đông người mua bia thì cũng mua thêm tã trẻ em. Phát hiện này được giải thích như sau. Tại một số vùng ở phương Tây, người chồng thường được "phân công" trông con nhỏ trong khi người vợ làm việc gia đình. Để người chồng "thuận tiện nhất" khi trông con trẻ thì bé được đóng bỉm trẻ em còn người chồng ngồi uống bia và chơi với con. Có thể đến thời điểm nào đó khác trong tương lai của các vùng dân cư nói trên hoặc tại các vùng dân cư khác, khi mà thị hiếu của người đàn ông trông trẻ có sự thay đổi, theo đó họ sẽ không mua bia nữa thì trong cơ sở dữ liệu giao dịch sẽ không tiềm ẩn "luật" nói trên nữa.

Như vậy, trong khai phá dữ liệu thì giả thiết **đã biết** về một tri thức miền phức tạp "*đầy đủ*" không còn là yếu tố **cốt lõi**, và quá trình phát hiện tri thức có tác dụng bổ sung thêm

các tri thức "mới" vào miền tri thức đó. Tính chất không đầy đủ của tri thức miền cho phép tri thức miền có thể có sẵn, có thể được bổ sung, thay đổi nhờ quá trình phát hiện tri thức từ dữ liệu.

1.4. MỘT SỐ LĨNH VỰC ỨNG DỤNG KHAI PHÁ DỮ LIỆU ĐIỂN HÌNH

Theo J. Han và M. Kamber [HK0106], ứng dụng của KDD được chia thành hai lớp chính bao gồm lớp các ứng dụng phân tích dữ liệu - hỗ trợ quyết định và lớp các lĩnh vực ứng dụng khác.

Lớp các ứng dụng trong *phân tích dữ liệu và hỗ trợ quyết định* bao gồm các ứng dụng trong phân tích và quản lý thị trường, phân tích và quản lý rủi ro, khám phá ngoại lai và các mảng không hữu ích. Dữ liệu trong các ứng dụng này là khá phong phú có được từ các giao dịch thẻ tín dụng, nghiên cứu đời sống cộng đồng...

Một số mục tiêu khai phá dữ liệu như là tìm ra các nhóm khách hàng định hướng tiếp thị dựa trên các đặc trưng về niềm hứng thú, mức thu nhập... cũng như phân tích thị trường chéo như tìm ra các mối liên kết, đồng quan hệ trong việc bán hàng để dự báo theo các kết hợp đó.

Một số ứng dụng điển hình nhất là phân tích hướng khách hàng theo từng loại sản phẩm để định hướng tiếp thị phù hợp, phân tích nhu cầu khách hàng, định danh loại sản phẩm thích hợp cho từng lớp khách hàng để đưa ra chiến lược kinh doanh đối với nhóm khách hàng mới, đưa ra các báo cáo tóm tắt đa chiều cũng như những thông tin tóm tắt về mặt thống kê...

Ngoài ra, ứng dụng trong lập kế hoạch tài chính và đánh giá lưu lượng tiền tệ... trong tài chính – ngân hàng cũng được phát triển. Trong công tác lập kế hoạch tài nguyên cũng đã xuất hiện nhiều ứng dụng của KDD. Hơn nữa, đã có nhiều cách tiếp cận khác nhau nhằm phát hiện tri thức đã được sử dụng trong các ứng dụng như vậy.

Trong nhóm phân tích dữ liệu và hỗ trợ quyết định, KDD còn được ứng dụng khá rộng rãi trong lĩnh vực bảo hiểm y tế, phục vụ thẻ tín dụng, viễn thông, thể thao, chính phủ vũ trụ.

Lớp các lĩnh vực *ứng dụng điển hình khác* bao gồm khai phá Text, khai phá Web, khai phá dữ liệu dòng, khai phá dữ liệu sinh học... Một số sản phẩm điển hình về khai phá Text và khai phá Web đã được khẳng định được tính hiệu quả, chẳng hạn các sản phẩm TextAnalyst*, TextracterTM, WebAnalyst và PolyAnalyst... của công ty Megaputer⁹, hoặc WebFountain của IBM...

Sự phát triển nhanh chóng của khai phá dữ liệu làm cho miền ứng dụng lĩnh vực ngày càng thêm phong phú và đa dạng, chẳng hạn quan niệm của J. Han và M. Kamber về các khu vực ứng dụng khai phá dữ liệu đã có sự thay đổi từ phiên bản 2001 tới phiên bản 2006 [HK0106]. Trong phiên bản 2006, J. Han và M. Kamber coi rằng các lĩnh vực điển hình của khai phá dữ liệu là phân tích dữ liệu tài chính, công nghiệp bán lẻ, công nghiệp truyền thông, phân tích dữ liệu sinh học, ứng dụng các ngành khoa học khác, sự xâm nhập sai trái...

Còn theo Gregory Piatetsky-Shapiro [Pia06], các miền ứng dụng điển hình của khai phá dữ liệu là:

- Ứng dụng trong khoa học như thiên văn học, tin sinh học, y học (sáng chế các dược phẩm)...
- Ứng dụng trong thương mại như quản lý quan hệ khách hàng (Customer Relationship Management: CRM), phát hiện gian lận, thương mại điện tử, sản xuất, thể thao/giải trí, dịch vụ viễn thông, tiếp thị định hướng, bảo hiểm y tế...
- Ứng dụng trong World Wide Web như máy tìm kiếm, quảng cáo trực tuyến, khai phá web và khai phá text...
- Ứng dụng trong hoạt động chính quyền như phát hiện tội phạm, phát hiện lừa đảo thuế thu nhập cá nhân...

⁹ <http://www.megaputer.com/>

Bảng 1.2. Các ứng dụng khai phá dữ liệu nổi bật (Số trong ngoặc là số người bỏ phiếu từng năm: một người có thể làm nhiều ứng dụng)

Loại ứng dụng	2006 (111)	2007 (138)	2008 (107)	2009 (180)	2010 (213)
CRM/ consumer analytics	39.1	26.1	38.3	32.8	26.8
Banking	0.9	23.9	31.8	24.4	19.2
Health care/ HR	4.5	7.2	9.3	11.7	13.1
Fraud Detection	21.8	18.8	19.6	13.9	12.7
Other	13.6	13.0	13.1	7.8	11.7
Finance	×	7.2	16.8	11.1	11.3
Direct Marketing/ Fundraising	20.0	20.3	14.0	16.1	11.3
Telecom / Cable	12.7	15.2	12.1	14.4	10.8
Insurance	10.9	8.7	10.3	10.0	10.3
Science	10.9	18.8	10.3	10.6	10.3
Education	×	×	×	4.4	9.9
Advertising	×	×	12.1	10.6	9.9
Web usage mining	10.9	10.1	7.5	8.3	8.9
Manufacturing	6.4	6.5	8.4	3.3	8.0
Medical/ Pharma	7.3	9.4	7.5	7.8	8.0
Retail	10.0	10.1	12.1	11.7	8.0
Credit Scoring	19.1	18.8	13.1	15.6	8.0
e-Commerce	5.6	5.8	7.5	10.0	7.0
Search/Web content mining	13.6	6.5	5.6	6.7	6.6
Social Networks	×	×	1.9	7.8	6.6
Government/Military	6.4	7.2	3.7	8.9	6.1
Investment / Stocks	10.0	2.9	13.1	6.7	5.6
Biotech/Genomics	15.5	11.6	11.2	7.8	5.6
Entertainment/ Music	1.8	4.3	2.8	1.7	3.3
Security / Anti-terrorism	4.5	3.6	5.6	5.0	1.9
Travel / Hospitality	4.5	2.2	2.8	2.8	1.4
Junk email / Anti-spam	1.8	2.2	2.8	0.6	0.9
Social Policy/Survey analysis	×	3.6	7.5	1.7	0.9
None	×	×	1.9	×	×

Khai phá dữ liệu là lĩnh vực ứng dụng có sự phát triển nhanh, thích hợp với sự phát triển xu thế "bùng nổ dữ liệu" và xu thế biến động của nhu cầu xã hội. Chẳng hạn, trong thời gian gần đây, mạng xã hội (social network) và phương tiện xã hội (social

media) đã trở thành một trong những lĩnh vực ứng dụng nổi bật của khai phá dữ liệu. Bảng 1.2 cho biết tình hình về các ứng dụng khai phá dữ liệu nổi bật qua thăm dò tại trang web KDnuggets¹⁰. Khuynh hướng phát triển ứng dụng của khai phá dữ liệu sẽ được trình bày tại Chương 10.

1.5. KIỂU DỮ LIỆU TRONG KHAI PHÁ DỮ LIỆU

Bảng 1.3 cho biết tình hình về các kiểu dữ liệu được khai phá qua thăm dò tại trang web KDnuggets¹¹. Về nguyên lý chung, nguồn dữ liệu được sử dụng để tiến hành khai phá dữ liệu nhằm phát hiện tri thức là rất phong phú và đa dạng, trong đó điển hình nhất là CSDL quan hệ, kho dữ liệu, CSDL giao dịch, các hệ thống dữ liệu và thông tin mở rộng khác.

1.5.1. Cơ sở dữ liệu quan hệ

Thứ nhất, tính phổ biến của hệ thống CSDL quan hệ hiện nay tạo ra một hệ quả tự nhiên quy định CSDL quan hệ là một nguồn đầu vào điển hình nhất, được quan tâm trước hết của khai phá dữ liệu. Thứ hai, một trong những mẫu được quan tâm là mẫu về các loại "quan hệ" mà với bản chất của mình, hệ thống CSDL quan hệ tiềm ẩn các mẫu dạng như thế. Như đã biết trong lý thuyết CSDL, hệ thống CSDL quan hệ thường bao gồm một tập các bảng (hai chiều dọc và ngang). Theo chiều dọc, bảng gồm một số cột (còn được gọi là thuộc tính, trường hay đặc trưng) và theo chiều ngang bảng chứa một tập rất lớn các dòng (còn được gọi là bản ghi hay bộ). Số lượng cột của bảng còn được gọi là số chiều. Hệ thống CSDL quan hệ còn bao gồm một mô hình ngữ nghĩa mà thông thường là mô hình thực thể - quan hệ.

¹⁰ <http://www.kdnuggets.com/polls/>

¹¹ <http://www.kdnuggets.com/polls/>

Bảng 1.3. Kiểu dữ liệu được khai phá (Số trong ngoặc là số người bỏ phiếu từng năm: một người có thể dùng nhiều kiểu dữ liệu)

Loại dữ liệu	2006 (106)	2007 (120)	2008 (108)	2009 (95)	2010 (144)	2011 (206)
table data (fixed n. columns)	70.8	85.8	77.8	80.0	70.8	69.4
time series	34.0	38.3	37.0	45.3	38.9	41.7
itemsets / transactions	28.3	21.7		28.4	36.1	32.5
text (free-form)	33.0	34.2	36.1	37.9	29.9	25.7
anonymized data	25.5	18.3	17.6	18.9	26.4	21.8
location/geo/mobile data	x	x	x	x	x	19.4
other	14.2	10.0	1.9	9.5	15.3	14.1
social network data	8.5	9.2	8.3	12.6	19.4	12.6
email	10.4	11.7	5.6	8.4	10.4	10.7
web content	5.7	14.2	10.2	13.7	13.2	10.2
web clickstream	8.5	11.7	5.6	8.4	10.4	8.7
images / video	4.7	7.5	5.6	12.6	7.6	6.8
XML data	6.6	10.8	5.6	14.7	11.8	4.9
music / audio	7.5	4.2	0.9	7.4	2.1	3.4
spatial data 2D 3D	14.2	13.3	13.0	9.5	x	x

1.5.2. Kho dữ liệu

Theo J. Han và M. Kamber, tồn tại nhiều cách hiểu về kho dữ liệu, nhưng cách hiểu phổ dụng nhất là theo định nghĩa của W.H. Inmon, một chuyên gia hàng đầu về kho dữ liệu. Theo W.H. Inmon [Inm02], "kho dữ liệu là tập hợp các dữ liệu *định hướng theo chủ đề, được tích hợp lại, có tính phiên bản theo thời gian và kiên định* được dùng để hỗ trợ việc tạo quyết định quản lý". Tên gọi của bốn thuộc tính "*định hướng theo chủ đề*", "*được tích hợp lại*", "*có tính phiên bản theo thời gian*" và "*kiên định*" trên đây của kho dữ liệu mới chỉ cung cấp một số nét cơ bản nhất về các đặc trưng của kho dữ liệu. W.H. Inmon (cũng như J. Han và M. Kamber) đã giải thích nội dung chi tiết về bốn thuộc tính này.

Kho dữ liệu là một kết quả xuất hiện trong quá trình tiến hóa các hệ hỗ trợ quyết định. Thuật ngữ "tạo kho dữ liệu" (Data warehousing) được dùng để chỉ quá trình xây dựng và sử dụng kho dữ liệu. Như vậy, quá trình phát hiện tri thức trong CSDL tiếp nhận đầu vào là các hệ thống CSDL, "*các kho dữ liệu tích hợp dữ liệu từ các nguồn và các dữ liệu mô tả*". Cần chú ý rằng, để đáp ứng bốn thuộc tính trên đây kho dữ liệu được coi chỉ bao gồm các dữ liệu được coi là "có chất lượng" thông qua các khâu chọn lựa, tiền xử lý và có thể bao gồm cả khâu chuyển dạng trong quá trình phát hiện tri thức trong CSDL (Hình 1.4).

Các nghiên cứu và triển khai liên quan tới kho dữ liệu chỉ dẫn khuynh hướng hiện tại của các hệ thống thông tin quản lý (MIS: Management Information Systems) phổ biến là nhằm vào việc thu thập, làm sạch dữ liệu giao dịch và tạo cho chúng độ linh hoạt khi tìm kiếm trực tuyến. Một tiệm cận phổ biến đối với phân tích kho dữ liệu gọi là OLAP (On-Line Analytical Processing), thông qua một tập các nguyên lý được Codd đề xuất vào năm 1993. Các bộ công cụ OLAP chú trọng tới việc cung cấp tới SQL các tiện ích phân tích dữ liệu đa chiều chất lượng cao bằng các tính toán giản lược và phân tách nhiều chiều. Cả phát hiện tri thức lẫn OLAP được coi là hai khía cạnh quan hệ mật thiết nhau được tích hợp trong một thế hệ mới các bộ công cụ trích lọc và quản lý thông tin.

Đồng thời với sự phát triển của công nghệ kho dữ liệu, các hệ thống tích hợp các nguồn dữ liệu cả dữ liệu trong quá khứ lẫn dữ liệu tác nghiệp đã được xây dựng. Nhiều hệ thống khai phá dữ liệu có đầu vào từ siêu dữ liệu (metadata) cùng các dữ liệu nguồn trong các kho dữ liệu.

1.5.3. Cơ sở dữ liệu giao dịch

Một lớp bài toán khai phá dữ liệu phổ biến là khai phá quan hệ kết hợp, trong đó điển hình là bài toán khai phá *luật kết hợp*, được xuất phát từ việc xem xét các CSDL giao dịch (bán hàng). Dữ liệu giao dịch chính là dữ liệu nguyên thủy xuất hiện trong định nghĩa về luật kết hợp cùng với các độ đo của luật như độ hỗ trợ và độ tin cậy. Khi mở rộng dữ liệu từ dữ liệu giao dịch sang dữ liệu vô hướng hoặc dữ liệu phức tạp hơn có trong các CSDL quan hệ, các giải pháp khai phá luật kết hợp được cải tiến để thích ứng với sự biến đổi này (bao gồm bước chuyển dạng dữ liệu trong quá trình phát hiện tri thức từ các CSDL).

1.5.4. Các hệ thống dữ liệu mở rộng

Trong quá trình phát triển, các phương pháp và thuật toán khai phá dữ liệu thích hợp đối với các CSDL mở rộng và các kiểu kho chứa dữ liệu được đề xuất. Các phương pháp và thuật toán này

được phù hợp với dữ liệu trong CSDL hướng đối tượng, CSDL không gian-thời gian, CSDL tạm thời, dữ liệu chuỗi thời gian (bao gồm dữ liệu tài chính), dữ liệu dòng, CSDL Text và CSDL đa phương tiện, CSDL hỗn tạp và CSDL thừa kế, và World Wide Web.

Hệ thống CSDL quan hệ - đối tượng có thể được coi là sự bổ sung theo tiếp cận hướng đối tượng tới các hệ thống CSDL quan hệ. Mô hình dữ liệu quan hệ - đối tượng mô tả ngũ nghĩa của hệ thống CSDL quan hệ - đối tượng, được phát triển từ mô hình quan hệ với việc bổ sung các kiểu dữ liệu giàu ngũ nghĩa. Thực thể từ mô hình quan hệ thực thể được phát triển thành đối tượng trong mô hình quan hệ đối tượng. Để khai phá dữ liệu đối với CSDL quan hệ - đối tượng.

1.6. CÁC BÀI TOÁN KHAI PHÁ DỮ LIỆU ĐIỂN HÌNH

Khai phá dữ liệu là lĩnh vực nghiên cứu mang tính thực tiễn cao, đồng thời lại đòi hỏi một nền tảng toán học mạnh trong việc xây dựng các mô hình toán học phù hợp nhất cho miền dữ liệu của bài toán đang được quan tâm. Bước khai phá dữ liệu trong quá trình KDD thường áp dụng một phương pháp khai phá dữ liệu cụ thể, liên quan đến các khái niệm mẫu và mô hình. Như đã được giới thiệu trong mục 1.1, mẫu là một biểu thức trong một ngôn ngữ mô tả L nào đó được chọn. Mô hình được coi là một biểu thức tổng quát trong ngôn ngữ mô tả L nói trên; tính tổng quát của mô hình được thể hiện thông qua các tham số mô hình, trong trường hợp đó, một mẫu là một thể hiện của mô hình. Chẳng hạn, biểu thức $a \times 2 + bx$ (với hai tham số a và b) là mô hình còn $3 \times 2 + x$ là một mẫu trong mô hình đó (đối với mẫu này thì các tham số mô hình a và b đã được cho giá trị cụ thể, $a = 3$ và $b = 1$).

Nhiệm vụ của bài toán khai phá dữ liệu là từ một tập dữ liệu quan sát (tập các sự kiện) đã có thì hoặc cần phải xác định mô hình phù hợp với tập dữ liệu quan sát đó, hoặc cần tìm ra các mẫu từ tập dữ liệu đó.

Bài toán khai phá dữ liệu thường hướng tới một trong hai loại mô hình đó là *mô hình theo tiếp cận thống kê* (mô hình thống kê)

hoặc *mô hình lôgic*. Mô hình thống kê được định hướng tới loại mô hình bao hàm các yếu tố chưa xác định, chẳng hạn như mô hình $ax + e$, trong mô hình này thì x là biến trong ngôn ngữ mô tả L, còn e có thể là biến ngẫu nhiên Gauss (thể hiện tính chưa xác định của mô hình). Ngược lại, mô hình lôgic định hướng tới loại mô hình xác định hoàn toàn, chẳng hạn ax , trong đó không thừa nhận yếu tố không rõ ràng khi mô hình hóa. Mô hình thống kê được dùng hầu khắp đối với các ứng dụng khai phá dữ liệu thực tế.

Hầu hết các phương pháp khai phá dữ liệu đã được xây dựng có nội dung từ các phương pháp học máy, thiết kế mẫu và thống kê (phân lớp, phân đoạn, mô hình đồ thị...). Thuật toán giải quyết mỗi bài toán nói trên cuốn hút một phạm vi người quan tâm đa dạng bao gồm cả các chuyên gia phân tích dữ liệu lẫn những người chưa hề có kinh nghiệm.

Ở mức cao - tổng quát, hai mục tiêu chủ yếu của khai phá dữ liệu là dự báo và mô tả, mà chúng ta coi hai mục tiêu này tương ứng với hai bài toán tổng quát của khai phá dữ liệu. Bài toán dự báo sử dụng một số biến (hoặc trường) trong CSDL để dự đoán về hoặc giá trị chưa biết (dù đã có) hoặc giá trị sẽ có trong tương lai của các biến. Bài toán mô tả hướng tới việc tìm ra các mẫu mô tả dữ liệu. Dự đoán và mô tả có tầm quan trọng khác nhau đối với các thuật toán khai phá dữ liệu riêng. Trong ngữ cảnh KDD thì vấn đề mô tả có khuynh hướng quan trọng hơn vấn đề dự báo, và điều này là trái ngược với nội dung chủ yếu của các ứng dụng nhận dạng mẫu và học máy thì vấn đề dự báo là quan trọng hơn. Điều có vẻ trái ngược đó có thể được giải thích khi xem xét, phân tích nội dung của chính khái niệm "phát hiện tri thức trong CSDL"; khái niệm này đã bao hàm tình huống sẵn có dữ liệu để phát hiện các mẫu tiềm ẩn trong dữ liệu đó, các mẫu tiềm ẩn đó liên quan tới bài toán mô tả dữ liệu. Mặt khác, mô tả được mô hình dữ liệu thì cũng rất thuận tiện cho dự báo.

Ở mức chi tiết - cụ thể, dự báo và mô tả được thể hiện thông qua các bài toán cụ thể như mô tả khái niệm, quan hệ kết hợp, phân cụm, phân lớp, hồi quy, mô hình phụ thuộc, phát hiện biến đổi và độ lệch và một số bài toán cụ thể khác như trình bày dưới đây.

1.6.1. Mô tả khái niệm

Nội dung của bài toán mô tả khái niệm (concept description) là tìm ra các đặc trưng và tính chất của khái niệm để "mô tả" khái niệm đó. Điển hình nhất trong lớp bài toán này là các bài toán như tổng quát hóa, tóm tắt, phát hiện các đặc trưng dữ liệu ràng buộc.

Bài toán tóm tắt là một bài toán mô tả điển hình, áp dụng các phương pháp để tìm ra một mô tả cô đọng đối với một tập con dữ liệu. Một ví dụ điển hình về bài toán tóm tắt là bài toán tính kỳ vọng và độ lệch chuẩn của một tập dữ liệu trong thống kê xác suất; hai giá trị này chính là hai đặc trưng điển hình nhất về một hiện tượng có dãy giá trị thể hiện mà chúng ta đã quan sát được.

Nhiều phương pháp đã được biện luận đòi hỏi việc thu nhận được các quy tắc tóm tắt, kỹ thuật hiển thị đa biến, phát hiện quan hệ hàm giữa các biến. Kỹ thuật tóm tắt thường được áp dụng trong phân tích dữ liệu thăm dò có tương quan và tự động hóa sinh ra các thông báo.

Trong khai phá Text và khai phá Web, tóm tắt văn bản là một biểu hiện cụ thể của tóm tắt, theo đó từ một văn bản đã có, cần tìm ra văn bản ngắn gọn (với độ dài 100 từ, 200 từ hoặc 500 từ) mà vẫn giữ được ngữ nghĩa cơ bản của văn bản gốc.

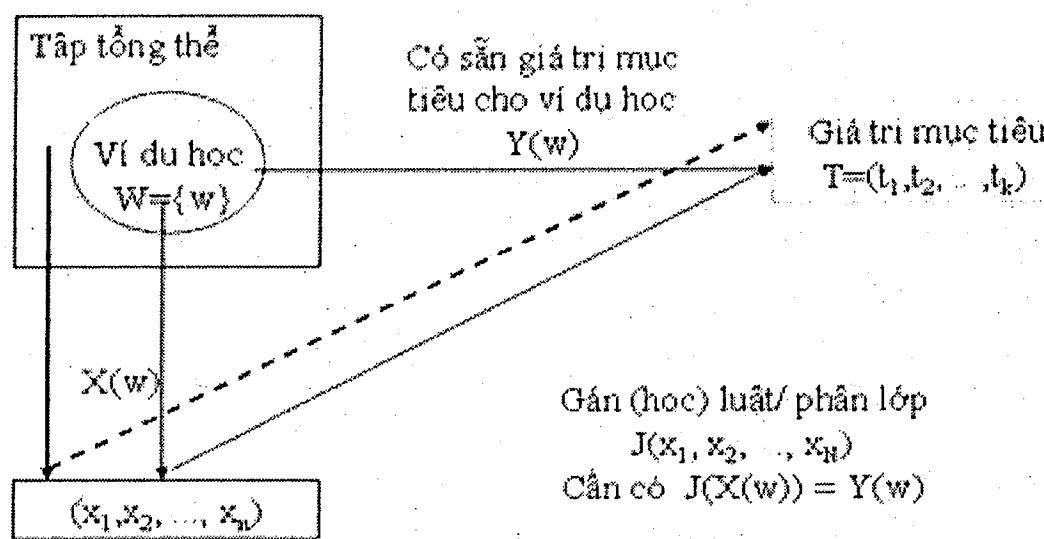
1.6.2. Quan hệ kết hợp

Phát hiện mối quan hệ kết hợp (associative relation) trong tập dữ liệu là một bài toán quan trọng trong khai phá dữ liệu. Một trong những mối quan hệ kết hợp điển hình là quan hệ kết hợp giữa các biến dữ liệu, trong đó bài toán khai phá luật kết hợp (associative rule) là một bài toán điển hình. Bài toán khai phá luật kết hợp (thuộc lớp phát hiện quan hệ kết hợp), thực hiện việc phát hiện ra mối quan hệ giữa các tập thuộc tính (các tập biến) có dạng $X \rightarrow Y$, trong đó X, Y là hai tập thuộc tính. Về hình thức, luật kết hợp có dạng giống như phụ thuộc hàm trong CSDL quan hệ, tuy nhiên, nó không được định sẵn từ tri thức miền.

Trong khai phá text và khai phá web tồn tại nhiều bài toán phát hiện quan hệ kết hợp, điển hình như bài toán phát hiện quan hệ ngữ nghĩa (chẳng hạn như quan hệ nhân-quả, quan hệ toàn bộ - bộ phận, quan hệ chung-riêng...) trong văn bản (hoặc trong tập văn bản), bài toán phát hiện mối quan hệ giữa nội dung trang web người sử dụng đang quan tâm tới các trang web mà họ có thể sẽ hướng tới...

1.6.3. Phân lớp

Phân lớp (Classification/Categorization) thực hiện việc xây dựng (mô tả) các mô hình (hàm) dự báo nhằm mô tả hoặc phát hiện các lớp hoặc khái niệm cho các dự báo tiếp theo. Một số phương pháp điển hình là cây quyết định, luật phân lớp, mạng neuron. Nội dung của phân lớp chính là học một hàm ánh xạ các dữ liệu vào một trong một số lớp đã biết. Ví dụ, phân lớp một văn bản (bao gồm cả trang web) vào một trong một số lớp văn bản (trang web) đã biết, phân lớp khuynh hướng trong thị trường tài chính, phát hiện tự động các đối tượng đáng quan tâm trong CSDL ảnh lớn.



Hình 1.11. Sơ đồ biểu diễn mô hình học máy: cần học ánh xạ biểu diễn bằng đường liên nét xiên [KV01] (Lưu ý, học không giám sát không có giá trị mục tiêu cho ví dụ học: không có đường liên nét)

Hình 1.11 mô tả sơ bộ về bài toán phân lớp (thường được tương ứng với học có giám sát), theo đó đường ngang liền nét cho biết đã biết thuộc tính lớp đối với một tập hợp dữ liệu nào đó (tập dữ liệu học). Nội dung chi tiết hơn về bài toán phân lớp sẽ được trình bày chi tiết hơn trong các chương sau.

1.6.4. Phân cụm

Phân cụm (Clustering) thực hiện việc nhóm dữ liệu thành các "cụm" (có thể coi là các lớp mới) để có thể phát hiện được các mẫu phân bố dữ liệu trong miền ứng dụng. Phân cụm là một bài toán mô tả hướng tới việc nhận biết một tập hữu hạn các cụm hoặc các lớp để mô tả dữ liệu. Các cụm (lớp) có thể tách rời nhau và toàn phần (tạo nên một phân hoạch cho tập dữ liệu) hoặc được trình bày đẹp hơn như phân lớp có thứ bậc hoặc có thể chồng lên nhau (giao nhau). Ví dụ như bài toán phát hiện các nhóm người tiêu dùng trong CSDL tiếp thị hoặc nhận biết các loại quang phổ trong tập phép đo không gian hồng ngoại... Thông thường, mục tiêu định hướng của bài toán phân cụm là cực đại tính tương đồng giữa các phần tử trong mỗi cụm và cực tiểu tính tương đồng giữa các phần tử thuộc các cụm khác nhau.

Trong nhiều trường hợp, phân cụm còn được gọi là học máy không giám sát (unsupervised learning) và phân lớp còn được gọi là học máy giám sát (supervised learning). Sơ bộ về mô hình học máy (có giám sát và không giám sát) được diễn tả như tại Hình 1.11 [KV01]. Tuy cùng sử dụng học máy như phân lớp thuộc loại khai phá dữ liệu dự báo còn phân cụm thuộc loại khai phá dữ liệu mô tả.

Trong một số ứng dụng, bài toán phân đoạn (segmentation) cần được giải quyết. Về nội dung, phân đoạn là tổ hợp của phân cụm và phân lớp, trong đó phân cụm được tiến hành trước và sau đó là phân lớp.

1.6.5. Hồi quy

Hồi quy (regression) là một bài toán điển hình trong phân tích thống kê và dự báo, trong đó tiến hành việc dự đoán các giá trị của

một hoặc một số biến phụ thuộc vào giá trị của một tập hợp các biến độc lập. Mô hình hồi quy là khá thông dụng trong dự báo dài hạn. Trong khai phá dữ liệu, bài toán hồi quy được quy về việc học một hàm ánh xạ dữ liệu nhằm xác định giá trị thực của một biến theo một số biến khác. Tình huống ứng dụng hồi quy rất đa dạng, chẳng hạn như dự đoán số lượng sinh vật phát quang trong khu rừng nhờ đo vi sóng các sensor từ xa, hoặc ước lượng xác suất người bệnh có thể chết theo kết quả test triệu chứng, hoặc dự báo nhu cầu người tiêu dùng đối với một sản phẩm mới được coi như một hàm của quảng cáo tiêu dùng, hoặc dự báo chuỗi thời gian mà các biến đầu vào được coi như bản trễ thời gian của biến dự báo...

1.6.6. Mô hình phụ thuộc

Bài toán xây dựng mô hình phụ thuộc hướng tới việc tìm ra một mô hình mô tả sự phụ thuộc có ý nghĩa giữa các biến. Mô hình phụ thuộc gồm hai mức: mức cấu trúc của mô hình mô tả (thường dưới dạng đồ thị) trong đó các biến là phụ thuộc bộ phận vào các biến khác, trong khi mức định lượng của mô hình mô tả sức mạnh của tính phụ thuộc khi sử dụng việc đo tính theo giá trị số. Ví dụ, lưới phụ thuộc xác suất cần đảm bảo tính độc lập điều kiện nhằm định rõ diện mạo cấu trúc của mô hình và xác suất hoặc tương quan để mô tả sức mạnh của tính phụ thuộc. Phân tích khuynh hướng và tiến hóa cũng được coi thuộc vào loại khai phá mô hình phụ thuộc. Trong phân tích khuynh hướng và tiến hóa, các phương pháp phân tích xu thế, khai phá mẫu kế tiếp, phân tích dựa trên tính tương tự... thường được áp dụng.

1.6.7. Phát hiện biến đổi và độ lệch

Tập trung vào việc phát hiện hầu hết sự thay đổi có ý nghĩa dưới dạng độ đo đã biết trước hoặc giá trị chuẩn, cung cấp những tri thức về sự biến đổi và độ lệch cho người dùng. Bài toán phát hiện biến đổi và độ lệch còn được ứng dụng trong bước tiền xử lý trong quá trình phát hiện tri thức trong CSDL. Chính vì lý do đó, cần tránh suy nghĩ cho rằng sự biến đổi và độ lệch mang ý nghĩa

"không chính quy" mà phải quan niệm sự biến đổi và độ lệch đó (có thể là bất thường) là một nội dung bản chất của dữ liệu.

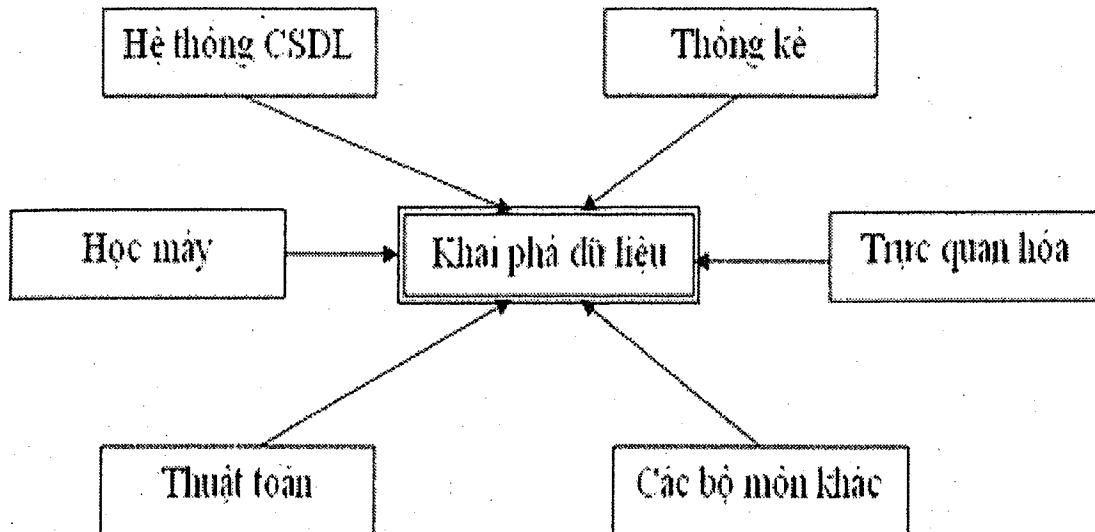
Ngoài ra có thể kể tới bài toán phân tích định hướng mẫu và một số bài toán khai phá dữ liệu kiểu thống kê khác.

1.7. TÍNH LIÊN NGÀNH CỦA KHAI PHÁ DỮ LIỆU

KDD nhận được sự quan tâm đặc biệt của các nhà nghiên cứu trong các lĩnh vực học máy, thu nhận mẫu, CSDL, thống kê, trí tuệ nhân tạo, thu nhận tri thức đối với hệ chuyên gia được trình bày trong Hình 1.12 [HK0106]. Hệ thống KDD lôi cuốn các phương pháp, thuật toán và kỹ thuật từ các lĩnh vực rời rạc nhau này. Mục tiêu thống nhất là trích lọc tri thức từ dữ liệu trong ngữ cảnh các CSDL lớn.

ZH Zhou [Zhou03] nhận định rằng khai phá dữ liệu nhận được sự đóng góp của rất nhiều ngành như CSDL, học máy, thống kê, thu hồi thông tin, trực quan hóa dữ liệu, tính toán song song và phân tán.... Ba ngành đóng góp chính là CSDL, học máy, thống kê. Trong khai phá dữ liệu, CSDL đóng góp các kỹ thuật quản lý dữ liệu, học máy đóng góp các kỹ thuật phân tích dữ liệu thực tiễn, và thống kê đóng góp các nền tảng lý thuyết vững chắc. Tác giả ẩn dụ rằng khai phá dữ liệu nếu không có sự đóng góp của CSDL và học máy sẽ như "tìm kim trong đống cỏ", nếu không có sự đóng góp của thống kê sẽ như "xây dựng lâu đài trong không khí".

Một số lập luận được trình bày tại các mục trước (1.2, 1.3) đã chỉ dẫn rằng khai phá dữ liệu là bước phát triển mới của **công nghệ CSDL**, vì vậy nhiều nội dung trong khai phá dữ liệu là gần gũi với CSDL [HK0106]. Đồng thời, một số dấu hiệu phân biệt giữa hệ thống CSDL điều hành tác nghiệp truyền thống với hệ thống khai phá dữ liệu cũng đã được thảo luận; các dấu hiệu điển hình nhất bao gồm quan niệm về một giả thiết săn có một tri thức miền ứng dụng đầy đủ, loại hình các câu hỏi thể hiện mục tiêu của hệ thống và kích thước tập dữ liệu đối tượng khảo sát.



Hình 1.12. Tính đa/liên ngành của khai phá dữ liệu

Tài nguyên dữ liệu đầu vào cho các hệ thống khai phá dữ liệu gồm có các CSDL, các kho dữ liệu và các loại nguồn chứa dữ liệu khác. Chính vì lý do đó, trong không ít trường hợp, lĩnh vực **kho dữ liệu** được coi là một bộ phận của lĩnh vực khai phá dữ liệu và phát hiện tri thức trong CSDL.

Đối với các lĩnh vực **học máy** và **thu nhận mẫu**, sự đan xen với khai phá dữ liệu (và KDD) trải theo các nghiên cứu về lý thuyết và thuật toán đối với các hệ thống trích lọc mẫu và mô hình dữ liệu (chủ yếu đối với các phương pháp khai phá dữ liệu). Các phương pháp học máy giám sát (phân lớp), không giám sát (phân cụm), bán giám sát (phân lớp và phân cụm) đã rất phổ biến trong khai phá dữ liệu, nhằm lựa chọn mô hình và xác định tham số mô hình trong các hệ thống KDD. Trọng tâm của KDD đối với việc mở rộng các lý thuyết và thuật toán học máy hướng tới bài toán tìm ra các mẫu đặc biệt (những mẫu mà trong một số ngữ cảnh còn được gọi là *tri thức hữu dụng* hoặc *hấp dẫn*) trong các tập hợp dữ liệu có dung lượng lớn của thế giới thực. Như vậy, khai phá dữ liệu mở rộng nội dung học máy thông qua các công việc lựa chọn dữ liệu đầu vào, trình diễn mẫu, đánh giá mẫu đầu ra... trong ngữ cảnh miền dữ liệu cần xử lý có dung lượng rất lớn.

Cùng với tiếp cận mô hình lôgic, mô hình thống kê là tiếp cận phổ biến trong các bài toán phát hiện tri thức trong cơ sở dữ liệu, vì vậy, chuyên ngành KDD có rất nhiều điểm chung với chuyên ngành **thống kê**, đặc biệt là phân tích dữ liệu thăm dò (EDA: Exploratory Data Analysis) cũng như dự báo [Fried97, HD03]. Hệ thống KDD thường gắn kết với các thủ tục thống kê đặc biệt đối với mô hình dữ liệu và nắm bắt nhiều trong một khung cảnh phát hiện tri thức tổng thể. Các phương pháp khai phá dữ liệu dựa theo thống kê nhận được sự quan tâm đặc biệt tạo nên lớp phương pháp khai phá dữ liệu rộng lớn dựa trên học máy thống kê. Robert Nisbet và cộng sự [NEM09], Trevor Hastie và cộng sự [HTF09] cung cấp các nội dung khá toàn diện và bổ ích về các phương pháp học máy thống kê và khai phá dữ liệu thống kê. Robert Nisbet và cộng sự trình bày một cách hệ thống quá trình tiến hóa của thống kê toán học, bao gồm cả sự phát triển các nội dung của thống kê toán học tới khai phá dữ liệu thống kê.

Vì khai phá dữ liệu và xử lý dữ liệu thống kê rất gần gũi với nhau và một số nội dung trong xử lý dữ liệu thống kê được tích hợp vào quá trình khai phá dữ liệu, tuy nhiên, cũng cần nêu ra một số khác biệt giữa bài toán thống kê toán học và bài toán khai phá dữ liệu.

Đầu tiên, khai phá dữ liệu khác biệt với phân tích thống kê trong bài toán thống kê toán học về các giả định cơ bản, trong đó phân tích thống kê yêu cầu các điều kiện chặt chẽ về phân bố dữ liệu, về tham số lỗi trong khi đó khai phá dữ liệu không đòi hỏi những giả định như vậy. Trong bài toán khai phá dữ liệu, tri thức miền tương ứng với giả định đòi hỏi của phân tích thống kê là kết quả của công việc tìm hiểu dữ liệu công phu mà không phải là sẵn có theo giả định. Như vậy, phương pháp phân tích thống kê có thể được huy động trong bước “hiểu dữ liệu” của quá trình khai phá dữ liệu.

Thứ hai, mục tiêu của phân tích thống kê là kiểm thử giả thiết hoặc xác định tham số, trong khi đó mục tiêu của khai phá dữ liệu là xác định mô hình dự báo và độ chính xác của mô hình dự báo đó. Cụ thể hơn, trong bài toán phân tích kiểm định giả

thiết thống kê, cho trước một giả thiết thống kê thì công việc cần tiến hành là kiểm tra xem tập hợp toàn bộ các dữ liệu quan sát được có phù hợp với giả thiết thống kê nói trên hay không, hay cũng vậy, giả thiết thống kê có đúng trên toàn bộ dữ liệu quan sát được hay không. Nếu kiểm định cho kết quả không phù hợp có nghĩa là giả thiết thống kê là không đúng trên tập dữ liệu quan sát. Như vậy, tính đúng đắn của giả thiết thống kê được xem xét trên tập dữ liệu quan sát đã có.

Thứ ba, phân tích thống kê coi tập dữ liệu xử lý là phần lấy mẫu của tập dữ liệu toàn cục trong khi khai phá dữ liệu coi tập dữ liệu cần xử lý là toàn bộ dữ liệu thuộc miền ứng dụng. Trong khai phá dữ liệu, mô hình kết quả khai phá dữ liệu là không được xác định trước cần phải phù hợp với tập toàn bộ dữ liệu của miền ứng dụng mà không phải chỉ với tập dữ liệu quan sát được (tập dữ liệu quan sát được chỉ là một bộ phận mà thường là rất nhỏ so với miền dữ liệu của thế giới thực, xem Hình 1.8) do đó cần đảm bảo các tham số mô hình không phụ thuộc vào cách chọn tập dữ liệu học. Chính vì lý do cốt lõi này mà bài toán học khai phá dữ liệu đòi hỏi đáp ứng yêu cầu là tập dữ liệu học cũng như tập dữ liệu kiểm tra cần có tính "đại diện" cho toàn bộ dữ liệu trong miền ứng dụng và hai tập dữ liệu này cần độc lập nhau. Trong một số bài toán khai phá dữ liệu, hai tập dữ liệu này (hoặc tập dữ liệu kiểm tra) được công bố dưới dạng chuẩn.

Thứ tư, phân tích có đòi hỏi khá rõ ràng về kích thước tập dữ liệu mẫu và có tính chất tinh (ổn định), trong khi đó khai phá dữ liệu tiếp cận theo hướng "càng nhiều càng tốt", hơn nữa dữ liệu có thể động. Tiếp theo, khai phá dữ liệu cho phép thi hành lặp để cải thiện mô hình kết quả trong khi đó việc thi hành lặp có thể dẫn tới kết luận sai lầm trong phân tích thống kê.

Cuối cùng, các thuật ngữ dùng trong hai lĩnh vực nghiên cứu này cũng là dấu hiệu phân biệt chúng, chẳng hạn, lĩnh vực khai phá dữ liệu dùng các thuật ngữ *biến ra/biến mục tiêu, thuật toán khai phá dữ liệu, thuộc tính/đặc trưng, bản ghi...* trong khi đó thì lĩnh vực xử lý dữ liệu thống kê dùng các thuật ngữ tương ứng là *biến phụ thuộc, thủ tục thống kê, biến giải thích, quan sát...*

Như đã được trình bày, quá trình phát hiện tri thức làm việc với tập hợp dữ liệu lớn mà trong nhiều trường hợp tập dữ liệu trở nên khổng lồ. Phạm vi tác động to lớn và đa dạng đòi hỏi các **thuật toán** khai phá dữ liệu phải đúng đắn và hiệu quả; chính vì điều đó cho nên rất nhiều thuật toán khai phá dữ liệu đã được đề xuất. ZH Zhou [Zhou03] giới thiệu về bốn thành phần của một thuật toán khai phá dữ liệu là các mô hình và mẫu, các hàm đánh giá, các phương pháp tìm kiếm và tối ưu hóa, và chiến lược quản lý dữ liệu.

Xindong Wu và cộng sự [WKQ08] cung cấp một danh sách gồm mười thuật toán khai phá dữ liệu nổi tiếng nhất, đó là các thuật toán C4.5, *k*-Means, SVM, Apriori, EM, PageRank, AdaBoost, *k*NN, Naive Bayes, và CART. Các tác giả cũng giới thiệu những nội dung cơ bản nhất của mỗi trong mười thuật toán nói trên. Một số nội dung cơ bản nhất của hầu hết các thuật toán trong mười thuật toán này sẽ được giới thiệu trong các chương từ 4-7 của tài liệu này.

Như đã được khẳng định tại các phần trước đây là không phải tất cả các mẫu đều hữu dụng và hệ thống cần đưa ra các tiêu chí để lọc các mẫu được coi là hấp dẫn nhất. Thông thường các hệ thống sử dụng một ngưỡng hấp dẫn cực tiểu cho các mẫu được coi là tri thức, chẳng hạn trong bài toán phát hiện luật kết hợp, người ta chỉ giữ lại các luật vượt qua ngưỡng độ hỗ trợ tối thiểu và độ tin cậy tối thiểu. Ngay cả trong trường hợp đó, không phải mọi “tri thức” được hệ thống coi là “hữu dụng” đều hoàn toàn phù hợp với người sử dụng. Bước **trực quan hóa** trong quá trình KDD hiển thị các tri thức được hệ thống phát hiện một cách trực quan nhất để tạo thuận lợi cho người sử dụng (thông qua tri thức và kinh nghiệm) lựa chọn ra các tri thức thực sự hữu dụng cho mục đích ứng dụng của người sử dụng.

Phát hiện máy với mục tiêu là phát hiện các luật kinh nghiệm từ quan sát và thử nghiệm và **mô hình nhân quả** phát hiện các kết luận của mô hình nhân quả từ dữ liệu là những lĩnh vực nghiên cứu có mối liên hệ với nhau.

Khai phá dữ liệu và phát hiện tri thức từ dữ liệu cũng chứng kiến sự thâm nhập rộng lớn của *lý thuyết tập mờ* (chẳng hạn, [EM03, HP03, STH06] và các công bố khoa học trong dãy hội nghị quốc tế International Conference on Fuzzy Systems and Knowledge Discovery: FSKD¹² và một số hội nghị quốc tế uy tín khác), *lý thuyết tập thô* (chẳng hạn, [Zia94, Ohrn99, SZ00, Li07, NS08, Szczu11] và các công bố khoa học tại chuỗi hội nghị quốc tế "Rough Sets and Knowledge Technology: RSKT¹³") và lý thuyết kết hợp tập mờ – thô [Jenssen11]. Chương 9 trình bày các nội dung chi tiết về khai phá dữ liệu dựa trên lý thuyết tập mờ, tập thô và tập mờ-thô.

Khai phá dữ liệu và phát hiện tri thức từ dữ liệu là lĩnh vực nghiên cứu và ứng dụng có quan hệ mật thiết với sự phát triển kinh tế – xã hội, vì vậy, theo thời gian, khai phá dữ liệu đã và đang thu hút thêm sự tham gia của nhiều ngành, chuyên ngành khác không chỉ trong lĩnh vực CNTT mà còn ở các lĩnh vực khác.

CÂU HỎI VÀ BÀI TẬP

- 1.1** Nội dung, ý nghĩa định hướng công nghiệp và kinh tế của định luật Moore.
- 1.2** Phân biệt bài toán quản trị cơ sở dữ liệu tác nghiệp với bài toán khai phá dữ liệu.
- 1.3** Phân tích vai trò của cơ sở tri thức trong một hệ thống khai phá dữ liệu.
- 1.4** Phân biệt bài toán khai phá dữ liệu với bài toán kiểm nghiệm giả thiết thống kê.
- 1.5** Han và Kamber [HK0106] quan niệm khai phá dữ liệu và phát hiện tri thức trong CSDL là bước phát triển mới của công nghệ CSDL. Hãy lập luận làm sáng tỏ quan niệm trên.

¹² <http://icnc-fskd.dhu.edu.cn/>

¹³ <http://rskt.cs.uregina.ca/>

- 1.6** Trình bày một số mẫu truy vấn trong hệ thống quản trị cơ sở dữ liệu và hệ thống khai phá dữ liệu. Phân tích làm sáng tỏ các mẫu truy vấn trong hệ thống khai phá dữ liệu là phức tạp hơn mẫu truy vấn trong hệ thống quản trị CSDL.
- 1.7** Hệ thống khai phá dữ liệu có nhất thiết có nguồn đầu vào là kho dữ liệu hay không? Phân tích một số lợi điểm khi hệ thống khai phá dữ liệu có nguồn dữ liệu đầu vào chỉ là các kho dữ liệu.
- 1.8** Phân tích về tính "không tầm thường" của quá trình phát hiện tri thức trong CSDL.
- 1.9** Phân biệt bài toán khai phá dữ liệu mô tả với bài toán khai phá dữ liệu dự báo.
- 1.10** Phân tích tầm quan trọng của khâu làm sạch dữ liệu và tiền xử lý dữ liệu trong quá trình khai phá dữ liệu và trình bày sơ bộ về nội dung của khâu này.
- 1.11** Phân tích về sự cần thiết phải tiến hành tính toán giá trị một số độ đo nào đó trong các bài toán khai phá dữ liệu.

Chương 2.

CÔNG NGHỆ TRI THỨC VÀ PHÁT HIỆN TRI THỨC TỪ DỮ LIỆU

Như đã được đề cập tại Chương 1, thế giới ngày nay đang chuyển đổi từ kinh tế hàng hóa (good economic) sang kinh tế dịch vụ (service economic). Ba khái niệm kinh tế nổi bật là kinh tế tri thức, kinh tế thông tin và kinh tế dịch vụ. Sự chỉ đỏ xuyên suốt nội dung ba khái niệm kinh tế nói trên là tri thức. Sử dụng tri thức là động lực chủ chốt cho tăng trưởng kinh tế quốc gia, cũng chính là động lực chủ chốt cho tăng cường lợi thế cạnh tranh của doanh nghiệp, tổ chức. Trong xu thế phát triển đó, CNTT ngày càng khẳng định tầm quan trọng chiến lược. Đặc biệt, ngành công nghiệp dựa trên dữ liệu đã được hình thành và đang phát triển với tốc độ cao. Khai phá dữ liệu và phát hiện tri thức trong dữ liệu là nền tảng của ngành công nghiệp dựa trên dữ liệu.

Chương 1 cũng đã trình bày một số nội dung khái quát về khai phá dữ liệu và phát hiện tri thức từ dữ liệu. Chương 2 sẽ giới thiệu chi tiết về vai trò và nội dung của công nghệ tri thức mà một nội dung cơ bản trong đó là phát hiện tri thức từ dữ liệu.

2.1. VAI TRÒ CỦA CNTT TRONG KINH TẾ TRI THỨC

Nghiên cứu khoa học liên lĩnh vực cho thấy phát triển CNTT và phát triển kinh tế có mối quan hệ hữu cơ mật thiết, trong đó các quốc gia có trình độ CNTT phát triển cao cũng chính là các quốc gia có nền kinh tế phát triển cao. Từ vị thế được kỳ vọng có phần quá cường điệu và mơ hồ ban đầu, CNTT ngày càng khẳng định vị thế chiến lược trong phát triển kinh tế, trong tăng trưởng hiệu quả của doanh nghiệp và tổ chức. Tuy nhiên, cần có một nền

tảng nhận thức chính xác và toàn diện về vị thế chiến lược của CNTT để xác định chiến lược phát triển dựa trên CNTT đúng đắn và ngăn ngừa được các biểu hiện sai lệch trong nhận thức về vai trò của CNTT, hoặc theo hướng ngộ nhận và lạm dụng vai trò của CNTT dẫn tới lãng phí, tham nhũng hoặc theo hướng phủ nhận vị thế chiến lược của CNTT.

Đầu tiên, mục con 2.1.1. giới thiệu một số luận điểm theo hướng phủ nhận vị thế chiến lược của CNTT, điển hình là luận điểm của Robert M. Solow vào năm 1987 và luận điểm của Nicolas Carr vào những năm 2003-2004. Tiếp theo, nhằm cung cấp một số nội dung làm sáng tỏ vị thế chiến lược của CNTT, khái niệm về kinh tế tri thức và vai trò của CNTT trong kinh tế tri thức sẽ được giới thiệu trong mục con 2.1.2.

2.1.1. Nghịch lý hiệu quả của CNTT của Robert Solow và luận điểm của N. Carr

2.1.1.1. Nghịch lý hiệu quả của CNTT

Vào năm 1987, Robert M. Solow, một nhà kinh tế người Mỹ được tặng giải thưởng Nobel về kinh tế, phát biểu "Chúng ta nhìn thấy máy tính ở mọi nơi ngoại trừ trong thống kê hiệu quả" (nguyên văn: You can see the computer age every where but in the productivity statistics) [Solow87]. Phát biểu này được Erik Brynjolfsson [Bryn93] chỉ dẫn như là "nghịch lý hiệu quả của CNTT (Productivity Paradox of Information Technology). Theo Erik Brynjolfsson, thống kê hiệu quả được R. M. Solow luận cứ trong nghịch lý hiệu quả của CNTT được diễn tả như dưới đây.

- Trong bốn thập niên (1960- 1990), tỷ lệ đầu tư cho máy tính của nước Mỹ tính theo GDP tăng nhanh từ 0,003 % GDP (thập niên 1960), 0,05% (thập niên 1970s), 0,3% (thập niên 1980s), tới 3,1% (thập niên 1990s) nhưng tỷ lệ tăng GDP trung bình theo năm lại giảm từ 4,5% (thập niên 1960s) xuống 2,95% (thập niên 1970s) rồi 2,75 (thập niên 1980s) và 2,20% (thập niên 1990s). Tăng đầu tư CNTT có vẻ như không góp phần vào tăng GDP nước Mỹ nếu không nói là còn làm giảm đi.

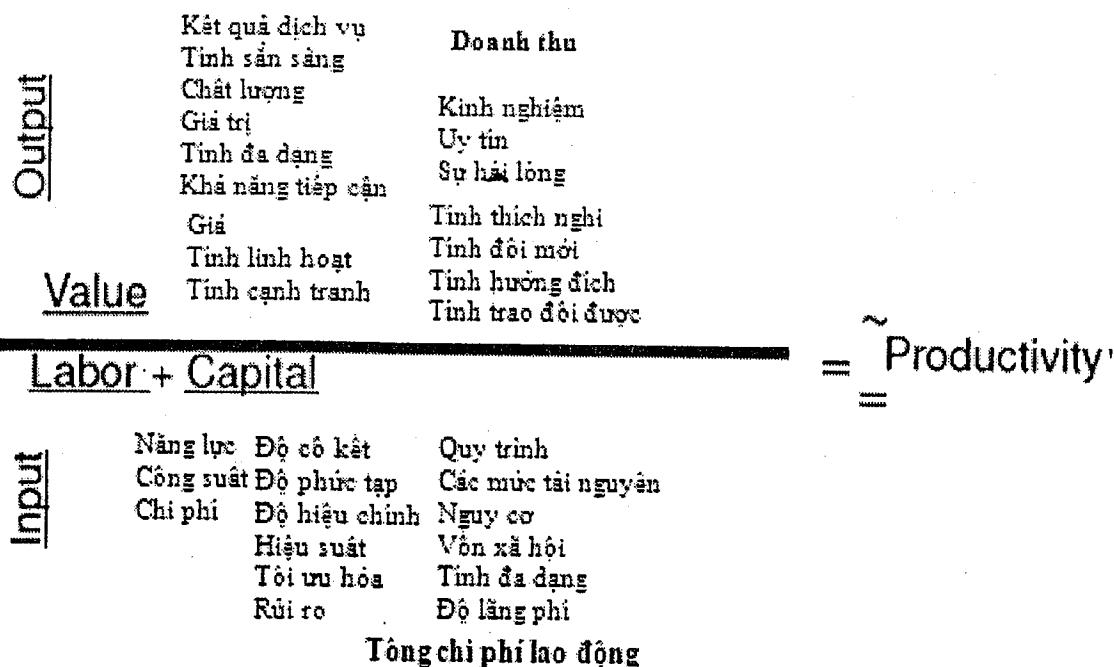
- Theo thống kê từ hàng trăm nghìn doanh nghiệp Mỹ, đầu tư CNTT tính theo đầu nhân viên và hiệu quả kinh doanh cũng không có mối quan hệ rõ ràng, không hướng tới kỳ vọng "đầu tư CNTT tăng thì hiệu quả kinh doanh cũng tăng". Có một trường hợp đặc biệt, riêng với các doanh nghiệp trong lĩnh vực tài chính – ngân hàng, hiệu quả kinh doanh có quan hệ tỷ lệ thuận với đầu tư CNTT.

- Tình trạng đầu tư CNTT một cách lãng phí cũng diễn ra đối với một bộ phận cá nhân và hộ gia đình.

E. Brynjolfsson [Bryn93] đưa ra nhận định rằng thực chất các hiện tượng trên đây không thực sự là nghịch lý hiệu quả của CNTT. Tác giả cung cấp bốn giải thích dưới đây về các hiện tượng nói trên:

- *Lỗi đo lường trong công thức tính hiệu quả*. Lỗi đo lường thể hiện theo hai khía cạnh chính. Công thức tính hiệu quả của kinh tế cổ điển có một lỗi lớn khi chỉ đo lường tài nguyên trực tiếp liên quan tới vốn, lao động và giá trị. Trong thời đại kinh tế tri thức (xu thế chuyển đổi từ kinh tế hướng hàng hóa sang kinh tế hướng dịch vụ), các yếu tố tài nguyên gián tiếp (tri thức nhân viên và tri thức doanh nghiệp, tài nguyên quy trình tổ chức của doanh nghiệp...) ngày càng đóng vai trò quan trọng trong phát triển kinh tế quốc gia và cạnh tranh doanh nghiệp thì chúng cần phải xuất hiện trong công thức đo lường hiệu quả dịch vụ với sự thanh giá của nhiều yếu tố tài nguyên gián tiếp cả ở đầu vào và đầu ra. Lưu ý rằng, đo lường tài nguyên gián tiếp (thuộc cả đầu ra lẫn đầu vào) lại là một bài toán rất khó.

- *Không giống như các khoản đầu tư cơ sở hạ tầng, đầu tư CNTT có một khoảng thời gian trễ để phát huy hiệu quả*. Điều này có nguyên nhân từ việc nhân viên trong doanh nghiệp phải có một khoảng thời gian (theo E. Brynjolfsson, thường là 2-3 năm) mới có thể sử dụng thành thạo các công cụ của CNTT. Tác giả cũng khuyến nghị về việc cần thực hiện giải pháp rút ngắn độ trễ này.



Hình 2.1. Một công thức đo lường hiệu quả.

- **Tính phân phối lại về tài nguyên thông tin.** Thông tin và tri thức vừa là tài nguyên quan trọng của doanh nghiệp song cũng được coi là một dạng "sản phẩm hàng hóa công cộng", cho nên, chi phí đầu tư CNTT để phát triển của một doanh nghiệp có thể bị bao gói thêm chi phí đầu tư CNTT cho doanh nghiệp khác. Đối với tình huống này, các doanh nghiệp cần phải đảm bảo được một yêu cầu là trong vòng đời của thông tin và tri thức của doanh nghiệp, chúng phải làm lợi nhiều nhất cho chính bản thân doanh nghiệp đã đầu tư.

- **Sai lầm trong quản lý đầu tư CNTT.** Các khoản đầu tư CNTT được thi hành song quyết định đầu tư chúng lại có thể không được định hướng tới lợi ích của doanh nghiệp. Tình trạng này có nguyên nhân từ các quyết định đầu tư là lỏng lẻo dẫn tới việc xây dựng các hệ thống không hiệu quả, hoặc đơn giản là sử dụng các chiến lược tạo quyết định lối thời khi quyết định đầu tư CNTT. Sử dụng chính công cụ CNTT, đặc biệt là công cụ khai phá dữ liệu, là một biện pháp khắc phục được hiện tượng này. Lưu ý rằng, ở đây không đề cập tới một vấn đề tiêu cực xã hội là tham nhũng trong đầu tư cho CNTT.

Như vậy, ngay từ những năm đầu tiên của thập niên 1990, các nhà khoa học đã khẳng định được rằng "nghịch lý hiệu quả của CNTT" là không đúng trong thực tiễn. Không những thế, vai trò chiến lược của CNTT ngày càng được nhấn mạnh trong phát triển kinh tế tri thức [OECD96]. Tuy nhiên, một vài nhà kinh tế, điển hình là Nicolas Carr, vẫn bảo thủ và bày tỏ mối nghi ngờ về vai trò chiến lược của CNTT.

2.1.1.2. Luận điểm của N. Carr

Vào năm 2003, N. Carr trình bày một số luận điểm sau đây phủ nhận vai trò chiến lược của CNTT [Carr03]:

- CNTT xuất hiện khắp nơi và tầm quan trọng chiến lược của nó đã giảm. Cách tiếp cận đầu tư và quản lý CNTT cần phải được thay đổi một cách đáng kể!
- Khi một tài nguyên (ý nói CNTT) trở thành bản chất để cạnh tranh nhưng đã không quan trọng cho chiến lược thì rủi ro mà nó tạo ra lại trở nên quan trọng hơn so với các lợi thế mà nó cung cấp.
- Với việc nhanh chóng biến mất các cơ hội đạt được lợi thế chiến lược từ CNTT, nhiều doanh nghiệp cần phải có một cái nhìn nghiêm khắc trong đầu tư vào CNTT và quản lý các hệ thống của họ.

Đồng thời, N. Carr đưa ra ba quy tắc hướng dẫn cho tương lai với định hướng phủ nhận vai trò chiến lược của CNTT. Năm 2005, N. Carr lại công bố một bài viết khác [Carr05] nhằm củng cố các luận điểm trên đây. Luận điểm phủ nhận vai trò chiến lược của CNTT mà N. Carr phát biểu đã tạo ra một làn sóng phản bác mạnh mẽ. Chính vì vậy, N. Carr đã lọt vào danh sách 100 người có tên được nhắc đến nhiều nhất trên thế giới.

Sai lầm của N. Carr là ở chỗ ông đã quan niệm CNTT như là một loại công nghệ hạ tầng (giống như điện năng), từ đó dẫn đến việc không nhận thức được vai trò chiến lược của CNTT trong phát triển tri thức quốc gia cũng như tri thức doanh nghiệp. Thông qua các phân tích liên quan tới 11 nhận định của

N. Carr, Paul A. Strassmann (*Executive Advisor, NASA; Former CIO of General Foods, Kraft, Xerox, the Department of Defense, and NASA*) đã làm sáng tỏ các sai lầm trong các bài viết của N. Carr¹⁴.

Tuy nhiên, tương tự như giải thích "nghịch lý hiệu quả của CNTT" từ yếu kém trong quản lý đầu tư CNTT, khuyến cáo về cách thức tiếp cận đầu tư và quản lý CNTT của N. Carr cũng mang ý nghĩa tích cực. Dưới đây là một số nhận định của một số nhà quản lý liên quan tới khuyến cáo này (xem chú thích 14):

- Nếu có một điều mà chúng ta học được từ những năm 1990 là sự khởi đầu dựa trên CNTT, tưởng như một vụ nổ vũ trụ nhưng lại hiếm khi tạo ra một đền đáp tương xứng như kỳ vọng. Lê ra phải giúp các doanh nghiệp hiểu rằng CNTT chỉ là một công cụ, các nhà cung cấp công nghệ lại nhầm tới nó như một thuốc bách bệnh “Mua công nghệ này đi và các vấn đề của anh sẽ được giải quyết!” (*John Seely Brown, Former Chief Scientist, Xerox Palo Alto, California và John Hagel III, Management Consultant and Author, Burlingame, California*).

- Công việc của CTO (Chief Of Technical: người đứng đầu bộ phận công nghệ) và CIO (Chief Of Information: người đứng đầu về thông tin) của tổ chức sẽ trở nên quan trọng chưa từng có trong các thập niên tiếp theo. Gói kỹ năng cần thiết trong một tổ chức sẽ thay đổi rất nhanh để cạnh tranh trong thời đại thông tin (*F. Warren McFarlan, Albert H. Gordon Professor of Business Administration, Harvard Business School, Boston và Richard L. Nolan, William Barclay Harding Professor of Business Administration, Harvard Business School, Boston*)

- Tôi đồng tình nhiều với khuyến cáo của Nicholas Carr về cách thức các doanh nghiệp nên có phản ứng với một thực tế không thể chịu đựng được là CNTT đã trở thành một loại hàng hóa. Nhưng tại sao Carr lại khuyến cáo các điều lo lắng tới các nhà quản lý CNTT? Phải chăng là vì các bài toán lãnh đạo như quản lý và kiểm soát rủi ro về kinh phí ít hứa hẹn hoặc thách thức

¹⁴ Harvard Bussiness Review, June 2003

hơn so với việc theo đuổi lợi thế cạnh tranh? CNTT luôn luôn quan trọng – là vấn đề trong mọi quan niệm. CNTT bắt buộc hỗ trợ kinh doanh – không chỉ bằng áp dụng lôgic về công nghệ mà còn bằng áp dụng lôgic về bản chất chung (*Jason Hittleman, IT Director, RKA Petroleum Companies, Romulus, Michigan*).

Liên quan tới đầu tư cho CNTT, thông qua việc khảo sát về đầu tư và hiệu quả CNTT của trên 5700 doanh nghiệp Mỹ, Paul A. Strassmann đã đưa ra một số khuyến nghị [Strass07]:

- Có thể chi tiêu cho CNTT hơn hoặc kém so với mức trung bình của các doanh nghiệp đồng hạng (gọi là *mức thông thường*), nhưng về tổng thể thì chi tiêu như thế cần đưa tới *hiệu quả đo lường* được mà không phải chỉ là hiệu quả nói chung.
- Có thể chi tiêu cho CNTT hơn mức thông thường khi mà *hiệu quả thông tin* đạt được vẫn hơn mức thông thường.
- Có thể chi tiêu cho CNTT hơn mức thông thường khi mà *giá trị tri thức của nhân viên* đạt được vẫn hơn mức thông thường.

Như vậy, hiệu quả đầu tư CNTT trong doanh nghiệp cần phải đo lường được và được đo lường theo nhiều tiêu chí, trong đó *hiệu quả thông tin* và *hiệu quả về giá trị tri thức của nhân viên* được Paul A. Strassmann coi là hai tiêu chí quan trọng. Điều này hoàn toàn phù hợp với công thức tính hiệu quả trong lý thuyết kinh tế hiện đại, trong công thức đó, giá trị tri thức vừa là yếu tố đầu vào, vừa là yếu tố đầu ra. Theo Mörten Simonsson [Simon08], doanh nghiệp đương đại phần lớn phụ thuộc vào CNTT, vì vậy việc ra quyết định về CNTT của doanh nghiệp có ý nghĩa rất quan trọng.

2.1.2. Vai trò của CNTT trong nền kinh tế tri thức

Theo Ngân hàng Thế giới [WB06], “nền kinh tế tri thức (*Knowledge Economy*) hay nền kinh tế dựa trên tri thức (*Knowledge-Based Economy*) là nền kinh tế mà việc sử dụng tri thức là động lực chủ yếu cho tăng trưởng kinh tế”. Phát biểu trên đây khẳng định vai trò “tài nguyên chủ yếu” của tri thức trong

nền kinh tế. Các quốc gia có nền kinh tế phát triển nhất cũng chính là các quốc gia có trình độ kinh tế tri thức cao nhất, và ngược lại, các quốc gia nghèo nhất cũng chính là các quốc gia có trình độ kinh tế tri thức thấp nhất¹⁵.

Nền kinh tế tri thức dựa trên bốn cột trụ:

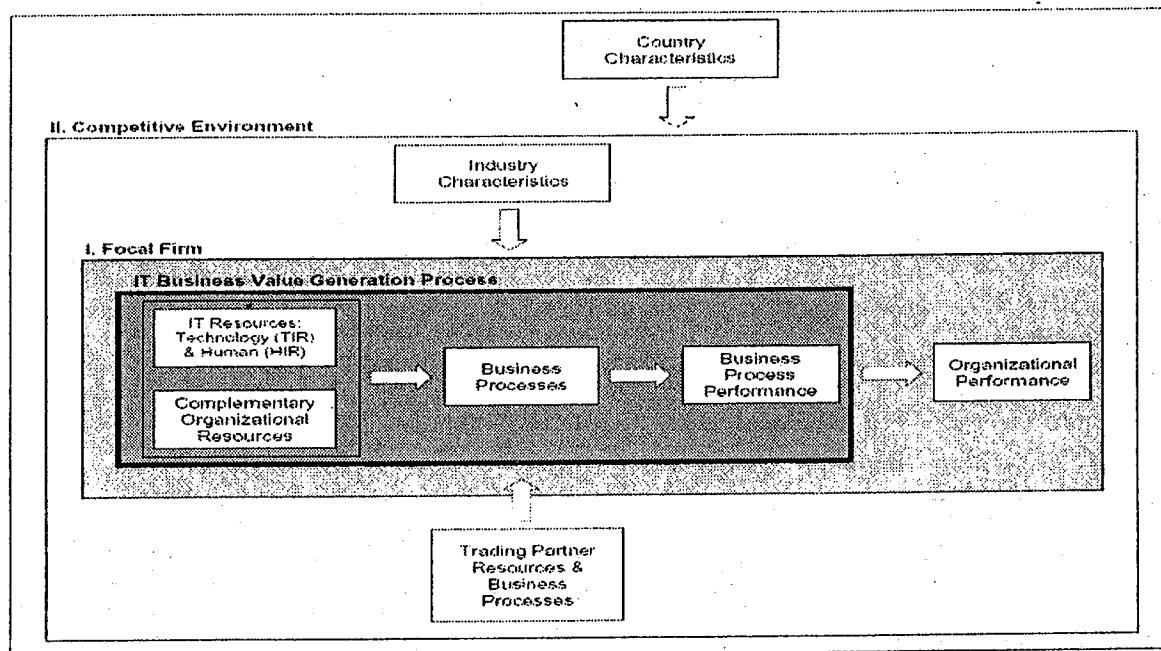
- (1) Một thiết chế xã hội pháp quyền và khuyến khích kinh tế (*An economic incentive and institutional regime*);
- (2) Một lực lượng lao động được giáo dục và lành nghề (*An educated and skilled labor force*);
- (3) Một hệ thống cách tân hướng tri thức hiệu quả (*a effective innovation system*);
- (4) Một hạ tầng thông tin hiện đại và đầy đủ (*a modern and adequate information infrastructure*).

Để nâng cao trình độ kinh tế tri thức thông qua các cột trụ kinh tế tri thức, các quốc gia kinh tế phát triển nhất thế giới đã chú trọng tăng cường đầu tư về tri thức, chú trọng đầu tư cho nghiên cứu-phát triển, phần mềm và giáo dục đại học. Có thể lấy một ví dụ từ bài học Hàn Quốc [WB06]. *Giáo dục và nguồn nhân lực* là hai yếu tố tài nguyên tri thức đóng góp chủ chốt cho sự tăng trưởng kinh tế kỳ diệu của Hàn Quốc trong suốt bốn thập niên 1960-1990. Vào năm 2004, phần đóng góp của tài nguyên tri thức cho sự tăng trưởng GDP tính theo đầu người đã gấp hơn ba lần so với phần đóng góp của tài nguyên cơ bản (bao gồm vốn và lao động). Trong [WB06], Ngân hàng Thế giới cung cấp số liệu về tỷ lệ đầu tư tính theo GDP cho tri thức (đầu tư cho nghiên cứu-triển khai, cho phần mềm và cho giáo dục đại học) và đầu tư cho máy móc và trang thiết bị của các nền kinh tế phát triển nhất thế giới vào năm 2002 cho thấy đầu tư cho tri thức chiếm một tỷ trọng cao. Hơn nữa, trong giai đoạn 1994-2002, xu thế chung tại các nền kinh tế phát triển nhất thế giới là tỷ lệ đầu tư tính theo GDP cho tri thức tăng và tỷ lệ đầu tư tính theo GDP cho máy móc và trang thiết bị giảm. Theo thống kê vào năm 2010 của Tổ chức hợp tác và

¹⁵ http://info.worldbank.org/etools/kam2/KAM_page5.asp.

phát triển kinh tế (Organisation for Economic Co-operation and Development: OECD), tổng đầu tư nội địa cho R&D tính theo GDP vào năm 2008 là cao hơn so với năm 1999 ở đa số các quốc gia (31/41) thuộc tổ chức này, đưa tỷ lệ đầu tư cho R&D trung bình của toàn khối OECD tăng từ 2,16% GDP năm 1999 lên 2,28% GDP năm 2008¹⁶.

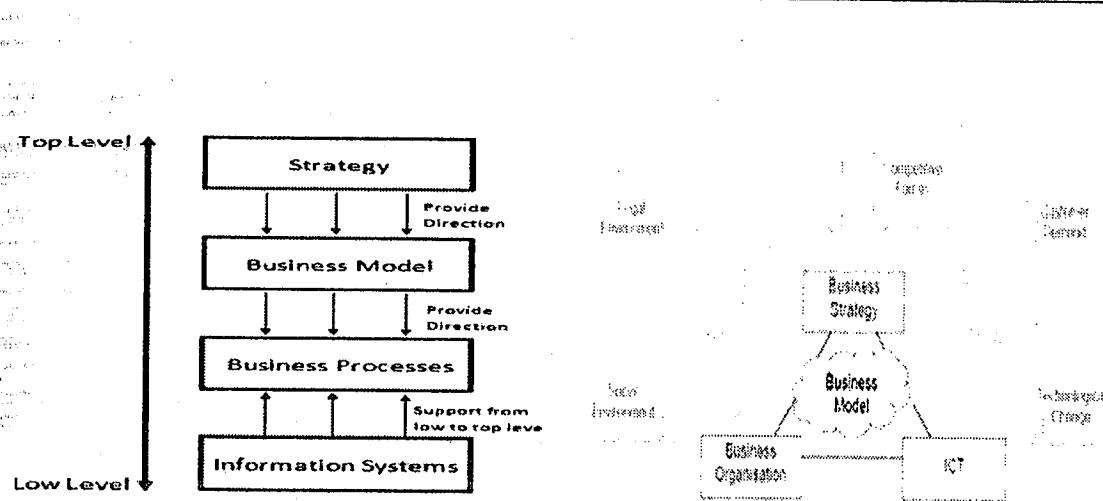
Việc sử dụng tri thức trong nền kinh tế tri thức được diễn ra trong các loại hoạt động là yêu cầu tri thức, phát sinh tri thức, phổ biến tri thức và vận dụng tri thức một cách hiệu quả cho tăng trưởng kinh tế. Ở cấp độ quốc gia, tri thức là nguồn tài nguyên chủ yếu cho tăng trưởng kinh tế, còn ở cấp độ doanh nghiệp, tri thức là nguồn tạo ra lợi thế cạnh tranh cho doanh nghiệp.



Hình 2.2. Vị trí của CNTT trong kinh tế vĩ mô [MKG04].

Nigel Melville và cộng sự [MKG04] đã cho một khung kinh tế vĩ mô với sự tham gia của CNTT (Hình 2.2), trong đó tập trung vào vị trí trong các doanh nghiệp địa phương.

¹⁶ <http://dx.doi.org/10.1787/820860264335>



Hình 2.3. Khung xác định mô hình kinh doanh (trái) và vị trí của mô hình kinh doanh trong doanh nghiệp (phải) [SG10].

Sơ đồ bên phải của Hình 2.3 cho thấy một giao kết bộ ba mật thiết giữa Tổ chức kinh doanh, Chiến lược kinh doanh và Công nghệ Thông tin – Truyền thông (CNTT-TT) và điều này càng khẳng định vai trò chiến lược của ICT đối với tổ chức. Tham gia vào bộ ba đó, Khai phá dữ liệu và phát hiện tri thức từ dữ liệu được coi là một bộ phận tích cực của CNTT.

Vai trò chiến lược của CNTT còn được thể hiện ở chỗ, các hệ thống cung cấp thông tin (nói chung) cũng như các ứng dụng khai phá dữ liệu (nói riêng) tại các doanh nghiệp đều cần phải xuất phát từ yêu cầu kinh doanh của doanh nghiệp. Như vậy, nhằm tăng cường tri thức tại doanh nghiệp, các bài toán khai phá dữ liệu được đặt ra và chúng có điểm xuất phát từ nhu cầu kinh doanh và phục vụ chiến lược kinh doanh của doanh nghiệp.

Vai trò chiến lược của CNTT đối với doanh nghiệp càng trở nên đặc biệt quan trọng trong giai đoạn suy thoái kinh tế. Dự báo về thông minh kinh doanh năm 2009 của Gartner đã minh chứng cho nhận định này¹⁷. Nói riêng, cùng với dự báo hơn 35% trong số 5.000 doanh nghiệp toàn cầu hàng đầu sẽ thất bại trong việc ra quyết định sâu sắc về sự thay đổi đáng kể trong kinh doanh và thị trường do suy thoái kinh tế, Bill Hostmann (Phó Chủ tịch nghiên cứu và phân tích của Gartner) khuyến nghị "Nhà lãnh đạo CNTT trong các doanh nghiệp có nền văn hóa quản lý mạnh dựa trên

¹⁷ <http://www.gartner.com/it/page.jsp?id=856714>

thông tin nên tạo một lực lượng đặc nhiệm để đáp ứng sự thay đổi nhu cầu thông tin và phân tích điều hành. Nhà lãnh đạo CNTT trong các doanh nghiệp chưa có văn hóa như vậy nên soạn thảo các văn bản về chi phí và phương hướng để thích nghi với điều kiện mới và đề xuất một trường hợp kinh doanh cho đầu tư hạ tầng thông tin, quy trình và công cụ hỗ trợ ra quyết định".

Những nội dung được trình bày về tri thức và công nghệ tri thức (Knowledge Technology) ở mục 2.2 tiếp theo sẽ làm sáng tỏ thêm vai trò chiến lược của CNTT trong nền kinh tế và cho doanh nghiệp.

2.1.2. Vai trò của giám đốc thông tin trong doanh nghiệp và tổ chức

Cùng với sự hình thành và phát triển của ngành công nghiệp dữ liệu, cùng với sự phát triển kinh tế tri thức, tài nguyên tri thức nói chung và gói kỹ năng cần thiết nói riêng của một tổ chức cần được thay đổi một cách kịp thời để cạnh tranh trong thời đại thông tin. Sự thành công của một tổ chức phụ thuộc mạnh vào nhận thức một cách hệ thống về môi trường xung quanh và nội bộ của tổ chức cũng như các chính sách và chiến lược của tổ chức được thông qua mà hệ thống thông tin tổ chức có vai trò rất quan trọng.

Giám đốc bộ phận thông tin (CIO) trong một tổ chức có trách nhiệm quản lý toàn bộ thông tin và công cụ hỗ trợ việc quản lý thông tin, là điểm trung gian giữa các mục tiêu kinh doanh hàng đầu của tổ chức với chức năng đảm bảo thông tin trong suốt. Theo nghiệp vụ, CIO cần định danh và tổng hợp thông tin của tổ chức và cho phép các nhà quản lý cao cấp truy cập chúng. Ngoài ra, CIO cần xác định các thông tin được sử dụng, thiết lập chính sách thông tin và tiêu chuẩn, duy trì kiểm soát quản lý trên tất cả các tài nguyên thông tin trong bất kỳ phương tiện truyền thông.

Ngày nay, CIO đóng vai trò trung tâm và cực kỳ quan trọng hoạt động quản lý hệ thống thông tin đảm bảo sự gắn kết CNTT vào chiến lược phát triển tổ chức, vì vậy vai trò tham gia điều hành của CIO trong doanh nghiệp ngày càng nổi bật. Nhiều công trình nghiên cứu về vai trò và đặc trưng của CIO đã được công bố.

Những nội dung được trình bày dưới đây được tổng hợp từ các tài liệu khảo sát quan trọng về nội dung này [Haw04, Hunter10, Line07, PCVM10]. Ý kiến trao đổi của CIO của 16 doanh nghiệp hàng đầu thế giới¹⁸ được E. Yourdon [Your11] biên tập là những nội dung tham khảo tốt về vai trò và đặc trưng của CIO hiện nay.

2.1.2.1. Vai trò của CIO

CIO có vai trò của một thành viên của đội quản lý cao cấp (Top Management Team: TMT) của tổ chức. CIO phân biệt với các thành viên khác của đội quản lý cao cấp do đặc thù của chức năng quản lý hệ thống thông tin. Mỗi quan hệ chặt chẽ giữa CEO và CIO trong doanh nghiệp góp phần nâng cao hiệu suất cải tiến quy trình kinh doanh (Business Process Improvement: BPI) và năng lực cơ sở hạ tầng CNTT. CIO phải là người tham gia vào quá trình lập kế hoạch chiến lược tổng thể cho doanh nghiệp. Trong một số trường hợp, CIO có thể không là thành viên của đội quản lý cao cấp, thì lúc đó, CIO nên báo cáo tới một thành viên đội quản lý cao cấp không là CEO (chẳng hạn, giám đốc tài chính - The chief Financial Officer: CFO).

CIO là người quản lý hệ thống công nghệ và tài nguyên thông tin, chịu trách nhiệm cá nhân về lập kế hoạch CNTT, về phát triển các hệ thống CNTT mới, về xây dựng chính sách CNTT.

2.1.2.2. Kỹ năng chính của CIO

Để đáp ứng vai trò quan trọng trong tổ chức, CIO cần có năng lực của một chuyên gia có nhận thức, thấu hiểu công nghệ, quá trình kinh doanh, chiến lược hành động của tổ chức và đáp ứng được những thay đổi và nhu cầu thị trường, có khả năng tương tác và giao tiếp hiệu quả với người quản lý cao cấp, đội quản lý cao cấp, cũng như môi trường tổ chức, và đảm bảo rằng tổ chức đi theo sự sáng tạo trong môi trường doanh nghiệp. Các kỹ năng chính dưới đây thể hiện cụ thể hóa năng lực chung nói trên của CIO:

¹⁸ Benjamin Fried, Tony Scott, Monte Ford, Mittu Sridhara, Steve Rubinow, Lewis Temares, Mark Mooney, Dan Wakeman, Lynne Ellyn, Becky Blalock, Ken Bohlen, Roger Gurnani, Ashish Gupta, Joan Miller, Vivek Kundra, Paul Strassmann

- *Năng lực tư duy và hành động chiến lược:* Tư duy và hành động chiến lược là kỹ năng quan trọng đối với một CIO, bởi vì chỉ với tư duy và hành động chiến lược, CIO mới giúp tổ chức đáp ứng với những thay đổi trên thị trường.Thêm nữa, tư duy và hành động chiến lược mới giúp CIO có ảnh hưởng vượt ra ngoài bộ phận CNTT.

- *Năng lực hành động nhanh chóng:* Chỉ có khả năng hành động nhanh chóng, CIO mới có thể hoàn thành dự án đúng kế hoạch. Nếu dự án không hoàn thành theo đúng tiến độ kế hoạch và nguồn lực sẽ làm suy yếu độ tin cậy của các chức năng IS trong các tổ chức.

- *Năng lực hòa giải xung đột:* CIO có trách nhiệm giải quyết các tình huống xung đột trong bộ phận thông tin dựa trên năng lực nền tảng về mối quan hệ con người và các ý niệm về biến đổi tâm lý và xã hội học.

- *Năng lực lãnh đạo và động viên đội làm việc:* CIO phải có năng lực tạo nên động lực và dẫn dắt bộ phận áp dụng kỹ thuật và kỹ năng để giải quyết các vấn đề và hoàn thành dự án trong thời hạn cho phép.

- *Năng lực quản lý dự án:* CIO phải có năng lực quản lý dự án trong bối cảnh tổ chức, bao gồm các quá trình liên quan đến khởi động, lập kế hoạch, thực hiện, giám sát, và hoàn thiện dự án, cũng như quản lý tích hợp, phạm vi, thời gian, chi phí, giám sát, chất lượng và rủi ro đối với dự án.

- *Năng lực giao tiếp:* Giao tiếp là một năng lực quan trọng để làm việc hiệu quả với các đối tác kinh doanh trong việc tìm hiểu và nắm bắt các nhu cầu kinh doanh khác nhau.

- *Năng lực đổi mới công nghệ:* CIO phải thực hiện nghiên cứu và đánh giá các công nghệ mới nổi, xem xét làm phù hợp tiềm năng của các công nghệ này với yêu cầu tổ chức và tạo các cơ hội kinh doanh mới.

- *Khả năng quan hệ cá nhân:* Có kỹ năng, sự sáng suốt làm việc với đồng nghiệp dựa trên việc thấu hiểu được hành vi, động lực của đồng nghiệp và tiến hành sự lãnh đạo hiệu quả.

- *Khả năng tạo và cơ cấu đội làm việc:* CIO có trách nhiệm tuyển dụng và duy trì đội trong bộ phận thông tin. CIO có năng lực xác định đúng các vai trò của đội làm việc và ánh xạ từng vai trò tới các thành viên trong đội. Năng lực phân tích quy trình làm việc của CIO thúc đẩy hoạt động của đội.

- *Kỹ năng đàm phán:* Khả năng đàm phán là rất quan trọng để một CIO cho phép đạt được lợi nhuận mà không ảnh hưởng mối quan hệ hiện có (tương tác win-win).

- *Khả năng thích ứng với thay đổi:* Khả năng thích ứng với thay đổi cho phép một lãnh đạo tốt hơn của các quá trình

- *Có tri thức kinh doanh:* CIO phải có một tri thức vừa rộng vừa chuyên sâu về các phương diện kỹ thuật và kinh doanh để có thể phối hợp hiệu quả trong quá trình cạnh tranh. Để phát triển một chiến lược nhất quán với các giá trị và văn hóa tổ chức, thì cần thiết phải hiểu môi trường tổ chức thông qua sự hiểu biết về cơ cấu tổ chức, nguồn nhân lực và kỹ năng của họ, các mối quan hệ hiện có (chính thức hoặc không chính thức), phong cách quản lý, các mối quan hệ bên ngoài v.v.

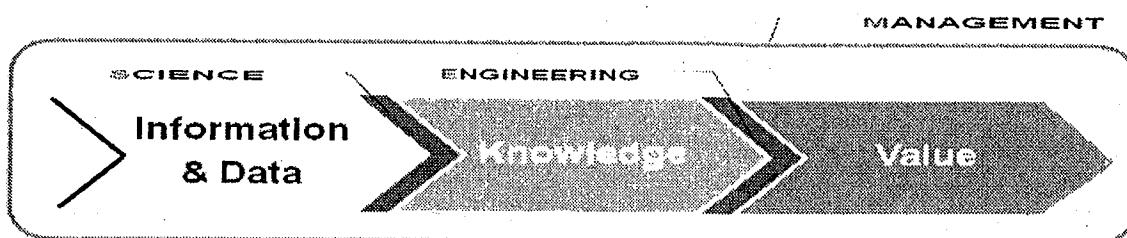
- *Trình độ kỹ thuật:* Tuy vai trò hướng tới kinh doanh ngày càng tăng nhưng CIO vẫn cần thực hiện trách nhiệm quản lý hoạt động công nghệ một cách hiệu quả. Thách thức đối với CIO là cần giữ một mức độ thích hợp kỹ năng kỹ thuật mà không xảy ra nguy cơ cho định hướng kinh doanh của tổ chức.

- *Năng lực ra quyết định:* Ra quyết định là một trong những kỹ năng chính hoặc vai trò chính của các nhà quản lý của tổ chức. Khi được công nhận ở trung tâm của quá trình ra quyết định, CIO tham gia chỉ đạo tổ chức tới những cơ hội mới để tăng khả năng cạnh tranh.

2.2. CÔNG NGHỆ TRI THỨC

Tăng cường tri thức cho cá nhân, doanh nghiệp và xã hội là một yêu cầu của mọi quốc gia trong xu thế phát triển kinh tế tri

thức hiện nay. Sơ đồ dưới đây thể hiện mô hình dịch vụ, yếu tố kinh tế cơ bản [Spoh06]:



Sơ đồ quá trình dịch vụ như trình bày trên đây cho thấy mối quan hệ của ba thành phần là khoa học, công nghệ và quản lý. Khoa học thi hành bước chuyển hóa thông tin và dữ liệu thành tri thức; công nghệ thi hành bước chuyển hóa tri thức thành giá trị; toàn bộ quá trình hai bước nói trên cần được quản lý tốt.

Công nghệ tri thức là thành phần tích cực của CNTT tham gia vào cả giai đoạn chuyển hóa dữ liệu – thông tin thành tri thức và cả giai đoạn chuyển hóa tri thức thành giá trị. Mục này đề cập tới hai khái niệm tri thức và công nghệ tri thức cùng một số nội dung liên quan.

2.2.1. Khái niệm tri thức

Chương 1 cung cấp một cách hiểu về khái niệm tri thức khi đặt khái niệm này trong bối cảnh của phát hiện tri thức trong dữ liệu “là những mẫu mới, có giá trị, hữu dụng, tiềm ẩn trong dữ liệu”. Theo C. Grube [Grube09], có hai dòng nghiên cứu tiếp cận tới tri thức, đó là, (1) tiếp cận theo khung nhìn triết học và tâm lý học dựa trên nhận thức luận, và (2) tiếp cận kinh tế học theo khung nhìn dựa trên tri thức của doanh nghiệp. Khung nhìn triết học và tâm lý học được thể hiện ở hầu hết nội dung trong khi khung nhìn kinh tế học được thể hiện tại mục tri thức của doanh nghiệp.

Bảng 2.1. Quan hệ một số cặp tri thức

	<i>Tri thức hiên</i>	<i>Tri thức ẩn</i>		<i>Tri thức hiên</i>	<i>Tri thức ẩn</i>
<i>Tri thức biết</i>	lý thuyết, khái niệm...	nhận thức, phán đoán..	<i>Tri thức khách quan</i>	sự kiện, quan trắc thực..	trực giác về các sự kiện..
<i>Tri thức làm</i>	phương pháp, thủ tục..	tài năng, kỹ năng...	<i>Tri thức chủ quan</i>	quan điểm, niềm tin rõ...	giả thiết ẩn, thế giới quan ẩn...

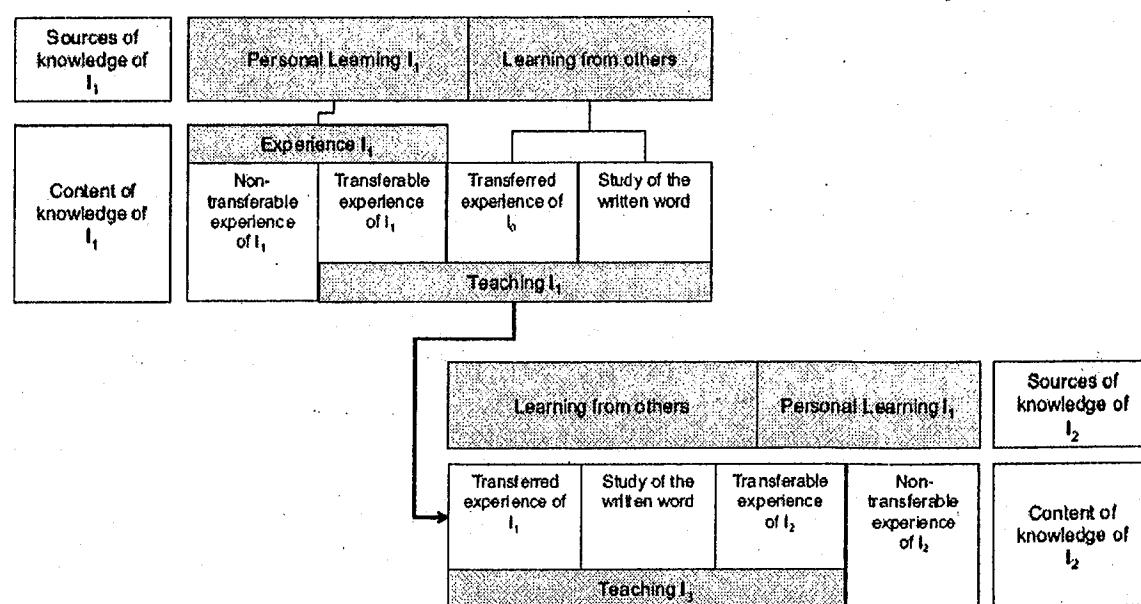
Theo nghĩa chung nhất (từ điển *Compact Oxford English Dictionary*) thì tri thức là “sự hiểu biết tinh thông và các kỹ năng mà con người thu nhận được theo kinh nghiệm và qua giáo dục”, “tổng hợp những gì mà con người biết rõ”, “nhận thức và hiểu biết tường minh về một sự việc hay một hiện tượng mà thu nhận được nhờ kinh nghiệm”. Trong phạm vi xác định vai trò của tri thức đối với cá nhân, tổ chức và xã hội, giáo trình này sử dụng nội dung trên đây cho khái niệm tri thức.

Tri thức được phân loại, thường được chia thành cặp tri thức, điển hình nhất là các cặp tri thức hiện – tri thức ẩn (Explicit knowledge – Tacit knowledge), tri thức chủ quan – tri thức khách quan (Objective knowledge – Subjective knowledge), tri thức biết – tri thức làm (Knowing that – Knowing how), trong đó hai cặp tri thức đầu tiên có tính đối ngẫu. Bảng 2.1 cung cấp mối quan hệ giữa cặp tri thức hiện – tri thức ẩn với hai cặp tri thức còn lại. Từ nội dung bảng 2.1 có thể thấy, tri thức hiện là tri thức mà mô tả được bằng văn bản. Chẳng hạn, các lý thuyết, khái niệm, phương pháp, thủ tục, sự kiện thực, quan trắc thực, quan điểm tường minh, niềm tin tường minh... là các dạng tri thức hiện. Đối ngẫu lại, tri thức ẩn là tri thức mà không thể mô tả được bằng văn bản. Chẳng hạn, nhận thức, phán đoán, tài năng, kỹ năng, trực giác, ngầm định... của các cá nhân là các dạng tri thức ẩn.

Trong quá trình vận động, tri thức được chuyển hóa từ dạng này sang dạng khác, trong đó có sự chuyển hóa từ tri thức ẩn sang

thi thực hiện. Sự hình thành và phát triển các ngành khoa học là thể hiện cho quá trình chuyển hóa này. Chẳng hạn, sự hình thành lĩnh vực công nghệ phần mềm được xuất phát từ một số cảm nhận ban đầu về tính đúng đắn của chương trình sau "cuộc khủng hoảng về lập trình" trong thập niên 1960.

Tồn tại một dạng tri thức đặc biệt "tri thức về tri thức" và được gọi là siêu tri thức (meta-knowledge). Siêu tri thức được chia thành 4 dạng và được ký hiệu là YKYK (You Know that You Known), DKYN (Do not Know that You Know), YKDK (You Know that you Do not Know), và DKDK (Do not Know that You don't Know) [WB98]. Một số ví dụ về siêu tri thức YKYK là (1) Ta biết về điều ta biết (qua quan sát trực tiếp của chính ta) là ô tô không thể chạy nếu thiếu nhiên liệu; (2) Ta biết về điều ta biết là nước sôi ở 100°C ; (3) Ta biết về điều ta biết là nếu ta cho xe máy chạy vượt đèn đỏ mà công an nhìn thấy thì ta sẽ bị phạt... Một số ví dụ về siêu tri thức YKDK là (1) Ta biết về một điều ta không biết (ta không trực tiếp quan sát được) là trung bình mưa tại vùng cao rộng lớn của Amazon là hơn 78 inches hàng năm; (2) Ta biết về một điều mà ta không biết về góc quay của Trái đất theo quỹ đạo của nó xung quanh mặt trời theo một góc $23,5$ độ...



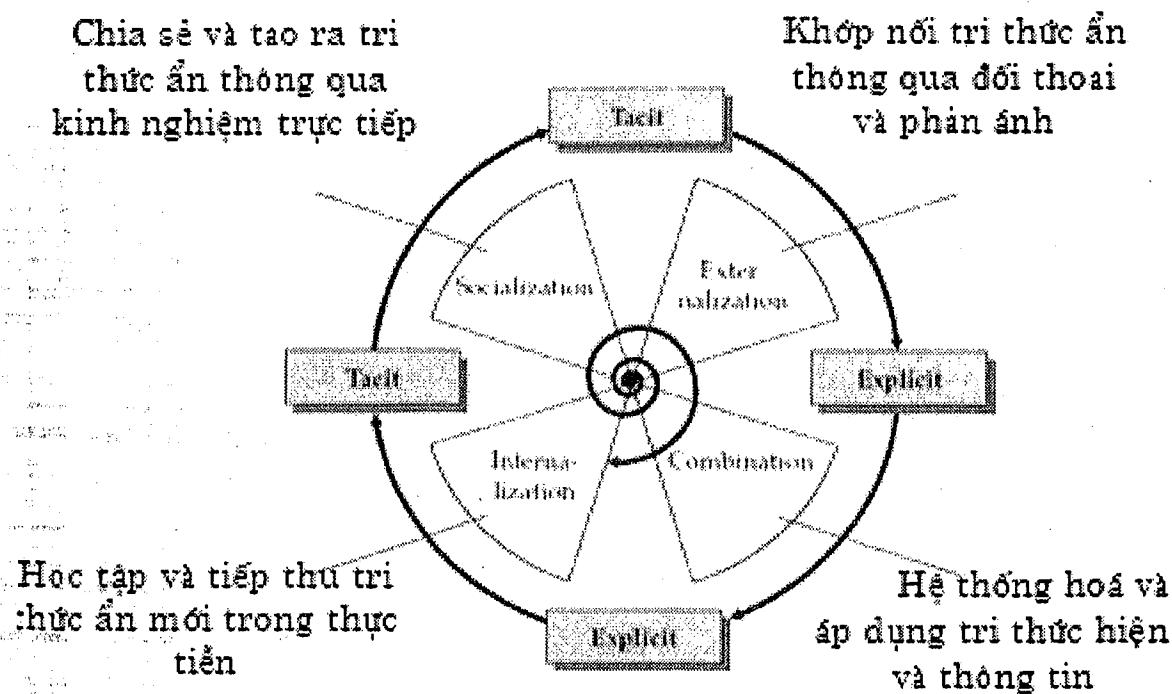
Hình 2.4. Nguồn tri thức của cá nhân [Grube09]

Lĩnh vực khai phá dữ liệu và phát hiện tri thức từ dữ liệu có mục tiêu chuyển đổi tri thức hiện từ dạng dữ liệu quan sát được thành các tri thức hiện dưới dạng các mẫu trong một ngôn ngữ biểu diễn, có nghĩa là chuyển đổi siêu tri thức dạng DKYK (tri thức tiềm ẩn trong dữ liệu) thành siêu tri thức dạng YKYK [BNGC00].

2.2.2. Nguồn tri thức cho cá nhân và tổ chức

2.2.2.1. Nguồn tri thức cho cá nhân

Theo C. Grube [Grube09], tri thức của cá nhân có được từ học tập và từ kinh nghiệm. Hình 2.4 trình bày phương án tăng cường tri thức của cá nhân thông qua tự học (qua đúc rút kinh nghiệm) và học từ người khác. Kinh nghiệm mà cá nhân đúc rút được gồm có kinh nghiệm chuyển giao được và kinh nghiệm không chuyển giao được cho người khác. Tri thức có được do học hỏi người khác theo hai kiểu là thông qua kinh nghiệm chuyển giao được của người khác hoặc từ các nghiên cứu đã được viết ra thành lời (được văn bản hóa).

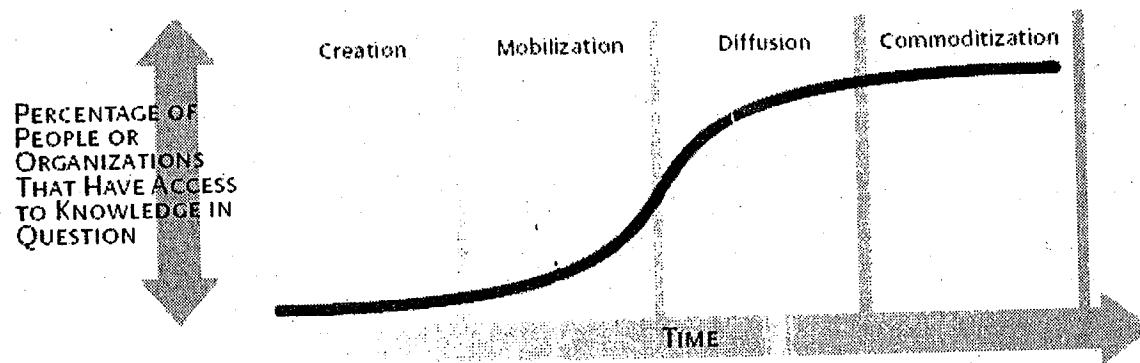


Hình 2.5. Quy trình xoắn ốc tri thức SECI [Hiro06]

Takeuchi Hirotaka [Hiro06] mô tả quá trình phát triển tri thức theo một quy trình chuyển hóa tri thức dạng xoắn ốc SECI (Hình 2.5) được phát triển từ ma trận chuyển hóa tri thức SECI (Socialization – Xã hội hóa, Externalization – Ngoại hiện, Combination - Kết hợp, Internalization - Tiếp thu) được Ikujiro Nokata và Takeuchi Hirotaka giới thiệu vào năm 1995. Trong quy trình này, Socialization chuyển tri thức ẩn sang tri thức ẩn thông qua hoạt động chia sẻ và đúc rút kinh nghiệm bản thân. Externalization kết nối tri thức ẩn thành tri thức hiện thông qua hoạt động đối thoại và phản ánh. Combination thực hiện việc hệ thống hóa, áp dụng tri thức và thông tin để có tri thức hiện mới từ tri thức hiện đã có. Áp dụng tri thức hiện đã có vào thực tiễn, Internalization là quá trình tiếp thu tri thức ẩn mới trong hoạt động thực tiễn.

2.2.2.2. Nguồn tri thức của tổ chức, doanh nghiệp

Như đã được giới thiệu, tiếp cận kinh tế học dựa trên khung nhìn tri thức của doanh nghiệp là một trong hai dòng nghiên cứu chính về tri thức. Trước hết, doanh nghiệp tồn tại dưới dạng và dựa trên một gói tài nguyên tri thức chuyên ngành, hay nói khác đi, doanh nghiệp tồn tại để tạo, chuyển giao, áp dụng và bảo vệ tài nguyên tri thức của nó.



Hình 2.6. Quá trình tiến hóa tri thức trong doanh nghiệp [BS02]

Quá trình tiến hóa tri thức trong doanh nghiệp theo thời gian diễn ra qua bốn giai đoạn phát triển là sáng tạo, huy động,

phổ biến và hàng hóa (Hình 2.6). Khi tri thức doanh nghiệp đã trở nên truy cập được đối với càng nhiều người hơn - đầu tiên trong một tổ chức, sau đó trong nhiều tổ chức, và cuối cùng cho đại chúng - các doanh nghiệp phải sử dụng các chiến lược khác nhau để thu nhận được giá trị lớn nhất của tri thức.

Bảng 2.2 đưa ra một khung nhìn về mô hình SECI theo định nghĩa, phương pháp và nội dung của mỗi cơ chế chuyển đổi tri thức. Một số ví dụ trong bảng có liên quan tới chuyển đổi tri thức cá nhân tại doanh nghiệp.

Bảng 2.2. Chuyển đổi tri thức cá nhân trong doanh nghiệp

Ấn → Hiện: EXTERNALIZATION		Ấn → Ấn: SOCIALIZATION	
<i>Định nghĩa</i>	Phát biểu rõ ràng và hệ thống hóa tri thức ẩn	<i>Định nghĩa</i>	Chia sẻ kinh nghiệm giữa các cá nhân và giữa các tổ chức với cá nhân
<i>Phương pháp</i>	Viết	<i>Phương pháp</i>	Kinh nghiệm
<i>Nội dung</i>		<i>Nội dung</i>	Tri thức đồng cảm (ví dụ như mẫu tinh thần được san sẻ, kỹ năng kỹ thuật)
Hiện → Hiện: COMBINATION		Hiện → Ấn: INTERNALIZATION	
<i>Định nghĩa</i>	Kết hợp các phần chính yếu khác nhau của tri thức hiện	<i>Định nghĩa</i>	Chuyển đổi các hướng dẫn và các nguyên tắc thành các trực giác và thói quen
<i>Phương pháp</i>	Phương tiện trao đổi (ví dụ như tài liệu, cuộc đàm thoại, mạng CNTT)	<i>Phương pháp</i>	Học qua công việc
<i>Nội dung</i>	Tri thức hệ thống hóa (ví dụ như các mẫu, các công nghệ thành phần mới)	<i>Nội dung</i>	Tri thức hành động (ví dụ như quản lý dự án, quy trình sản xuất)

Các nghiên cứu theo khung nhìn kinh tế về tri thức doanh nghiệp cho thấy (1) Tri thức doanh nghiệp là nền tảng của sự tồn tại doanh nghiệp (ra đời, phát triển và bị diệt vong) trong nền kinh tế, đặc biệt là trong nền kinh tế thị trường tự do; (2) Tri thức doanh nghiệp là nguyên nhân của sự đa dạng doanh nghiệp hoạt động trong cùng một ngành sản xuất, kinh doanh.

Tri thức doanh nghiệp không phải đơn thuần là sự hợp cơ học từ tri thức của tập cá nhân thuộc doanh nghiệp mà doanh nghiệp cũng là một thực thể tri thức. Theo phạm vi doanh nghiệp, C. Grube [Grube09] giới thiệu một số luận điểm sau đây theo hướng tiếp cận kinh tế về tri thức doanh nghiệp:

- Doanh nghiệp là một thực thể tích hợp tri thức: Môi trường văn hóa doanh nghiệp và tính chất chuyên môn trình độ cao liên quan tới ngành nghề của doanh nghiệp tạo nên một cộng đồng đơn nhất doanh nghiệp thực hiện thu nhận và chuyển giao tri thức hướng tới mục tiêu tốt nhất hoặc hiệu quả nhất cho doanh nghiệp.

- Doanh nghiệp là một thực thể sáng tạo tri thức: Tri thức doanh nghiệp không chỉ đơn thuần là kết quả hợp tri thức phân tán của tập cá nhân mà doanh nghiệp còn tạo ra tri thức thông qua việc cung cấp cho các thành viên một ý thức cộng đồng, một bản sắc văn hóa và một mô hình của tinh thần san sẻ. Một tập hợp mạng quan hệ trong doanh nghiệp tạo điều kiện thuận lợi cho trao đổi và phát triển tri thức doanh nghiệp. Vào năm 2000, Giám đốc điều hành tập đoàn HP Lew Platt lúc đó nhận định "Nếu HP biết được những điều HP biết thì lợi nhuận của chúng tôi sẽ gấp ba lần"¹⁹ cho thấy tác dụng kinh tế của việc tạo được một môi trường tốt cho trao đổi và phát triển tri thức doanh nghiệp.

- Doanh nghiệp là thực thể bảo vệ tri thức. Một mặt, doanh nghiệp tạo điều kiện thuận lợi cho trao đổi và phát triển tri thức nội bộ, mặt khác, doanh nghiệp cần có cơ chế bảo vệ tri thức doanh nghiệp. Doanh nghiệp cần thực hiện các biện pháp điều khiển quá trình tiến hóa tri thức doanh nghiệp (Hình 2.5) để tri thức doanh nghiệp mang được lợi ích nhiều nhất cho doanh nghiệp.

Một số cơ chế phối hợp sau đây có thể được thực hiện trong thực thể tri thức doanh nghiệp:

- Các quy tắc tương tác giữa các cá nhân trong doanh nghiệp tạo điều kiện thuận lợi cho chuyển hóa tri thức ẩn thành tri thức hiện;

- Chuẩn hóa hoạt động mức doanh nghiệp như quá trình tiến hành các bước tham gia của các chuyên gia vào sản phẩm. Nên và chỉ nên sử dụng các quy trình chuẩn đối với các vấn đề quá phức tạp hoặc quan trọng và bất thường;

¹⁹ Nguyên văn, "If HP knew what HP knows, we would be three times profitable".

- Các thói quen được hình thành trong doanh nghiệp để hỗ trợ sự tương tác linh hoạt trong doanh nghiệp, một bộ phận quan trọng trong văn hóa doanh nghiệp. Hình thành được các thói quen như vậy đòi hỏi rất nhiều thời gian và công sức. Văn hóa doanh nghiệp là một tài nguyên quan trọng trong hoạt động tạo năng lực cạnh tranh, có ý nghĩa ngày càng quan trọng trong xu thế toàn cầu hóa ngày nay [RB10].

2.2.3. Công nghệ tri thức

2.2.3.1. Một số khái niệm liên quan

Công nghệ nghệ tri thức theo định nghĩa truyền thống là lĩnh vực liên quan tới quá trình thu nhận tri thức và giải thích dựa trên tri thức thu nhận được. Các bước trong quá trình công nghệ tri thức là thu nhận tri thức, biểu diễn tri thức, xây dựng một cơ chế suy luận, và thiết kế các công cụ giải thích.

Thu nhận tri thức là việc khai thác tri thức từ các nguồn dưới dạng “văn bản được” (hướng dẫn, phim ảnh, sách, cơ sở dữ liệu, tập tin văn bản, hình ảnh, băng hình, đầu ra cảm biến..) và dưới dạng “không văn bản được” (tâm trí con người, tâm tri chuyên gia) và chuyển như tri thức thu nhận được vào máy tính. Thu nhận tri thức là một công việc khó khăn do một số nguyên nhân như sự không phù hợp của biểu diễn tri thức từ các nguồn phức (như liệt kê ở trên), đòi hỏi số lượng không nhỏ lực lượng người thu thập tri thức, chuyển giao kết quả đầu ra của thu nhận tri thức cho máy tính, khó khăn của chuyên gia khi mô tả tri thức của họ. Có thể tiến hành một số kỹ thuật tự động thu thập tri thức, chẳng hạn như phép quy nạp, lập luận dựa trên trường hợp, tính toán nơron.

Biểu diễn tri thức liên quan đến việc tổ chức tri thức trong các cơ sở tri thức; tri thức được biểu diễn dưới dạng tri thức mô tả (cái đó là gì) và dưới dạng tri thức thủ tục (phổ biến là mối quan hệ IF-THEN). Tri thức thủ tục là phần tử cơ bản hình thành cơ chế suy luận, tri thức mô tả được sử dụng cho giải thích.

Quan sát lại sơ đồ hoạt động của một hệ thống khai phá dữ liệu được trình bày tại Chương 1, chúng ta nhận thấy rằng hệ thống khai phá dữ liệu bao gói một quá trình công nghệ tri thức. Như vậy, khai phá dữ liệu và phát hiện tri thức từ dữ liệu là một phương án của công nghệ tri thức, trong đó quá trình công nghệ tri thức (thu thập tri thức, biểu diễn tri thức, suy luận và giải thích) được thực hiện chủ yếu dựa trên các kỹ thuật tự động.

Trong hệ thống khai phá dữ liệu, phát hiện tri thức (một hình thức của thu thập tri thức) từ dữ liệu được coi là thành phần quan trọng nhất. Nguồn tri thức đầu vào của hệ thống này là tri thức dưới dạng văn bản (dữ liệu ghi nhận các sự kiện, các mô tả..). Tri thức dạng không văn bản (tâm trí chuyên gia) nếu có được sử dụng thì được thi hành trong một số khâu, trong đó có khâu tạo ví dụ mẫu (các ví dụ mẫu đó cũng là tri thức dạng văn bản). Trong các mô hình khai phá dữ liệu gần đây (chẳng hạn như trong [CYZ10]), việc thu nhận tri thức chuyên gia miền ứng dụng được thi hành ở rất nhiều pha của quá trình khai phá dữ liệu.

Pha thi hành thuật toán khai phá dữ liệu là pha quan trọng thực hiện cơ chế suy diễn từ dữ liệu đã có nhận được tri thức mới, tiềm ẩn, hữu ích, có giá trị.

Cơ sở tri thức của hệ thống khai phá dữ liệu cũng đảm nhận cơ chế suy diễn, đồng thời cũng bao gồm các công cụ giải thích dựa trên tri thức thuộc quá trình công nghệ tri thức.

Biểu diễn tri thức được thi hành không chỉ trong cơ sở tri thức của hệ thống khai phá dữ liệu mà còn được thi hành trong giai đoạn trực quan hóa biểu diễn tri thức cho người sử dụng.

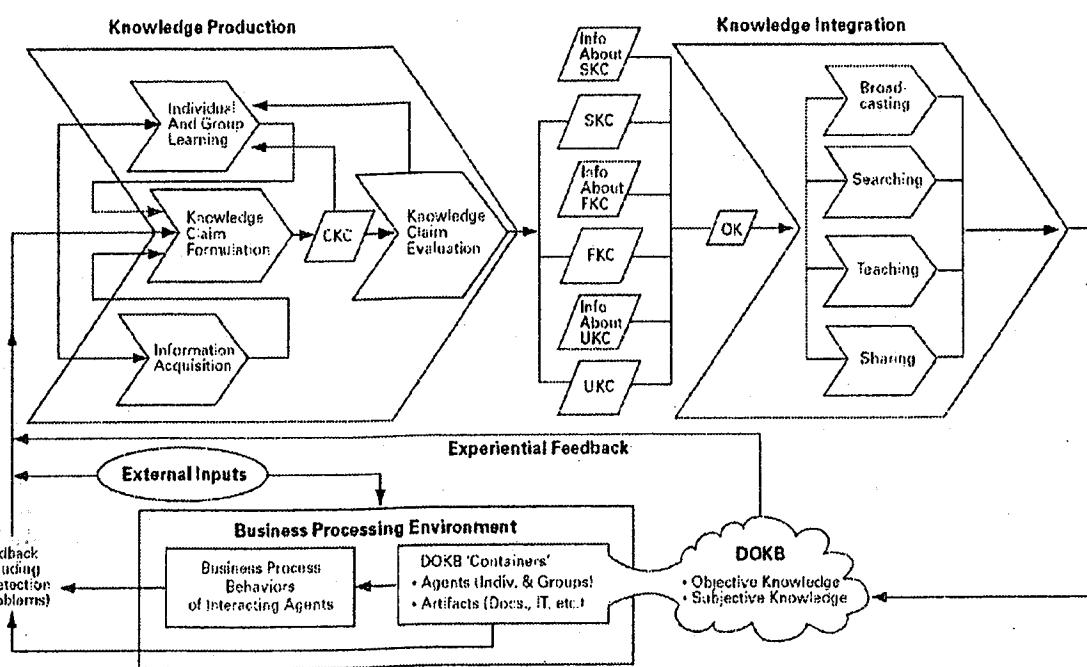
2.2.3.2. Vòng đời của tri thức doanh nghiệp

Hình 2.7 trình bày vòng đời tri thức doanh nghiệp theo trình bày của Mark W. McElroy [Elroy02]. Trong hình 2.7, CKC (Codified knowledge claim) là yêu cầu tri thức hợp lệ; COK (Codified organisational knowledge) là tri thức tổ chức hợp lệ; DOKB (Distributed organisational knowledge base): cơ sở tri thức tổ chức phân bố; FKC (Falsified knowledge claim): yêu cầu tri thức

giả mạo; OK (Organisational knowledge): tri thức tổ chức; SKC (Surviving knowledge claim): yêu cầu tri thức tồn đọng; UKC (Undecided knowledge claim): Yêu cầu tri thức chưa quyết định. Trong hình vẽ, các khối bình hành (không là khối chữ nhật) chỉ dẫn tập tri thức.

Theo Hình 2.7, trong vòng đời của mình, tri thức doanh nghiệp qua ba giai đoạn chính.

Sáng tạo tri thức (Knowledge Production) là giai đoạn đầu tiên, trong đó do kết quả học tập của cá nhân và nhóm, do nhu cầu thông tin và phản hồi của vòng đời tri thức trước đây (bao gồm sự phát hiện vấn đề mới), yêu cầu tri thức được tạo ra sơ bộ. Sau đó yêu cầu này được đánh giá, nếu hợp lệ được chuyển sang giai đoạn sau (Tích hợp tri thức), nếu chưa hợp lệ được quay lại việc học bổ sung của cá nhân và nhóm để có được yêu cầu tri thức hợp lệ. Thông qua các tập tri thức doanh nghiệp sẵn có (SKC, FKC, UKC và thông tin liên quan), tri thức doanh nghiệp được tạo ra.



Hình 2.7. Vòng đời tri thức doanh nghiệp [Elroy02]

Tích hợp tri thức (Knowledge Integration) là giai đoạn tiếp theo, trong đó tri thức doanh nghiệp được phổ biến, được tìm

kiếm, được giảng dạy và được chia sẻ để tạo thành tri thức chủ quan và tri thức khách quan được tập hợp vào cơ sở tri thức doanh nghiệp phân bố để được đưa vào sử dụng trong môi trường quá trình kinh doanh.

Trong môi trường quá trình kinh doanh (Business Processing Environment), tri thức doanh nghiệp được sử dụng và tạo ra giá trị doanh nghiệp. Quá trình cộng tác tạo giá trị của doanh nghiệp và khách hàng cũng đưa ra các phản hồi từ môi trường quá trình kinh doanh tạo ra yêu cầu tri thức doanh nghiệp mới.

Quá trình vòng đời tri thức doanh nghiệp kết hợp với hệ thống khai phá dữ liệu hợp thành một hệ thống công nghệ tri thức trọn vẹn.

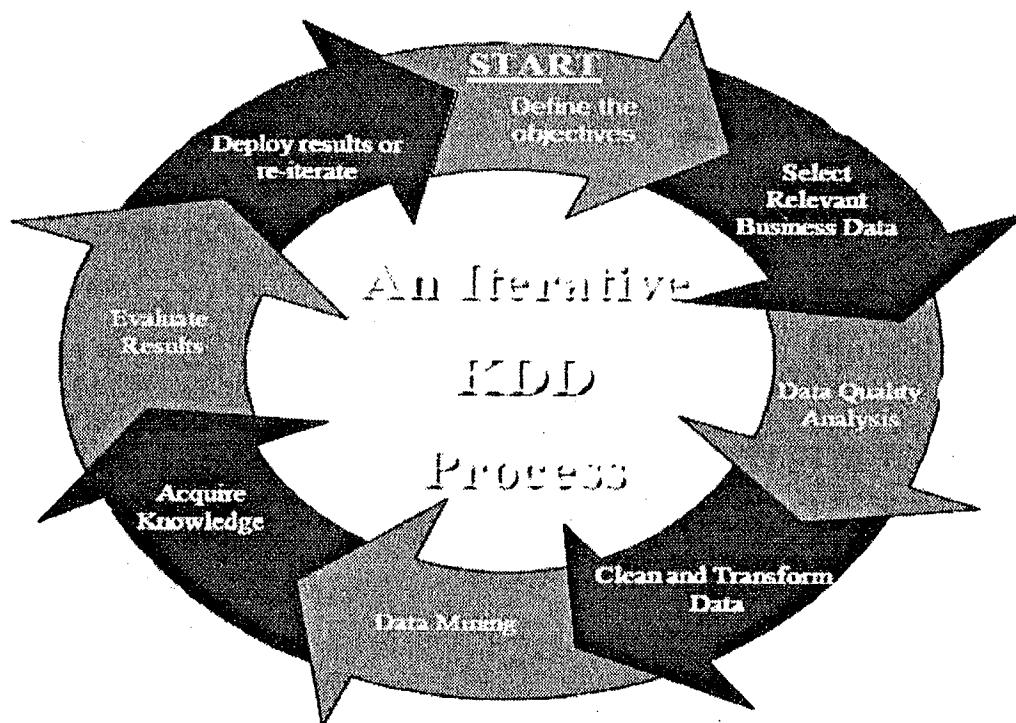
2.3. BÀI TOÁN PHÁT HIỆN TRI THỨC TỪ DỮ LIỆU

2.3.1. Sự tiến hóa của mô hình phát hiện tri thức

Chương 1 đã giới thiệu một mô hình KDD theo một tiếp cận mang tính thuần túy CNTT [FPS96] và đây được coi là một trong những mô hình hoàn chỉnh đầu tiên cho KDD. Như đã biết, mục tiêu cơ bản nhất của quá trình KDD là phát hiện ra các tri thức tiềm ẩn trong dữ liệu nhằm cung cấp các tri thức cho các tổ chức và cá nhân trong việc ra quyết định. Dù cho khai phá dữ liệu khoa học, công nghệ, đặc biệt là khai phá dữ liệu trong y sinh học đang phát triển mạnh mẽ [HG09], song lĩnh vực quản lý và kinh doanh luôn là miền ứng dụng quan trọng nhất của khai phá dữ liệu. Vì vậy, sự tiến hóa của mô hình phát hiện tri thức từ dữ liệu cũng theo hướng ngày càng gắn với quá trình quản lý và kinh doanh để tri thức được phát hiện ra trở thành tài nguyên phục vụ quá trình kinh doanh của doanh nghiệp (Hình 2.7). Một số mô hình được giới thiệu dưới đây cung cấp một số nét điển hình nhất về quá trình tiến hóa mô hình KDD.

Như đã giới thiệu, Usama Fayyad và cộng sự đã đưa ra một mô hình phát hiện tri thức từ dữ liệu [FPS96]. Nội dung các bước

thực hiện trong quá trình này đã được trình bày tại Chương 1. Sau này, mô hình khai phá dữ liệu do Usama Fayyad và cộng sự đề xuất được các tác giả khác gọi là mô hình phát hiện tri thức truyền thống. Mô hình khai phá dữ liệu truyền thống chưa nhấn mạnh định hướng kinh doanh của phát hiện tri thức từ dữ liệu dù rằng khi phân tích bước đặt bài toán phát hiện tri thức trong mô hình có đề cập tới mục tiêu phát hiện tri thức có bao gồm yếu tố kinh doanh.



Hình 2.8. Một mô hình phát hiện tri thức lặp, 1998 [CCG98]

2.3.1.1. Mô hình phát hiện tri thức lặp

Năm 1998, Collier K. và cộng sự tại Trung tâm hiểu dữ liệu (The Center for Data Inshight: CDI) tại Đại học Bắc Arizona, Mỹ (Northern Arizona University) [CCGMS98] đề nghị thay đổi mô hình phát hiện tri thức truyền thống thành mô hình phát hiện tri thức lặp (Hình 2.8). Trong mô hình truyền thống, Usama Fayyad và cộng sự cũng cho phép các bước của quá trình được thực hiện lặp một cách tùy ý. Mô hình lặp chỉ cho phép lặp lại sau khi đã hoàn thành chu trình thực hiện tất cả các bước.

Collier K. và cộng sự giải thích chi tiết nội dung các bước thực hiện trong mô hình phát hiện tri thức lặp như sau:

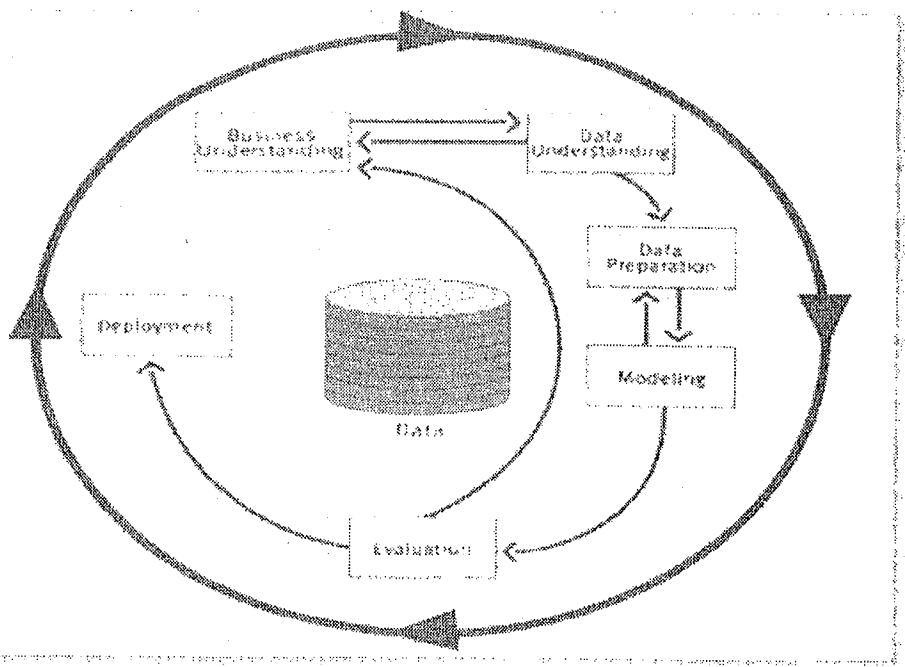
- Xác định mục tiêu kinh doanh. Bắt đầu với nhiều nhất ba mục tiêu kinh doanh để nghiên cứu có tính tập trung;
- Định danh dữ liệu doanh nghiệp mà chứa thông tin liên quan tới các mục tiêu kinh doanh đã được xác định;
- Khởi tạo tập dữ liệu mẫu chứa mọi thông tin liên quan;
- Định danh các chuyên gia miền lĩnh vực làm việc với nhóm thực nghiệm trong hệ thống phát hiện tri thức;
- Khởi tạo dữ liệu sao cho năng lực tính toán làm chủ được dữ liệu được khảo sát và thích hợp với công cụ phát hiện tri thức phù hợp mục tiêu kinh doanh;
- Chuyên gia miền ứng dụng làm việc với chuyên gia khai phá dữ liệu xác nhận bộ công cụ là thích hợp nhất với mục tiêu kinh doanh;
- Trích chọn quan hệ và mẫu từ tập dữ liệu kinh doanh;
- Chuyên gia miền ứng dụng làm việc với chuyên gia khai phá dữ liệu để xác định các quan hệ và mẫu thực sự liên quan tới mục tiêu kinh doanh. Kinh nghiệm tại CDI từ một số các dự án khai phá dữ liệu chỉ ra rằng một số kết quả kinh ngạc có thể xuất hiện ở bước này. Giả thiết cơ sở về cách thức của một thương vụ, cách thức của một thị trường hoặc cách thức hành vi của khách hàng có thể cần phải thay đổi.

Lưu ý rằng, nội dung các bước *làm sạch* và *chuyển dạng* dữ liệu, *khai phá dữ liệu*, *thu nhận tri thức* không có nhiều khác biệt so với mô hình truyền thống.

2.3.1.2. Mô hình chuẩn công nghiệp CRISP-DM

Trong khuôn khổ dự án chuẩn công nghiệp khai phá dữ liệu CRISP-DM (Cross-Industry Standard Process for Data Mining), Pete Chapman và cộng sự công bố tài liệu hướng dẫn về CRISP-DM [CCKKR00]. Hình 2.9 trình bày quy trình khai phá dữ liệu theo chuẩn công nghiệp. Chuẩn CRISP-DM cũng đặt nội dung "Hiểu kinh doanh" là giai đoạn đầu tiên của quá trình khai phá dữ

liệu. Chi tiết các bước trong quy trình khai phá dữ liệu theo chuẩn CRISP-DM như sau:



Hình 2.9. Chuẩn công nghiệp khai phá dữ liệu CRISP-DM, 2000 [CCKKR00]

- Hiểu kinh doanh (Business understanding): Giai đoạn này ban đầu tập trung vào sự hiểu biết các mục tiêu và các yêu cầu từ góc độ kinh doanh của dự án khai phá dữ liệu, sau đó chuyển đổi tri thức này thành một định nghĩa bài toán khai thác dữ liệu và một kế hoạch sơ bộ được thiết kế để đạt được các mục tiêu.

- Hiểu dữ liệu (Data understanding): Giai đoạn hiểu dữ liệu bắt đầu với một bộ sưu tập dữ liệu ban đầu và tiến hành các hoạt động để làm quen với dữ liệu, xác định các vấn đề chất lượng dữ liệu, để khám phá những hiểu biết đầu tiên vào các tập dữ liệu hoặc phát hiện các tập con dữ liệu thú vị nhằm hình thành giả thuyết cho thông tin ẩn. Tri thức kinh doanh có từ giai đoạn hiểu kinh doanh định hướng việc hiểu dữ liệu. Đồng thời, qua phân tích dữ liệu để hiểu dữ liệu có thể phản hồi, phối hợp với nội dung hiểu kinh doanh để làm rõ bài toán khai phá dữ liệu, mục tiêu và kế hoạch thực hiện.

- Chuẩn bị dữ liệu (Data preparation): Từ các bộ dữ liệu thô ban đầu, giai đoạn chuẩn bị dữ liệu bao gồm tất cả các hoạt động nhằm xây dựng các tập dữ liệu cuối cùng làm đầu vào cho công cụ mô hình hóa. Chuẩn bị dữ liệu bao gồm các hoạt động lập bảng, ghi lại, lựa chọn thuộc tính cũng như chuyển đổi, và làm sạch dữ liệu cho các công cụ mô hình hóa. Các thao tác chuẩn bị dữ liệu có thể được thực hiện nhiều lần và không theo một thứ tự quy định.

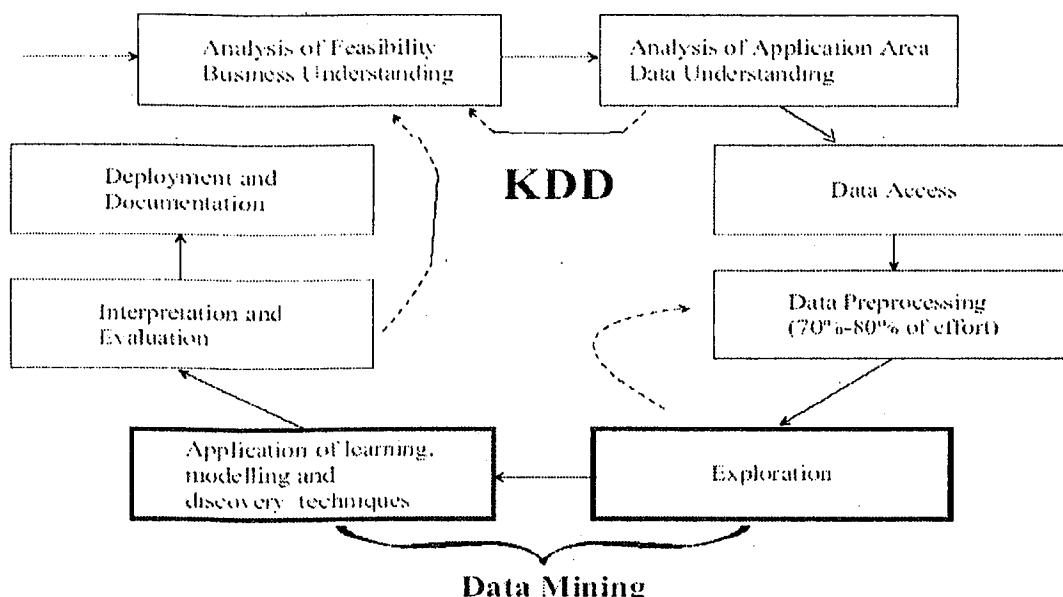
- Mô hình hóa (Modeling): Trong giai đoạn này, các kỹ thuật mô hình khác nhau được lựa chọn và áp dụng. Các thông số của các mô hình được xác định nhằm đạt tới giá trị tối ưu. Thông thường, một số kỹ thuật được sử dụng cho các loại dữ liệu với cùng một bài toán khai thác dữ liệu. Một số kỹ thuật đòi hỏi các yêu cầu cụ thể về dạng thức của dữ liệu đầu vào. Đưa dữ liệu về dạng thức phù hợp với các kỹ thuật (và công cụ) khai phá dữ liệu là một công việc được thực hiện trong giai đoạn chuẩn bị dữ liệu. Mô hình hóa và chuẩn bị dữ liệu có thể được thực hiện lặp một số lần nhằm đạt được mô hình có kết quả tối ưu.

- Đánh giá (Evaluation): Ở giai đoạn này, mô hình (có thể một số mô hình) kết quả với mục tiêu chất lượng cao theo góc độ phân tích dữ liệu được tìm ra. Trước khi đưa mô hình vào triển khai trong thực tiễn kinh doanh, cần đánh giá mô hình kết quả kỹ lưỡng hơn và xem xét các bước đã được thực hiện để xây dựng mô hình nhằm có được niềm tin chắc chắn rằng mô hình kết quả đạt được các mục tiêu kinh doanh theo đúng cách thức.

Một mục tiêu quan trọng của hoạt động đánh giá là xác định có hay không vấn đề kinh doanh quan trọng nào đó đã không được xem xét một cách toàn diện. Vào cuối của giai đoạn này, một quyết định về việc sử dụng các kết quả khai thác dữ liệu có thể đạt được.

- Triển khai (Deployment): Nói chung, tạo ra mô hình chưa phải là kết thúc dự án khai phá dữ liệu. Tri thức được phát hiện cần phải được tổ chức và trình bày theo cách mà khách hàng có thể triển khai sử dụng tri thức đó. Giai đoạn triển khai thường bao gồm việc áp dụng mô hình "sống" (thời gian thực) vào quyết

định của tổ chức triển khai dự án. Tuy nhiên, tùy thuộc vào yêu cầu, giai đoạn triển khai có thể được đơn giản như tạo ra một báo cáo hoặc phức tạp như thực hiện một quá trình khai thác dữ liệu lặp lại trên toàn doanh nghiệp. Trong nhiều trường hợp, khách hàng chủ không phải các nhà phân tích dữ liệu, thực hiện các bước triển khai. Tuy nhiên, ngay cả khi các nhà phân tích không thực hiện công việc triển khai, một yêu cầu quan trọng đối với các nhà phân tích dữ liệu là họ phải giúp khách hàng tường minh tiên liệu được những hành động mà họ cần phải được thực hiện để các mô hình đã được tạo ra thực sự được sử dụng.

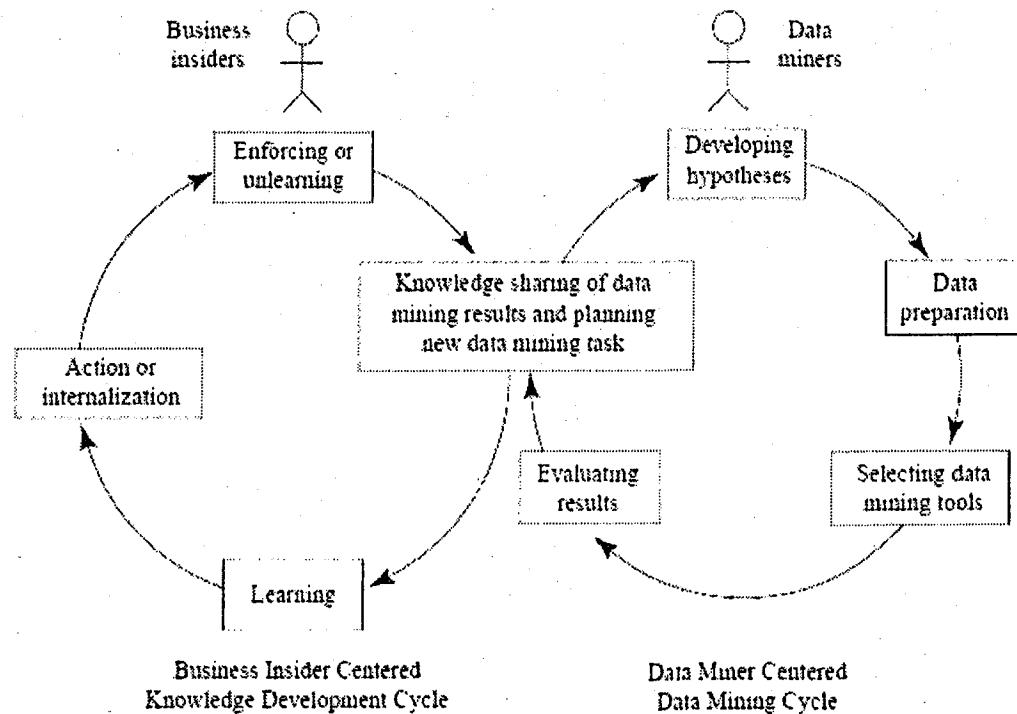


Hình 2.10. Một mô hình KDD, 2000 [Nauck00]

Trong [Nauck00], Detlef D.Nauck giới thiệu một mô hình phát hiện tri thức như trình bày ở Hình 2.10. Tương tự như mô hình CRISP-DM, mô hình này cũng có xuất phát điểm từ nhu cầu kinh doanh và phân tích dữ liệu miền ứng dụng có tương tác lẫn nhau với phân tích miền ứng dụng kinh doanh. Tác giả nhấn mạnh công việc tiền xử lý dữ liệu đòi hỏi khoảng 70-80% công sức của toàn bộ quá trình phát hiện tri thức (Chương 3 của sách này sẽ trình bày các nội dung chi tiết của hoạt động chuẩn bị dữ liệu). “Bước” khai phá dữ liệu bao gồm hai bài toán con là khảo sát và áp

dụng kỹ thuật học máy, mô hình hóa và phát hiện tri thức. Khảo sát có tương tác phản hồi với công việc chuẩn bị dữ liệu. Sau khi được trực quan hóa và đánh giá, tri thức được phát hiện sẽ được đưa vào ứng dụng và được văn bản hóa, bổ sung tri thức miền ứng dụng. Mô hình Detlef D.Nauck giới thiệu làm rõ hơn nội dung một số bước so với mô hình CRISP-DM.

2.3.2.3. Mô hình phát hiện tri thức kết hợp khung nhìn kinh doanh

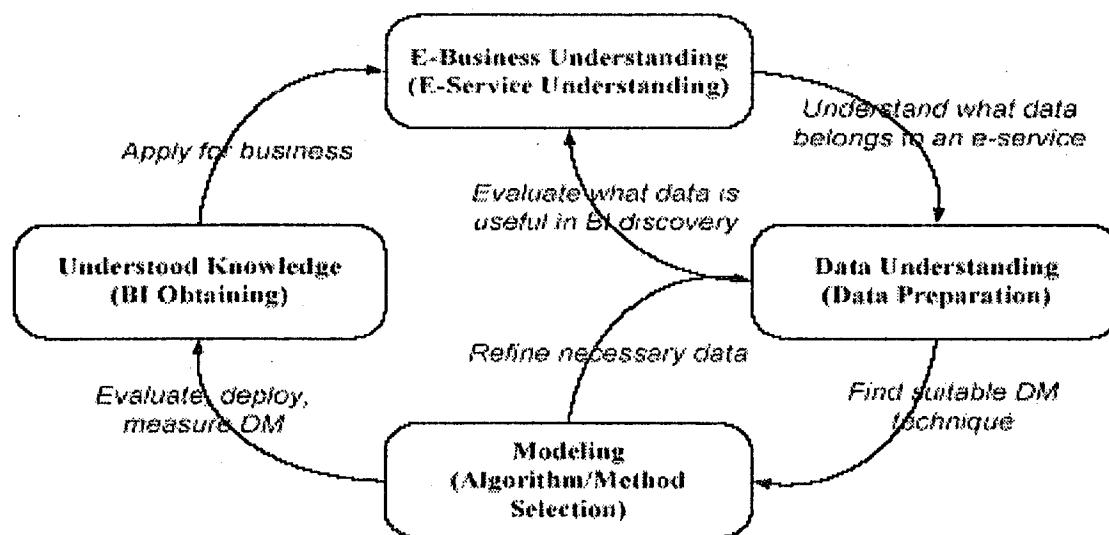


Hình 2.11. Một mô hình quản lý tri thức, 2008 [WW08]

Năm 2008, Wang, H. and S. Wang [WW08] đề nghị một mô hình quản lý tri thức (knowledge management) là tích hợp mô hình phát hiện tri thức định hướng khai phá dữ liệu và mô hình phát triển tri thức định hướng kinh doanh (Hình 2.11). Hai kiểu tác nhân chủ chốt trong mô hình này là nhân viên khai phá dữ liệu (data miner) và nhân viên kinh doanh của doanh nghiệp (business insider). Như vậy, nhân viên khai phá dữ liệu có thể là người của doanh nghiệp hoặc không. Giống như các mô hình đã nói, điểm đầu và điểm cuối của chu trình khai phá dữ liệu là sự tương tác với chu trình phát triển tri thức kinh doanh về kế hoạch bài toán khai phá dữ liệu mới và chia sẻ tri thức kết quả của khai

phá dữ liệu. Trong chu trình phát triển tri thức doanh nghiệp, tri thức kết quả khai phá dữ liệu được học tập nội bộ, được áp dụng và tiếp thu để tăng cường tài nguyên tri thức doanh nghiệp.

2.3.1.4. Mô hình phát hiện tri thức hướng thông minh doanh nghiệp



Hình 2.12. Mô hình phát hiện tri thức hướng thông minh doanh nghiệp, 2009 [HF09]

Trong [HF09], Yang Hang và Simon Fong trình bày một hệ thống ứng dụng khai phá dữ liệu trong miền ứng dụng thương mại điện tử. Các tác giả trình bày mô hình khung bốn tầng gồm tầng dữ liệu (data layer), tầng phương pháp (method layer), tầng dịch vụ điện tử (e-service layer) và tầng tri thức (knowledge layer). Tri thức được phát hiện trong hệ thống là tri thức dạng thông minh doanh nghiệp (Business Intelligence). Quá trình khai phá dữ liệu định hướng thông minh doanh nghiệp (BI - Driven Data Mining) cho thương mại điện tử được biểu diễn ở Hình 2.12. Trong mô hình này, xuất phát từ mục tiêu kinh doanh thương mại điện tử, một quá trình khai phá dữ liệu định hướng thông minh doanh nghiệp được thi hành để nhận được tri thức để áp dụng vào quá trình kinh doanh.

Trên hình vẽ, quá trình phát hiện tri thức được thi hành theo bốn pha chính.

- Pha hiểu miền ứng dụng thương mại điện tử/dịch vụ điện tử là pha đầu tiên của quá trình. Để khai phá dữ liệu định hướng thông minh doanh nghiệp miền ứng dụng thương mại điện tử thực sự hiệu quả thì cần hiểu rõ (có được tri thức miền ứng dụng) về dịch vụ điện tử được quan tâm. Mục tiêu khai phá dữ liệu dịch vụ điện tử này được xác định. Những tri thức bài toán về dịch vụ điện tử cho phép xác định được phạm vi và tính chất của tập dữ liệu cần thiết cho bài toán khai phá, làm cơ sở định hướng cho khâu chuẩn bị dữ liệu.

- Trong pha hiểu dữ liệu, hoạt động chuẩn bị dữ liệu được tiến hành theo định hướng từ tri thức bài toán. Trong quá trình chuẩn bị dữ liệu, tri thức miền ứng dụng vẫn được huy động để đánh giá tính hiệu quả của dữ liệu được chuẩn bị.

- Tại pha mô hình hóa, các thuật toán/phương pháp phù hợp với bài toán được chọn và thực hiện để xây dựng được mô hình khai phá dữ liệu phù hợp. Công việc mô hình hóa cũng đặt ra yêu cầu chỉnh lý lại dữ liệu cần thiết.

- Trong pha thu nhận tri thức, kết quả thực hiện thuật toán khai phá dữ liệu được đánh giá, đo lường để chọn ra được tri thức thông minh doanh nghiệp có giá trị tương ứng với dịch vụ điện tử. Sau đó, tri thức thông minh doanh nghiệp kết quả được áp dụng trong kinh doanh.

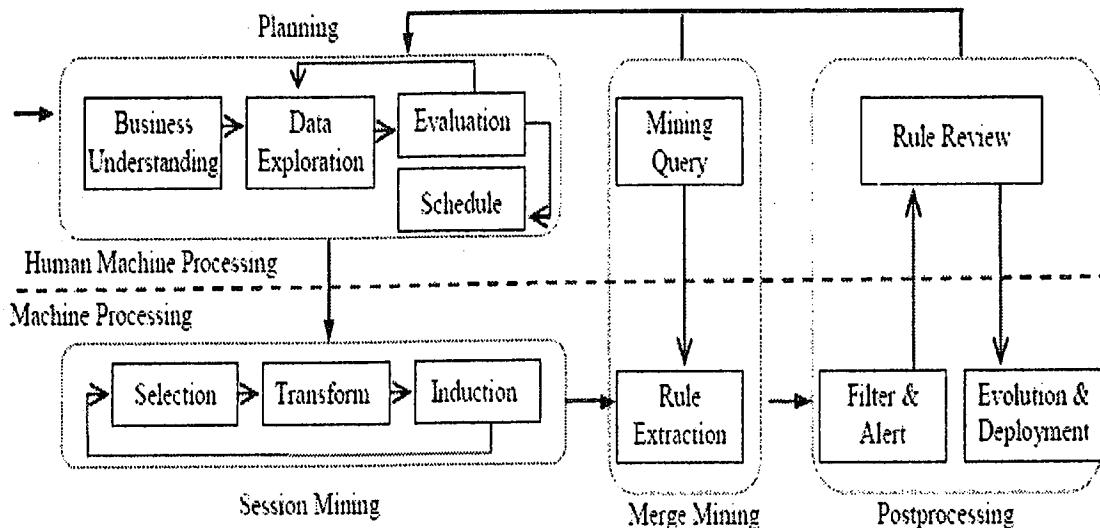
2.3.1.5. Mô hình gắn kết phát hiện tri thức từ dữ liệu

Phát hiện tri thức gắn kết từ CSDL (Cohesive Knowledge Discovery in Database: C-KDD) được quan tâm trong thời gian gần đây. Mô hình phát hiện tri thức C-KDD được biểu diễn tại Hình 2.13. Quá trình C-KDD gồm bốn giai đoạn: lập kế hoạch, khai phá phiên, hợp nhất khai phá, và hậu xử lý.

Trong giai đoạn *lập kế hoạch*, quá trình KDD bắt đầu với việc hiểu biết kinh doanh, bao gồm mục tiêu kinh doanh và lôgic kinh doanh. Thông qua thăm dò và thử nghiệm, phát hiện ra các mục đích, dữ liệu kinh doanh, và quá trình tiếp theo được xác định; các đặc tả lịch nhiệm vụ (task schedule: TS) phát hiện tri

thức được sinh ra. Tri thức miền bản thể học được sử dụng để loại bỏ các thuộc tính không phù hợp, cập nhật các yếu tố kinh doanh tiên nghiệm mơ hồ, suy luận các thuộc tính trừu tượng khác v.v.

Hơn nữa, tập các thuộc tính dữ liệu hợp lệ, các bước quá trình, và các thuật toán được cấu thành theo thứ tự dựa trên ước muốn của người sử dụng, bằng một bản thể học (ontology) khai phá dữ liệu.



Hình 2.13. Mô hình quá trình C-KDD, 2010 [Pan10]

Giai đoạn *khai thác phiên* thực hiện việc chọn dữ liệu - chuyển dạng dữ liệu - tiền khai phá dữ liệu và đạt được khai phá dữ liệu bộ phận. Giai đoạn này chú trọng quy nạp các luật cục bộ và tinh, và tiến hành quy nạp gia tăng dữ liệu theo chu kỳ, chẳng hạn như tháng. Do các hàm đã được đặc tả trong TS, chúng định kỳ thực hiện lặp trên dữ liệu gia tăng theo tần số hoặc theo điều kiện kích hoạt, và cho kết quả dưới hình thức một túi luật (rule bin: RB). Kiến thức bản thể học được sử dụng để hỗ trợ việc xác định các tính năng, các tham số v.v. được lựa chọn.

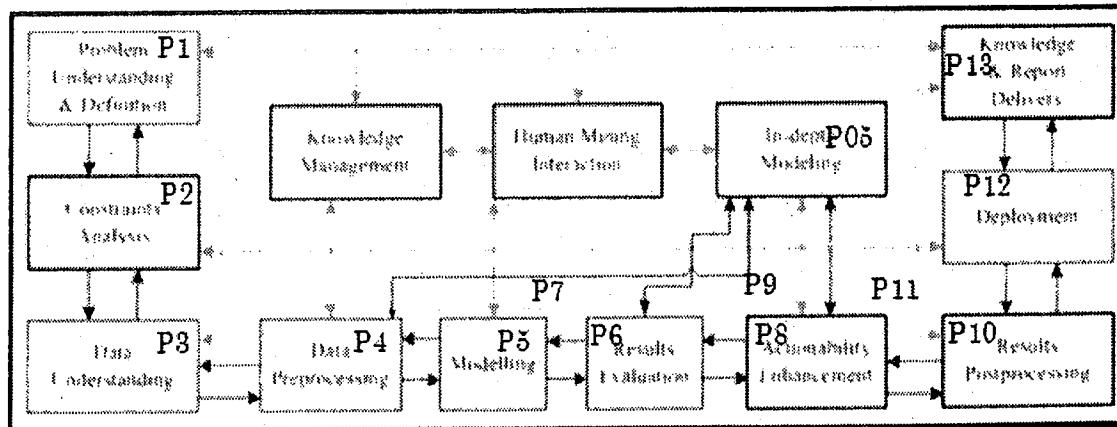
Giai đoạn *hợp nhất khai phá* được khởi động bằng các truy vấn khai phá dữ liệu hoặc bằng một sự kiện kích hoạt. Các nội dung truy vấn được lên danh sách khi tham chiếu với TS; người sử dụng có thể cam kết chúng theo yêu cầu của mình. Một sự kiện kích hoạt xuất hiện do gia tăng thời gian hoặc luật. Nó chú trọng vào việc phát hiện các luật tổng thể và động, trong một mô hình tương tác, các luật được hợp nhất và tinh chỉnh từ một số cơ sở luật. Các tham số và ràng buộc được bổ sung từ tri thức bản thể học.

Giai đoạn *hậu xử lý* bắt đầu bằng việc làm phù hợp luật phát hiện được với tri thức đã biết, lọc bỏ các luật vô dụng, sau đó phân lớp và xếp hạng các kết quả hấp dẫn theo độ hấp dẫn. Khi đạt tới một ngưỡng điểm tối hạn, một cảnh báo sẽ được kích hoạt. Khi đó người dùng có thể xem xét lại và xác nhận các phát hiện này. Nó cũng tích hợp các hiểu biết thú vị mới với tri thức đã biết nhằm thi hành việc tiến hóa và trình bày tri thức. Do đó nó tạo thành một giải pháp lặp đóng để duy trì quá trình phát hiện tri thức liên tục.

Khi không đáp ứng việc đảm bảo các ứng dụng thông minh hoặc xem xét lại luật, dòng quá trình quay lại giai đoạn lập kế hoạch để tái khám phá dữ liệu. Cuối cùng, các kết quả này được trả về cho người dùng cuối. Trong hệ thống C-KDD, mỗi thành phần là một đại lý thông minh, có các bản thể học thăm dò và tri thức đã biết thông qua dịch vụ bản thể học.”.

2.3.1.6. Mô hình khai phá dữ liệu hướng miền ứng dụng

Gần đây, khai phá dữ liệu hướng miền ứng dụng (Domain Driven Data Mining: D3M) là một trong những khuynh hướng nghiên cứu nổi bật của khai phá dữ liệu. Longbing Cao và cộng sự [CYZZ10] đề nghị mô hình quá trình khai phá dữ liệu hướng miền ứng dụng như được thể hiện trong hình 2.14. Các thành phần chức năng chính của mô hình được làm nổi bật bằng các hộp có viền dày, thể hiện những giai đoạn cụ thể D3M.



Hình 2.14. Mô hình quá trình khai phá dữ liệu hướng miền ứng dụng, 2010 [CYZ10]

Mô hình này cho một khung nhìn chi tiết hơn về quá trình phát hiện tri thức thông qua pha thực hiện, được ký hiệu từ P1 tới P13 như trên hình vẽ (P05 và P07 là các phương án thay thế cho P5 hoặc P7). Mỗi bước của quá trình D3M có thể liên quan đến sự thông minh khắp nơi và tương tác với người dùng doanh nghiệp và/hoặc với các chuyên gia miền. Nội dung chi tiết của mỗi bước trong vòng đời của quá trình D3M được trình bày như dưới đây, nhưng cần lưu ý rằng trình tự các bước không là cứng nhắc, một số giai đoạn có thể được bỏ qua hoặc có sự chuyển đổi qua lại để thích ứng với một vấn đề bài toán trong thực tiễn:

P1. Hiểu vấn đề (định danh và xác định các vấn đề, bao gồm cả phạm vi của nó và những thách thức...);

P2. Phân tích ràng buộc (định danh ràng buộc xung quanh các vấn đề ở trên, từ dữ liệu, miền ứng dụng, tính thú vị và cách phân bố);

P3. Định nghĩa các mục tiêu phân tích, và xây dựng đặc trưng (định nghĩa mục tiêu khai phá dữ liệu, và các đặc trưng được lựa chọn phù hợp hoặc xây dựng để đạt được các mục tiêu);

P4. Tiền xử lý dữ liệu (trích chọn, chuyển đổi và tải dữ liệu, nói riêng, chuẩn bị dữ liệu chẳng hạn như xử lý dữ liệu mốc tích và riêng tư);

P5. Lựa chọn phương pháp và mô hình hóa (lựa chọn được các mô hình và phương pháp thích hợp để đạt được các mục tiêu trên); hoặc

P05. Mô hình hóa chuyên sâu (áp dụng mô hình hóa chuyên sâu bằng cách sử dụng nhiều mô hình hiệu quả tiết lộ cốt lõi của vấn đề, hoặc dùng khai phá đa bước, khai phá kết hợp);

P6. Phân tích và đánh giá kết quả chung ban đầu (phân tích /đánh giá các phát hiện ban đầu);

P7. Là hoàn toàn hợp lý khi mà mỗi giai đoạn từ P1 có thể được lặp đi lặp lại thông qua phân tích ràng buộc và tương tác với các chuyên gia miền ứng dụng theo phương thức quay lui và xem xét; hoặc

P07. Khai phá chuyên sâu về kết quả chung ban đầu khi áp dụng;

P8. Đo lường và nâng cao khả năng hành động (kiểm tra tính thú vị theo quan điểm cả về kỹ thuật và kinh doanh, và tăng cường hiệu suất bằng cách áp dụng phương pháp hiệu quả hơn).

P9. Thực hiện qua lại giữa P7 và P8;

P10. Hậu xử lý kết quả (hậu phân tích hoặc hậu khai phá dữ liệu các kết quả ban đầu);

P11. Xem xét lại các giai đoạn từ P1 có thể được đòi hỏi;

P12. Triển khai (triển khai các kết quả vào các ngành kinh doanh);

P13. Cung cấp tri thức và báo cáo tổng hợp để ra quyết định thông minh (tổng hợp phát hiện cuối cùng thành báo cáo ra quyết định sẽ được chuyển giao cho người kinh doanh).

2.3.2. Về bài toán khai phá dữ liệu

Khai phá dữ liệu và phát hiện tri thức trong dữ liệu là vấn đề tăng cường tài nguyên tri thức của tổ chức, và vì vậy, đây là một vấn đề chiến lược. Nói riêng, trong doanh nghiệp, bài toán khai phá dữ liệu được đặt ra từ nhu cầu kinh doanh mà không phải là nhu cầu của công nghệ. Một số định hướng đầu tư CNTT đã được giới thiệu.

Quá trình tiến hóa mô hình khai phá dữ liệu khẳng định rằng công việc xác định bài toán khai phá dữ liệu được đặt lên hàng đầu. Các mô hình đã nói cũng nhất quán tiếp cận bài toán khai phá dữ liệu từ nhu cầu phát triển của đơn vị, nói riêng trong các doanh nghiệp thì đây là nhu cầu kinh doanh. Hiểu miền ứng dụng có tính quyết định cho việc xác định bài toán khai phá dữ liệu. Chuyên gia miền lĩnh vực không chỉ là bộ phận chủ chốt cho xác định bài toán khai phá dữ liệu mà còn trong cả toàn bộ quá trình phát hiện tri thức, tăng cường tài nguyên tri thức cho doanh nghiệp [WW08, HF09, Pan10, CYZ10]. Từ phương diện của chuyên gia khai phá dữ liệu bên ngoài, xác định đúng đối tượng mục tiêu chuyên gia miền lĩnh vực của doanh nghiệp là vấn đề then chốt để triển khai dự án khai phá dữ liệu. Khi xác định bài

toán khai phá dữ liệu, chuyên gia khai phá dữ liệu cần tránh định hướng tiếp cận theo phương diện công nghệ. Xác định đúng bài toán đúng đắn là một yếu tố quyết định thành công của dự án khai phá dữ liệu (Chương 10).

Trong [WB98], Christopher Westphal và Teresa Blaxton đưa ra một số khuyến nghị khi bắt đầu tiến hành một dự án khai phá dữ liệu:

- Khi đặt ra một bài toán khai phá dữ liệu thì cần tránh đưa ra sự kỳ vọng quá đáng về kết quả. Tương tự như sự kỳ vọng quá đáng đối với CNTT, một lĩnh vực đang nổi như khai phá dữ liệu có xu hướng tạo ra một độ ảo tưởng nào đó đối với một bộ phận cá nhân và tổ chức. Khi đặt ra bài toán khai phá dữ liệu, có thể có một ước đoán thô nào đó về kết quả phát hiện tri thức, tuy nhiên, đây mới chỉ là sự ước đoán ban đầu. Khai phá dữ liệu là một quá trình phát hiện các mẫu mới và xu hướng mới, tiềm ẩn trong dữ liệu, mà đã là "mới, tiềm ẩn" thì không thể tiết lộ trước một cách đầy đủ kết quả khai phá dữ liệu. Mặt khác, khai phá dữ liệu là quá trình tương tác khám phá, trong đó tương tác khám phá với các chuyên gia miền ứng dụng có ý nghĩa đặc biệt quan trọng. Huy động tri thức chuyên gia là vấn đề khó trong công nghệ tri thức nói chung và trong khai phá dữ liệu nói riêng.

Theo Christopher Westphal và Teresa Blaxton, khai phá dữ liệu là một quá trình độc đáo và đầy thử thách, đòi hỏi phải sử dụng kết hợp các phương pháp và công nghệ. Tuy khai phá dữ liệu là một quá trình thi hành bộ phận của vòng đời tri thức song chuyên gia khai phá dữ liệu không thể lặp đi lặp lại một kịch bản mà cần phải không ngừng cải tiến cách tiếp cận dựa trên các mẫu kết quả đã được phát hiện.

- Khi đặt ra bài toán khai phá dữ liệu thì cần tính đến tính thực tiễn của bài toán. Đầu tiên, vấn đề cần giải quyết là kinh phí đầu tư cho một dự án khai phá dữ liệu. Theo kinh nghiệm của Christopher Westphal và Teresa Blaxton, các công ty thường đầu tư cho khai phá dữ liệu vào khoảng 15%-20% giá trị làm giảm

thiệt hại được ước tính hoặc cải tiến được dự kiến. Thứ hai, tính kịp thời cần là một phẩm chất của các chuyên gia khai phá dữ liệu. Một dự án khai phá dữ liệu cần cho kết quả trong thời gian tính theo ngày hoặc cùng lăm tính theo tuần. Yêu cầu khai phá dữ liệu trong một thời hạn ngắn như vậy với bối cảnh khối lượng dữ liệu lớn, vì vậy không thể thực hiện khai phá dữ liệu trên toàn bộ dữ liệu và việc chọn lựa dữ liệu có vai trò rất quan trọng. Việc chọn lựa dữ liệu gắn kết với mục tiêu phát hiện tri thức (trong doanh nghiệp là mục tiêu kinh doanh), vì vậy, ý kiến chuyên gia nội bộ tạo thuận lợi cho việc hạn chế phạm vi dữ liệu. Hơn nữa, giao tiếp tốt với chuyên gia nội bộ giúp xác định tốt mục tiêu của khai phá dữ liệu. Thứ ba, khi thực hiện bài toán khai phá dữ liệu cũng cần dự đoán và vượt qua rào cản về thể chế. Việc tiếp nhận và sử dụng tri thức mới (mẫu mới, dự đoán mới) có thể khác lạ so với nội dung thể chế hiện hành.

- Truyền thông, bảo hiểm, bán lẻ, tài chính – ngân hàng, thương mại, hoặc hoạt động vận chuyển có những vùng hoạt động dễ bị tổn thương, mà ở đó gian lận có thể xảy ra. Gian lận không bị phát hiện bởi vì chúng được ẩn dật khôn khéo trong một lượng lớn các giao dịch bình thường. Tính mới của kết quả khai phá dữ liệu là điều cốt lõi song cần phát hiện các mẫu mới hoặc phổ biến hoặc hiếm.

2.4. ĐỘ ĐO HẤP DẪN TRONG KHAI PHÁ DỮ LIỆU

Tại Chương 1, khi giải thích nội dung định nghĩa KDD, các độ đo cho tính có giá trị, tính mới, tính hữu ích tiềm năng, và đặc biệt là tính hấp dẫn của một mẫu được giả định là đã có. Một mẫu phát hiện được có độ hấp dẫn vượt qua một ngưỡng cho trước thì nó được coi là tri thức mới được phát hiện. Độ đo hấp dẫn của một mẫu là *độ đo tổng thể về mẫu* là sự kết hợp của các tiêu chí *giá trị, mới, hữu ích và dễ hiểu*. Nội dung, tính chất của độ đo hấp dẫn chưa được đề cập.

Đo lường tri thức và đo lường kinh tế tri thức là những bài toán khó [Grube09, OEC96, CD05] và đo lường độ hấp dẫn của

mẫu trong khai phá dữ liệu cũng không nằm ngoài quy luật đó. Không có một độ đo hấp dẫn chung cho mẫu được phát hiện mà trong mỗi ngữ cảnh ứng dụng cần xác định các độ đo hấp dẫn phù hợp nhất. Tri thức được phát hiện qua khai phá dữ liệu được xác định dựa trên nhiều yếu tố ngữ cảnh ứng dụng, điển hình là yếu tố về loại bài toán khai phá dữ liệu. Mỗi loại bài toán khai phá dữ liệu có một lớp độ đo hấp dẫn phổ biến, chẳng hạn như khai phá luật kết hợp có hai độ đo phổ biến là độ hỗ trợ (support) và độ tin cậy (confidence) hoặc phân lớp dữ liệu có một số độ đo phổ biến là độ hồi tưởng (recall), độ chính xác (precision) và độ đo F, thuật toán phân lớp cây quyết định còn sử dụng các độ đo Gini hoặc độ đo lợi ích thông tin (information gain) để lựa chọn thuộc tính tốt... Độ đo hấp dẫn còn được sử dụng trong các bước khác của quá trình phát hiện tri thức, trong đó để việc hiểu dữ liệu hoặc lựa chọn thuộc tính cũng cần các độ đo hỗ trợ cho mẫu phát hiện được hấp dẫn. Đồng thời, khai phá dữ liệu loại này lại có thể sử dụng độ đo hấp dẫn của kiểu khai phá dữ liệu loại khác, chẳng hạn như, phân cụm có thể được thừa kế độ đo hấp dẫn của phân lớp.

Đo lường tính hấp dẫn của mẫu được phát hiện là một nội dung nghiên cứu tích cực và quan trọng trong khai thác dữ liệu và phát hiện tri thức từ dữ liệu. Nhiều công trình nghiên cứu khái quát và chuyên sâu về nội dung này, chẳng hạn [Garry05, Grube09, HGEK07, Yao03, HZ10, GH06, ZZNS09], đã được công bố. Dù chưa có sự công nhận rộng rãi cho một định nghĩa về độ đo hấp dẫn nhưng các tiêu chí cần đạt được của một mẫu hấp dẫn lại nhận được sự đồng thuận cao. Độ đo hấp dẫn cần đảm bảo tri thức được phát hiện là các mẫu có tính súc tích (conciseness), tính phổ dụng/bao trùm (*Generality/coverage*), tính tin cậy (*reliability*), tính đặc thù (*peculiarity*), tính đa dạng (*diversity*), tính mới lạ (*novelty*), tính ngạc nhiên (*surprisingness*), tính tiện ích (*utility*), và tính hành động (*actionability*). Nội dung của chín tính chất nói trên được trình bày như dưới đây [GH06].

Tính súc tích: Mẫu là súc tích nếu nó có chứa tương đối ít các cặp giá trị thuộc tính và một tập các mẫu là súc tích nếu nó chứa

tương đối ít các mẫu. Một mẫu hoặc tập mẫu súc tích là tương đối dễ dàng để hiểu và ghi nhớ và do đó được bổ sung dễ dàng hơn tri thức của người dùng. Ví dụ, hạn chế chỉ tìm kiếm các luật mạnh trong khai phá luật kết hợp, tìm cây tốt nhất có thể được trong phân lớp cây quyết định là những ví dụ về tìm tập mẫu súc tích. Độ đo F trong phân lớp dữ liệu nhằm đảm bảo các mẫu phân lớp có tính súc tích.

Tính phổ dụng/tính bao trùm: Một mẫu là phổ dụng nếu nó phủ một tập con lớn của tập dữ liệu theo nghĩa tập bản ghi phù hợp với mẫu trong tập toàn bộ dữ liệu chiếm một tỷ lệ lớn. Khi đó, mẫu phổ dụng sẽ đặc tả nhiều thông tin trong tập dữ liệu và vì vậy mẫu có xu hướng trở nên hấp dẫn hơn. Trong khai phá luật kết hợp, độ hỗ trợ (support) được đặt ra nhằm đảm bảo tri thức, luật tìm được có tính phổ dụng. Tập mục phổ biến là mẫu phổ dụng khi độ hỗ trợ của nó vượt qua một ngưỡng độ tối thiểu cho trước được gọi là độ hỗ trợ tối thiểu. Trong phân lớp Bayes, ngưỡng quyết định phân lớp được đặt ra đảm bảo một lớp chứa các bản ghi "phổ dụng" thuộc về nó. Tính phổ dụng thường xảy ra đồng thời với tính súc tích bởi vì các mẫu súc tích có xu hướng phổ dụng hơn các mẫu không súc tích.

Tính tin cậy: Một mẫu là tin cậy nếu mối quan hệ mà mẫu mô tả cho phép đạt một tỷ lệ cao khi đưa ra áp dụng. Ví dụ, một luật phân lớp là đáng tin cậy nếu dự đoán của nó chính xác cao, và một luật kết hợp là tin cậy nếu nó có độ tin cậy cao. Trong khai phá luật kết hợp, nhiều độ đo xác suất, thống kê, và thu hồi thông tin đã được đề xuất để đo độ tin cậy của các luật.

Tính đặc thù: Một mẫu có tính đặc thù nếu nó "xa" các mẫu được phát hiện khác theo một độ đo khoảng cách nào đó. Mẫu đặc thù được tạo ra từ dữ liệu đặc thù (hoặc ngoại lai), tương đối ít về số lượng và khác biệt đáng kể với phần còn lại của dữ liệu. Mẫu đặc thù có thể chưa được người sử dụng hình dung tới cho nên nó hấp dẫn. Trong khai phá luật kết hợp, luật hiếm (rare rule) là loại luật kết hợp có tính đặc thù.

Tính đa dạng: Tính đa dạng của mẫu thể hiện rằng các thành phần của nó khác biệt đáng kể với các thành phần khác, và một tập mẫu là đa dạng nếu các mẫu trong tập là khác biệt nhau đáng kể. Đa dạng là một tiêu chí phổ biến để đo lường tính hấp dẫn của tóm tắt dữ liệu: Một bản tóm tắt có thể được coi là đa dạng nếu phân bố xác suất của nó là khác biệt so với phân phối đồng nhất. Người dùng thường có xu hướng giả định bản tóm tắt giữ một phân phối đồng nhất cho nên bản tóm tắt đa dạng trở nên hấp dẫn.

Tính mới lạ: Mẫu là "mới lạ" cho một người nếu người đó không biết nó trước và không thể suy ra nó từ các mẫu khác đã biết. Đo lường tính mới lạ có đôi chút khác biệt với đo lường một số tiêu chí hấp hẫn khác có nghĩa là không thể đưa ra một ngưỡng để đo lường tính mới lạ. Thứ nhất, hệ thống khai phá dữ liệu không trình diễn mọi thứ mà người sử dụng đã biết: tính mới lạ không thể đo một cách rõ ràng khi tham chiếu tới tri thức miền ứng dụng của người sử dụng. Thứ hai, hệ thống khai thác dữ liệu không thể trình bày những điều mà người dùng chưa biết: tính mới lạ không thể đo một cách rõ ràng khi tham chiếu ngoài tri thức miền của người dùng. Thay vào đó, mẫu mới lạ được phát hiện thông qua (1) xác định rõ ràng cho người sử dụng thế nào là một mẫu mới lạ; (2) đưa ra một thông báo mẫu được phát hiện không thể được suy ra và không mâu thuẫn với các mẫu được phát hiện trước. Trong trường hợp thứ hai, các mẫu phát hiện trước được coi như một xấp xỉ với tri thức của người sử dụng.

Tính kinh ngạc: Mẫu là kinh ngạc (hoặc đột xuất) nếu nó mâu thuẫn với tri thức hiện có hoặc kỳ vọng của một người. Một mẫu được phát hiện khác biệt với một mẫu chung đã được phát hiện cũng có thể được coi là mẫu kinh ngạc. Mẫu kinh ngạc là hấp dẫn bởi vì chúng xác định sự thất bại trong tri thức trước đây và có thể đề xuất một khía cạnh mới cần được nghiên cứu về dữ liệu.

Sự khác biệt giữa tính kinh ngạc và tính mới lạ là ở chỗ một mẫu mới lạ là mẫu mới và không mâu thuẫn với bất kỳ mẫu nào

đã được người sử dụng biết, trong khi một mẫu kinh ngạc lại mâu thuẫn với tri thức trước đó hoặc mong đợi của người dùng.

Tính tiện ích: Mẫu là tiện ích nếu góp phần đạt được mục tiêu cho một người sử dụng nó. Những người sử dụng khác nhau có thể có những mục tiêu khác nhau liên quan đến những tri thức có thể được chiết xuất từ một tập dữ liệu. Ví dụ, một người có thể quan tâm tìm kiếm tất cả các doanh số bán hàng với lợi nhuận cao trong một tập dữ liệu giao dịch, trong khi người khác lại có thể quan tâm tìm kiếm tất cả các giao dịch với sự gia tăng lớn trong tổng doanh thu. Tính hấp dẫn dựa trên chức năng người dùng định nghĩa về tính tiện ích.

Tính hành động/áp dụng được. Mẫu có tính hành động (hoặc áp dụng được) trong một phạm vi nào đó nếu mẫu cho phép ra quyết định về những hành động trong tương lai thuộc miền ứng dụng. Tính hành động đôi khi được kết hợp với một mẫu lựa chọn chiến lược. Chưa có một phương pháp chung cho việc đo lường tính hành động và các độ đo hiện tại phụ thuộc vào các ứng dụng. Ví dụ, đo lường tính hành động như chi phí thay đổi tình trạng hiện tại của khách hàng để phù hợp với mục tiêu, hoặc đo lường tính hành động như là lợi nhuận mà một luật kết hợp có thể mang lại.

Các tiêu chí đo lường mẫu hấp dẫn nói trên có sự tương quan với nhau mà không phải độc lập hoàn toàn. Thứ nhất, chúng có sự tương đồng tương đối với nhau, chẳng hạn, tính thi hành được có thể là một xấp xỉ tốt cho tính kinh ngạc, và đối ngược lại; tính súc tích thường trùng hợp với tính phổ dụng; tính phổ dụng thường là độ nhạy giảm nhiễu cho nên cũng liên quan tới tính tin cậy. Thứ hai, chúng cũng có tính không tương đồng, chẳng hạn, tính phổ dụng có vẻ xung đột với tính đặc thù, trong khi đó nó (tính phổ dụng) lại tương đồng với tính mới lạ.

Các độ đo hấp dẫn được chia thành ba lớp chính là lớp các độ đo khách quan, độ đo chủ quan và độ đo dựa trên ngữ nghĩa dựa theo các tiêu chí mà độ đo đáp ứng.

Một độ đo được gọi là khách quan nếu đo lường nó chỉ dựa trên các dữ liệu thô, không có yêu cầu trực tiếp về tri thức của

người sử dụng hoặc gián tiếp thông qua một ứng dụng khác. Hầu hết các độ đo khách quan dựa trên lý thuyết xác suất, thống kê, hoặc lý thuyết thông tin. Các tiêu chí súc tích, phổ dụng, tin cậy, đặc thù, và đa dạng chỉ phụ thuộc vào các dữ liệu và các mẫu, do đó có thể được coi là khách quan.

Bảng 2.3. Độ đo hấp dẫn và công thức tính toán (trích, 2006 [HG06])

Measure	Formula
Support	$P(AB)$
Confidence/Precision	$P(B A)$
Coverage	$P(A)$
Prevalence	$P(B)$
Recall	$P(A B)$
Specificity	$P(\neg B \neg A)$
Accuracy	$P(AB) + P(\neg A \neg B)$
Lift/Interest	$P(B A)/P(B)$ or $P(AB)/P(A)P(B)$
Leverage	$P(B A) - P(A)P(B)$
Added Value/Change of Support	$P(B A) - P(B)$
Relative Risk	$P(B A)/P(B \neg A)$
Jaccard	$P(AB)/(P(A) + P(B) - P(AB))$
Certainty Factor	$(P(B A) - P(B))/(1 - P(B))$
Odds Ratio	$\frac{P(AB)P(\neg A \neg B)}{P(A \neg B)P(\neg B \neg A)}$
Yule's Q	$\frac{P(AB)P(\neg A \neg B) - P(A \neg B)P(\neg AB)}{P(AB)P(\neg A \neg B) + P(A \neg B)P(\neg AB)}$
Yule's Y	$\frac{\sqrt{P(AB)P(\neg A \neg B)} - \sqrt{P(A \neg B)P(\neg AB)}}{\sqrt{P(AB)P(\neg A \neg B)} + \sqrt{P(A \neg B)P(\neg AB)}}$
Klosgen	$\sqrt{P(AB)}(P(B A) - P(B)) - \sqrt{P(AB)}\max(P(B A) - P(B), P(A B) - P(A))$
Conviction	$\frac{P(A)P(\neg B)}{P(A \neg B)}$
Interestingness Weighting Dependency	$(\frac{P(AB)^k}{P(A)P(B)})^m - 1 + P(AB)^m$, where k, m are coefficients of dependency and generality, respectively, weighting the relative importance of the two factors.
Collective Strength	$\frac{P(AB) + P(\neg B \neg A)}{P(A)P(B) + P(\neg A)P(\neg B)} * \frac{1 - P(A)P(B) - P(\neg A)P(\neg B)}{1 - P(AB) - P(\neg B \neg A)}$
Laplace Correction	$\frac{N(AB)+1}{N(A)+2}$
Gini Index	$P(A) * (P(B A)^2 + P(\neg B A)^2) + P(\neg A) * (P(B \neg A)^2 + P(\neg B \neg A)^2) - P(B)^2 - P(\neg B)^2$
Goodman and Kruskal	$\sum_i \max_j P(A_i B_j) + \sum_j \max_i P(A_i B_j) - \max_i P(A_i) - \max_j P(B_j)$
Normalized Mutual Information	$\sum_i \sum_j P(A_i B_j) * \log_2 \frac{P(A_i B_j)}{P(A_i)P(B_j)} / (-\sum_i P(A_i) * \log_2 P(A_i))$
J-Measure	$P(AB)(\log_2 \frac{P(B A)}{P(\neg B A)}) + P(A \neg B)(\log_2 \frac{P(\neg B A)}{P(\neg B \neg A)})$
One-Way Support	$P(B A) * \log_2 \frac{P(AB)}{P(A)P(B)}$
Two-Way Support	$P(AB) * \log_2 \frac{P(AB)}{P(A)P(B)}$

Một độ đo được gọi là chủ quan nếu đo lường nó dựa trên cả dữ liệu và tri thức người sử dụng. Để có được tri thức người sử dụng, truy cập vào tên miền hoặc tri thức nền về dữ liệu của người dùng được yêu cầu. Truy cập này có thể thu được bằng cách tương tác với người sử dụng trong quá trình khai thác dữ liệu hoặc bằng cách tường minh đại diện cho tri thức hoặc kỳ vọng của người sử dụng. Tính mới lạ và tính kinh ngạc phụ thuộc vào người sử dụng các mẫu, cũng như các dữ liệu và các mẫu có sẵn, và do đó có thể được xem xét là có tính chủ quan.

Một độ đo được gọi là ngữ nghĩa nếu cần phải xem xét ngữ nghĩa và giải thích của mẫu. Bởi vì các độ đo ngữ nghĩa liên quan đến tri thức miên từ người sử dụng, độ đo ngữ nghĩa được xem xét như một loại độ đo chủ quan đặc biệt. Tính tiện ích và tính hành động phụ thuộc vào ngữ nghĩa của dữ liệu, và do đó có thể được xem xét là độ đo ngữ nghĩa. Một chức năng tiện ích đại diện ngữ nghĩa mục tiêu của người dùng cần được bổ sung và làm tối ưu hóa kết quả khai phá mẫu. Ví dụ, một hệ thống khai phá luật kết hợp hướng tới người sử dụng là quản lý cửa hàng nên có chức năng thể hiện ngữ nghĩa đảm bảo luật kết hợp có liên quan đến mặt hàng có lợi nhuận cao hơn được ưu tiên hơn những luật có ý nghĩa thống kê cao hơn.

Tồn tại ba phương pháp được dùng để xác định một mẫu là mẫu tri thức hay không dựa trên các tiêu chí đã có. Thứ nhất, tiến hành phân loại mẫu là hấp dẫn hay không, chẳng hạn, sử dụng kiểm thử thống kê χ^2 để phân biệt giữa các mẫu hấp dẫn và không hấp dẫn. Thứ hai, xác định một mối quan hệ ưu tiên giữa các mẫu để mô tả rằng một mẫu là hấp dẫn hơn các mẫu khác. Thứ ba, xếp hạng các mẫu khai phá được. Đối với hai phương pháp thứ nhất hoặc thứ ba, có thể xác định và sử dụng một độ đo hấp dẫn dựa trên chín tiêu chí nói trên.

Các độ đo hấp dẫn rất phong phú và phần lớn các độ đo khách quan dựa trên cơ sở độ đo thống kê, chẳng hạn, Bảng 2.3 trình bày một tập các độ đo khách quan dựa trên thống kê. Mỗi một dòng trong Bảng 2.3 tương ứng với một độ đo, bao gồm tên độ đo và công thức tính toán.

Như đã giới thiệu, đo lường tính hấp dẫn của mẫu, cụ thể là nghiên cứu về độ đo hấp dẫn là nội dung nghiên cứu năng động và quan trọng. Một số tài liệu như giới thiệu sau đây có thể cung cấp nền tảng tốt cho hướng nghiên cứu này.

Liqiang Geng và Howard J. Hamilton [GH06] cung cấp một khung nhìn vừa khái quát vừa chuyên sâu về độ đo hấp dẫn trong phát hiện tri thức. Một số nội dung chính trong nghiên cứu của hai tác giả đã được giới thiệu ở trên. Yao Y.Y. và cộng sự cũng có nhiều nghiên cứu về độ đo hấp dẫn, trong đó những phân tích của Yao Y.Y. [Yao03] cho cách tiếp cận tốt khi nghiên cứu về độ đo hấp dẫn. Xuan-Hiep Huynh và cộng sự [HGEK07] trình bày 36 độ đo hấp dẫn được khảo sát để đánh giá dựa trên đồ thị (phát triển các kết quả nghiên cứu từ luận án TS của Xuan-Hiep Huynh. Trong [HZ10], M.J. Heravi và O. R. Zaïane phân tích về 53 độ đo hấp dẫn khách quan. Yuejin Zhang và cộng sự [ZZNS09] trình bày một số phân tích về 12 độ đo hấp dẫn (8 độ đo khách quan và 4 độ đo chủ quan) tương ứng với 9 tiêu chí mục tiêu của độ đo hấp dẫn.

Trong bài toán phân lớp dữ liệu (Chương 6), hai bộ độ đo điển hình nhất là (i) bộ độ đo gồm độ chính xác (precision measure), độ hồi tưởng (recall measure) và kết hợp của chúng (F_β mà điển hình nhất là độ đo F_1); (ii) Bộ độ đo gồm độ chính xác (accuracy measure) và mức độ lỗi (error rate). Đối với các bộ dữ liệu "không cân đối" (lực lượng phần tử của các lớp là quá lệch nhau), nhiều nhà nghiên cứu (chẳng hạn như [NEM09]) cho rằng bộ độ đo (độ chính xác, độ hồi tưởng) mà đại diện là độ đo F có hiệu lực cao hơn cặp độ đo (độ chính xác, hệ số lỗi).

CÂU HỎI VÀ BÀI TẬP

- 2.1.** Đầu tư CNTT cần hướng tới các tiêu chí gì qua nội dung của nghịch lý hiệu quả của CNTT, luận điểm của Carr và bàn luận liên quan của cộng đồng.
- 2.2.** Phân tích vai trò và kỹ năng chính của người giám đốc thông tin (CIO) trong tổ chức.

- 2.3. Khái niệm kinh tế tri thức, bốn cột trụ của kinh tế tri thức và vai trò của CNTT đối với bốn cột trụ này.
- 2.4. Bốn dạng siêu tri thức (meta-knowledge).
- 2.5. Ma trận chuyển hóa tri thức SECI (Socialization – Xã hội hóa, Externalization – Ngoại hiện, Combination - Kết hợp, Internalization - Tiếp thu).
- 2.6. Những nội dung chính trong khung nhìn tri thức doanh nghiệp.
- 2.7. Vòng đời của tri thức doanh nghiệp.
- 2.8. Khái niệm và các thành phần chính của công nghệ tri thức.
- 2.9. Trình bày các nội dung chính về xu thế phát triển các mô hình phát hiện tri thức từ dữ liệu. Liên hệ với vai trò chiến lược của CNTT.
- 2.10. Mô hình khai phá dữ liệu theo chuẩn công nghiệp CRISP-DM.
- 2.11. Mô hình quá trình khai phá dữ liệu hướng miền ứng dụng theo [CYZZ10].
- 2.12. Những điểm cần lưu ý khi đặt bài toán khai phá dữ liệu.
- 2.13. Khái niệm và các tính chất nên có của một độ đo hấp dẫn trong khai phá dữ liệu.

149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
1000

Chương 3.

CHUẨN BỊ DỮ LIỆU

3.1. GIỚI THIỆU

Trong quá trình khai phá dữ liệu, việc hiểu được đặc tính của dữ liệu giúp cho quá trình phân tích dữ liệu trở nên hiệu quả hơn rất nhiều. Khái niệm hiểu dữ liệu ở đây liên quan chặt chẽ tới khái niệm chất lượng của dữ liệu. Trong thực tế khi xây dựng kho dữ liệu chuẩn bị cho bước khai phá dữ liệu, khả năng dữ liệu có thể bị nhiễu, không đầy đủ, và không nhất quán. Đây là những vấn đề rất hay xảy ra trong những nơi có trữ lượng dữ liệu lớn:

- ✓ Trường hợp dữ liệu không đầy đủ có thể có nhiều lý do cả khách quan lẫn chủ quan. Chẳng hạn như rất nhiều thông tin cần quan tâm về khách mua hàng ta không thể dễ dàng lấy được (vì rất nhiều người coi đó là thông tin riêng tư). Hoặc một số thông tin tại thời điểm thu thập ta không nghĩ nó quan trọng nên không lấy. Hoặc dữ liệu không thể thu thập được do lỗi thiết bị.
- ✓ Dữ liệu bị nhiễu cũng do nhiều nguyên nhân, chẳng hạn như lỗi thiết bị thu nhận hoặc truyền dẫn; khi nhập dữ liệu người nhập có thể nhập sai;
- ✓ Dữ liệu không nhất quán có thể phát sinh từ việc không sử dụng chung một chuẩn quy ước khi nhập dữ liệu, hoặc định dạng dữ liệu là khác nhau (ví dụ như định dạng ngày tháng có rất nhiều loại).

Vì lý do này mà ta cần có bước *chuẩn bị dữ liệu* nhằm đảm bảo dữ liệu đầu vào cho các thuật toán khai phá là chuẩn và chính xác.

xác, vì chất lượng của dữ liệu có ảnh hưởng rất lớn đến kết quả khai phá. Trong chương này sẽ trình bày các phương pháp chuẩn bị dữ liệu thông dụng hay được dùng trong thực tế là làm sạch dữ liệu, tích hợp dữ liệu, chuyển đổi dữ liệu và làm giảm dữ liệu.

3.2. HIẾU DỮ LIỆU

Để khai phá dữ liệu thành công, trước khi thực hiện các phương pháp khai phá ta cần phải có cái nhìn tổng quát về dữ liệu, trên cơ sở đó ta có thể phát hiện ra các đặc tính của dữ liệu, cũng như phát hiện ra đâu là dữ liệu nhiễu hay dữ liệu ngoại lai. Quan trọng hơn ta có thể tìm ra được phương pháp tiền xử lý và khai phá dữ liệu nào là phù hợp với tập dữ liệu ta đang xét. Một trong những tính chất của dữ liệu ta cần quan tâm là xu hướng tập trung và phân tán của dữ liệu. Độ tập trung của dữ liệu có thể đo được bằng các độ đo: *trung bình (mean)*, *trung vị (median)*, *mode* và *midrange*. Độ phân tán của dữ liệu có thể đo được thông qua các độ đo *quartile*, *interquartile range* và *variance*. Những độ đo trên được gọi là những thông tin tóm tắt về dữ liệu. Ta có thể hiển thị dữ liệu tóm tắt trên để có được cái nhìn trực quan về đặc tính của dữ liệu.

3.2.1. Đo độ tập trung của dữ liệu

Độ đo trung bình: Đây là độ đo phổ dụng nhất, nó đại diện cho trọng tâm của dữ liệu. Gọi x_1, x_2, \dots, x_N là N phần tử dữ liệu cho một thuộc tính nào đó, chẳng hạn thuộc tính *giá (price)*, khi đó giá trị trung bình của tập dữ liệu trên là:

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N} = \frac{x_1 + x_2 + \dots + x_N}{N} \quad (3.1)$$

Độ đo này cũng đã được tích hợp vào nhiều hệ quản trị cơ sở dữ liệu, nó chính là hàm *avg()* trong ngôn ngữ SQL. Trong một số trường hợp mỗi phần tử dữ liệu có trọng số w_i khác nhau, ta có công thức tương ứng cho giá trị trung bình như sau:

$$\bar{x} = \frac{\sum_{i=1}^N w_i x_i}{\sum_{i=1}^N w_i} = \frac{w_1 x_1 + x_2 w_2 + \dots + w_N x_N}{w_1 + w_2 + \dots + w_N} \quad (3.2)$$

Khi dữ liệu có nhiều giá trị bất thường, chẳng hạn như có một vài phần tử có giá trị cao vượt lên thì giá trị trung bình sẽ bị ảnh hưởng. Để khắc phục điều này, một trong những giải pháp đơn giản là sử dụng độ đo trung bình có cắt xén (trimmed mean). Cụ thể ta sắp xếp dữ liệu theo chiều tăng, sau đó loại bỏ đi một số giá trị cao nhất và thấp nhất (ví dụ loại bỏ 2%). Giá trị còn lại được dùng để tính giá trị trung bình.

Độ đo trung vị: Khi dữ liệu có phân bố lệch thì độ đo trung bình cũng không phù hợp, ta có thể sử dụng độ đo trung vị. Giả sử ta có N giá trị khác nhau được sắp xếp theo thứ tự tăng dần, khi đó trung vị của tập dữ liệu này là phần tử ở giữa (nếu N lẻ), và bằng trung bình của 2 phần tử ở giữa (nếu N chẵn). Trong trường hợp tổng quát thì cách tính trên không còn đúng nữa, ta có thể tính xấp xỉ trung vị như sau. Ta nhóm dữ liệu vào các nhóm tương ứng với các khoảng dữ liệu. Ví dụ ta có thể nhóm trường giá (price) ở trên vào các khoảng 10000-20000, 20000-30000,... Gọi $freq_{median}$ là số lượng (tần suất) các phần tử dữ liệu nằm trong nhóm chứa trung vị tính theo công thức ở trên; L_1 là cận dưới của các giá trị dữ liệu; $width$ là độ lớn của nhóm chứa trung vị; $(\sum freq)l$ là tổng số các phần tử dữ liệu của các nhóm có giá trị nhỏ hơn nhóm chứa trung vị; N là tổng số lượng các phần tử dữ liệu, khi đó công thức tính trung vị cho cả tập dữ liệu là:

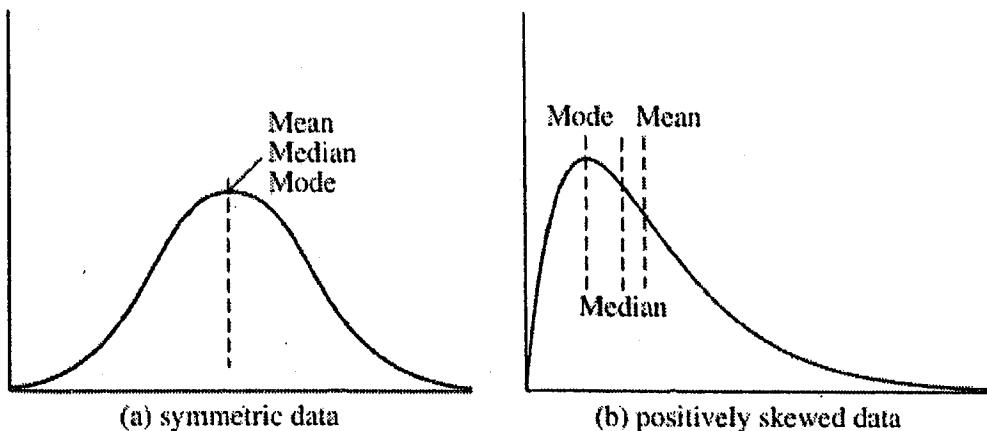
$$median = L_1 + \left(\frac{N/2 - (\sum freq)l}{freq_{median}} \right) width \quad (3.3)$$

Mode: Là một độ đo nữa đo độ tập trung của dữ liệu, nó là tập con dữ liệu xuất hiện với tần suất cao nhất trong tập dữ liệu. Trong trường hợp tổng quát, có thể tồn tại nhiều tập con dữ liệu cùng xuất hiện với tần suất cao nhất, khi đó ta nói dữ liệu là *multimodal*. Trường hợp dữ liệu có 1, 2 hay 3 thì các tên tương

ứng với nó là unimodal, bimodal và trimodal. Nếu tập dữ liệu có các phân tử dữ liệu có giá trị hoàn toàn khác nhau (tần suất xuất hiện của các phân tử dữ liệu là 1) thì không tồn tại mode. Trong trường hợp dữ liệu có 1 mode, thì ta có công thức tính đơn giản như sau:

$$\text{mean-mode} = 3(\text{mean}-\text{median})$$

Nếu dữ liệu có phân bố đối xứng thì các giá trị mean, median và mode là trùng nhau, trường hợp dữ liệu có phân bố không đối xứng thì chúng có các giá trị khác nhau như minh họa trên Hình 3.1.



Hình 3.1. Vị trí của các giá trị mean, median và mode

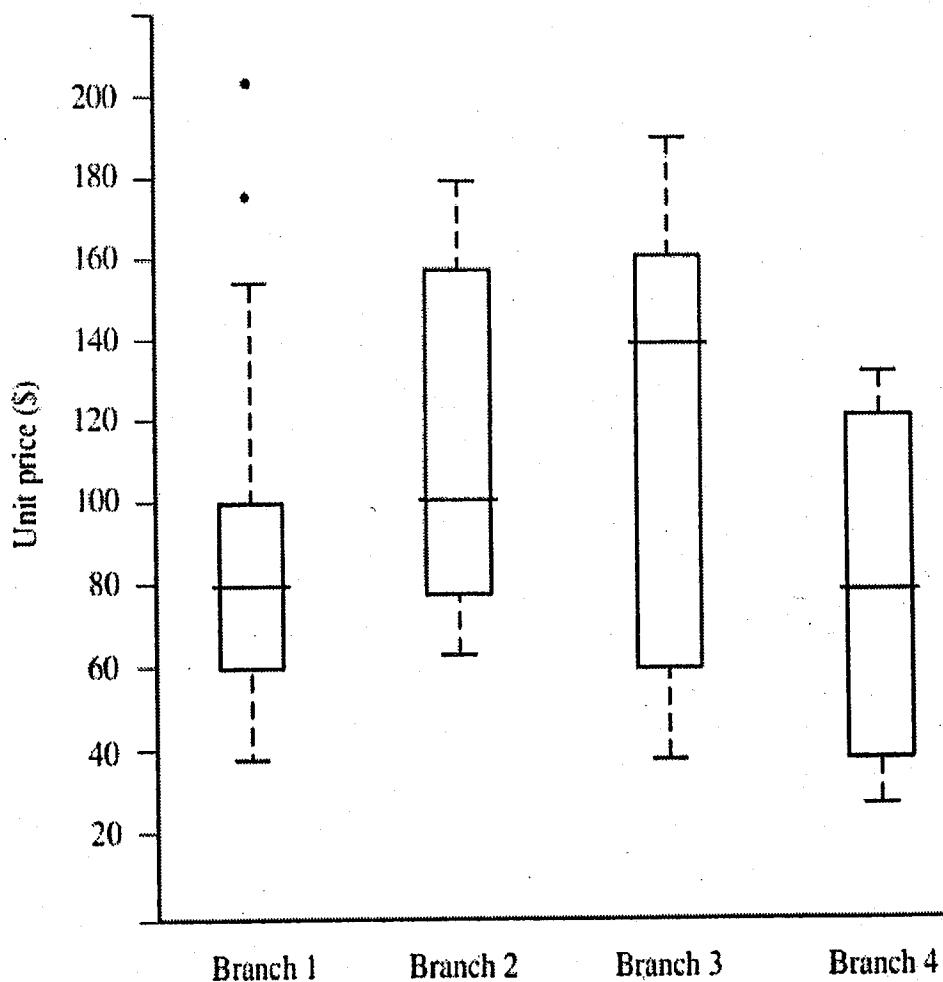
Midrange: Độ đo này cũng được dùng để đánh giá độ tập trung của dữ liệu, nó là giá trị trung bình của giá trị lớn nhất (hàm `max()` trong SQL) và thấp nhất (hàm `min()` trong SQL) trong tập dữ liệu.

3.2.2. Đo độ phân tán của dữ liệu

Gọi x_1, x_2, \dots, x_N là N là tập quan sát cho một thuộc tính nào đó được sắp xếp theo thứ tự tăng dần, chẳng hạn thuộc tính *giá* (*price*). Miền giá trị (range) của tập dữ liệu này là [Min, Max], trong đó Min là giá trị nhỏ nhất và Max là giá trị lớn nhất trong tập dữ liệu này. Phân tử thứ $k\%$ là phân tử x_i sao cho x_i có giá trị lớn hơn hoặc bằng các phân tử nằm trong phần $k\%$ tính từ đầu

dãy. Như vậy trung vị (median) ở phần trên là phần tử 50%. Phần tử hay được dùng hơn trung vị trong phần này là phần tứ (quartile), phần tử thứ nhất ký hiệu là Q_1 là phần tử 25%, phần tử thứ 2 (Q_2) là phần tử 50%, phần tử thứ 3 (Q_3) là phần tử 75%. Các giá trị này thể hiện trung tâm, độ bao phủ và hình dạng của phân bố dữ liệu. Khoảng cách từ phần tử thứ nhất đến phần tử thứ 3 là độ đo đơn giản thể hiện sự bao phủ của dữ liệu, hay nó chính là miền giá trị của phần nửa giữa của dữ liệu. Khoảng cách này được gọi là interquartile range (IQR):

$$IQR = Q_3 - Q_1$$



Hình 3.2. Boxplot cho dữ liệu giá bán cho các chi nhánh

Giá trị này cũng rất hữu ích để phân tích dữ liệu có phân bố lệch. Ngoài ra nó còn có thể dùng để phát hiện ra các phần tử ngoại lai, phần tử ngoại lai là phần tử có giá trị nhỏ hơn $1,5 \times IQR$ giá trị phần tử thứ nhất, hoặc lớn hơn $1,5 \times IQR$ giá trị phần tử

thứ 3. Vì phần tử thứ 3 chưa chứa thông tin về dữ liệu nằm ở cuối dãy nên trong thực tế, để mô tả dữ liệu, người ta tạo ra *bộ 5 tóm tắt dữ liệu* (five-number summary) gồm: Min, Q_1 , Median, Q_3 , Max. Bộ 5 tóm tắt này được biểu diễn bằng một boxplot như hình 3.2 mô tả phân bố của dữ liệu giá bán một mặt hàng tại các chi nhánh khác nhau. Trong đó phần dưới cùng là Min, phần tiếp theo (đáy của hình chữ nhật) là Q_1 , đoạn thẳng nằm trong hình chữ nhật là Median, cạnh trên của hình chữ nhật là Q_3 , và cao nhất là Max.

Nếu ta nhận thấy không có dữ liệu bất thường thì ta giữ nguyên giá trị của Max và Min, ngược lại ta thay giá trị của Max bằng $1,5 \times \text{IQR} + Q_3$ và $\text{Min} = Q_1 - 1,5 \times \text{IQR}$. Các điểm dữ liệu xuất hiện ngoài khoảng này được coi là dữ liệu ngoại lai. Ví dụ như ở chi nhánh 1 trên hình 3.2 ta có 2 phần tử ngoại lai ở phía trên giá trị Max.

Phương sai và độ lệch chuẩn: phương sai (variance) của một tập dữ liệu gồm N phần tử x_1, x_2, \dots, x_N là:

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2 = \frac{1}{N} \left[\sum_{i=1}^N x_i^2 - \frac{1}{N} \left(\sum_{i=1}^N x_i \right)^2 \right] \quad (3.4)$$

trong đó \bar{x} là giá trị trung bình. Độ lệch chuẩn (standard deviation) σ chính là căn bậc 2 của phương sai. Độ lệch chuẩn hay được dùng cùng giá trị trung bình khi độ trung bình được lựa chọn là trung tâm, nó thể hiện sự bao phủ (độ lệch) của dữ liệu quanh giá trị trung bình. Nếu dữ liệu là giống nhau thì $\sigma = 0$, ngược lại $\sigma > 0$. Giá trị của σ càng lớn thì giá trị của dữ liệu càng khác nhau nhiều.

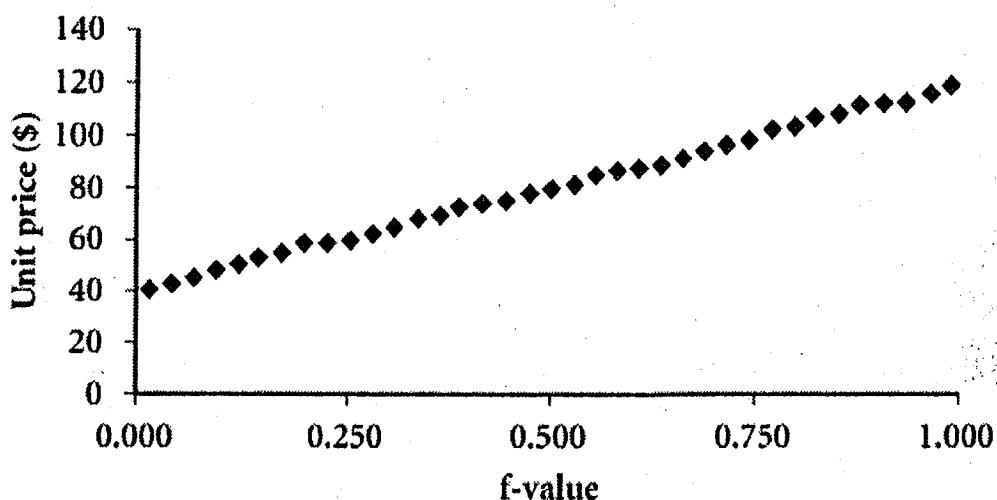
3.2.3. Hiển thị dữ liệu tóm tắt

Ngoài các biểu đồ, đồ thị dùng để hiển thị dữ liệu, ta còn có các cách hiển thị các thông tin tóm tắt về dữ liệu bao gồm: *biểu đồ tần suất* (histogram), *q-q plot*, *scatter plot* và *loes curve*. Boxplot cũng là một cách hiển thị dữ liệu tóm tắt. Ví dụ về biểu đồ tần suất có thể xem ở mục 3.6.2.

Đồ thị quantile plot: là một phương pháp hiển thị dữ liệu đơn giản trên dữ liệu một chiều (univariate). Qua hình ảnh hiển thị ta có thể có một cái nhìn tổng thể về dữ liệu cũng như những

giá trị bất thường trong dữ liệu. Gọi x_i là tập giá trị dữ liệu ($1 \leq i \leq N$) được sắp xếp theo chiều tăng dần, mỗi giá trị x_i được gán với giá trị phần trăm f_i là giá trị xấp xỉ với i/N (tỉ lệ % số lượng dữ liệu nhỏ hơn hoặc bằng x_i) được đề cập ở trên. Gọi là f_i xấp xỉ vì có thể không tồn tại dữ liệu thực thỏa mãn điều kiện trên và giá trị của f_i được tính bằng công thức sau: $f_i = (0,5-i)/N$. Như vậy giá trị của $f_1 = 0,25$ sẽ tương đương với Q_1 , $f_2 = 0,5$ sẽ tương đương với Q_2 , $f_3 = 0,75$ sẽ tương đương với Q_3 .

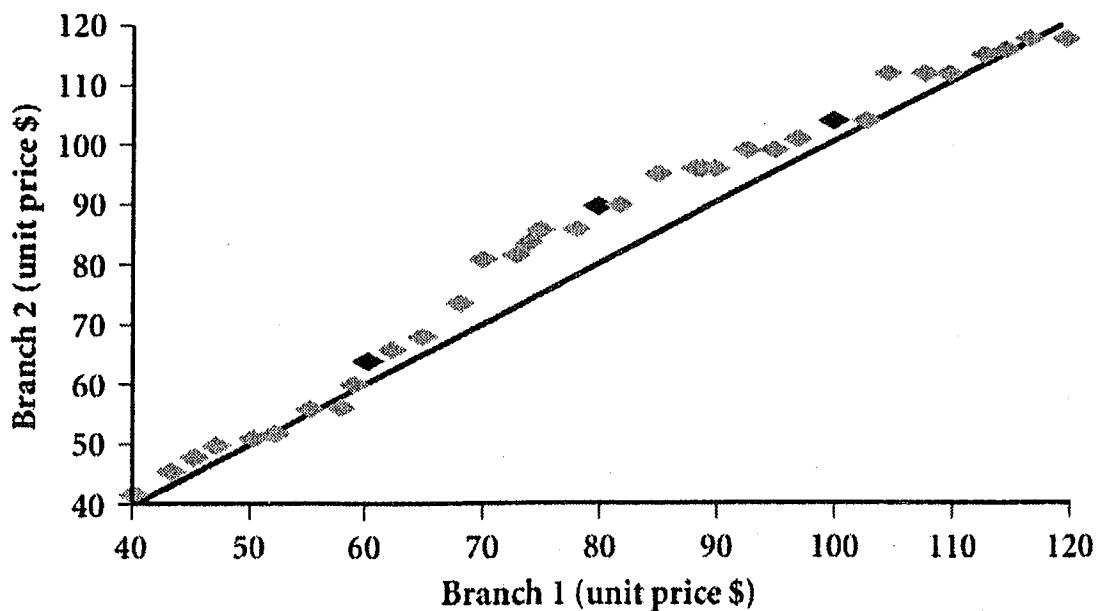
Khi biểu diễn trên đồ thị, giá trị x_i sẽ được vẽ tương ứng với f_i . Giả sử có 2 tập dữ liệu về giá bán của một chi nhánh tại 2 thời điểm khác nhau, đồ thị quantile plot sẽ cho chúng ta có thể so sánh được phân bố dữ liệu tại 2 thời điểm khác nhau. Hình 3.3 là một đồ thị quantile plot về giá tiền của mặt hàng.



Hình 3.3. Đồ thị quantile plot cho thuộc tính giá (price)

Đồ thị quantile-quantile plot (q-q plot): để so sánh phân bố dữ liệu của 2 chi nhánh khác nhau ta có thể sử dụng đồ thị này. Gọi x_1, x_2, \dots, x_N là N phần tử dữ liệu đã được sắp xếp của chi nhánh thứ nhất; y_1, y_2, \dots, y_M là M phần tử dữ liệu đã được sắp xếp của chi nhánh thứ 2. Nếu $N = M$ thì ta chỉ cần vẽ x_i tương ứng với y_i . Nếu $M < N$ khi đó ta chỉ vẽ M điểm $(i-0.5)/M$ của dữ liệu x tương ứng với y .

Hình 3.4 minh họa đồ thị q-q plot của dữ liệu cho thuộc tính price ở 2 chi nhánh khác nhau. Để dễ so sánh ta vẽ thêm đường thẳng đi qua các điểm có giá trị bằng nhau trên 2 trục số. Điểm thấp nhất trong đồ thị là tương ứng với 0,03 quantile, các ô được tô đậm tương ứng với Q_1 , trung vị và Q_3 . Qua đồ thị này ta có thể thấy ngay được giá bán tại chi nhánh 1 thấp hơn một chút so với chi nhánh 2, nhưng tại một số điểm, chẳng hạn như điểm cao nhất thì chi nhánh 1 lại cao hơn chi nhánh 2.

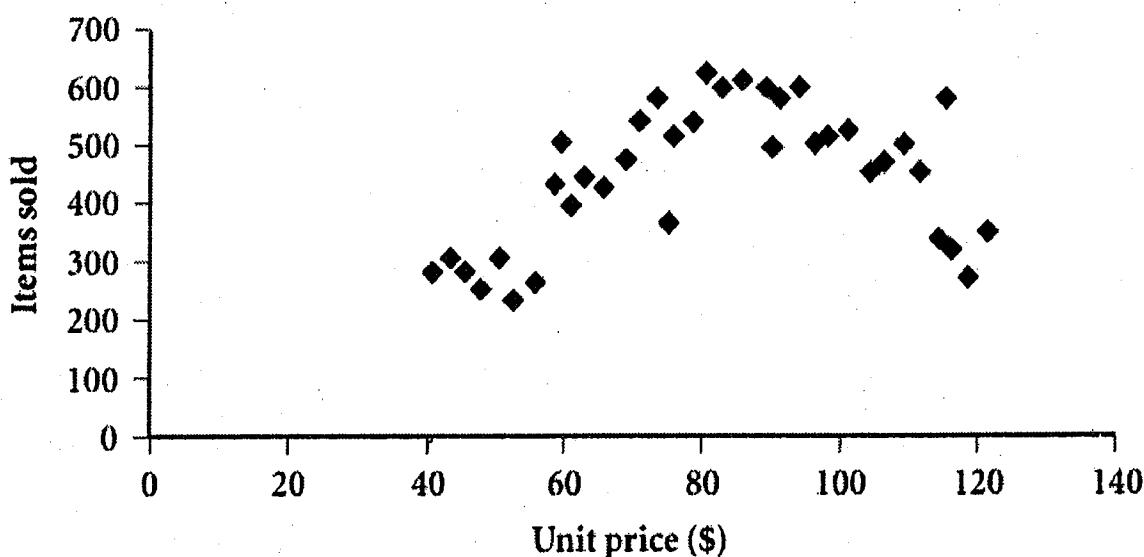


Hình 3.4. Đồ thị q-q plot so sánh 2 chi nhánh với nhau

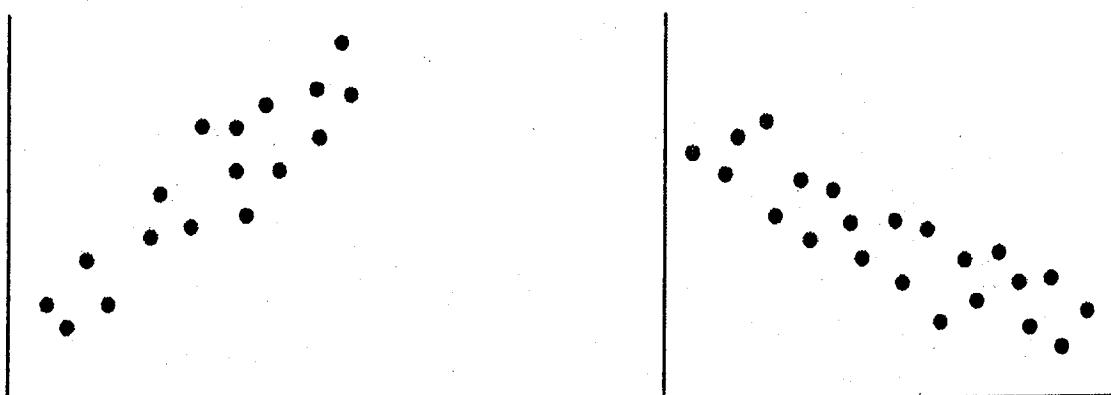
Đồ thị scatter plot: là một trong những công cụ đồ họa mạnh nhất, nó cho phép ta có thể kiểm tra xem liệu có mối quan hệ, mẫu hay xu hướng giữa 2 thuộc tính số. Đồ thị này đơn giản chỉ vẽ lên mặt phẳng các điểm tương ứng với giá trị của cặp thuộc tính trên (do đó có thể nó chỉ thích hợp khi số lượng dữ liệu là nhỏ). Hình 3.5 là đồ thị scatter plot của tập dữ liệu về giá. Đồ thị này có thể cho ta cái nhìn trực quan về dữ liệu, từ đó có thể phát hiện ra đặc tính của dữ liệu, sự tương quan giữa các thuộc tính và có thể phát hiện ra luôn cả các giá trị ngoại lai. Trong hình này, ta không thấy sự tương quan nào giữa 2 thuộc tính. Hình 3.6 là một

đồ thị scatter plot khác cho thấy tồn tại sự tương quan giữa 2 thuộc tính. Hình bên trái là tương quan dương, hình bên phải là tương quan âm.

Loes curve: là đồ thị xấp xỉ phân bố dữ liệu, nó là một công cụ quan trọng cung cấp cho người phân tích về mối quan hệ giữa 2 thuộc tính. Từ *loes* là viết tắt của từ hồi quy cục bộ (local regression). Hình 3.7 minh họa đồ thị loes curve cho tập dữ liệu được vẽ ở hình 3.5.



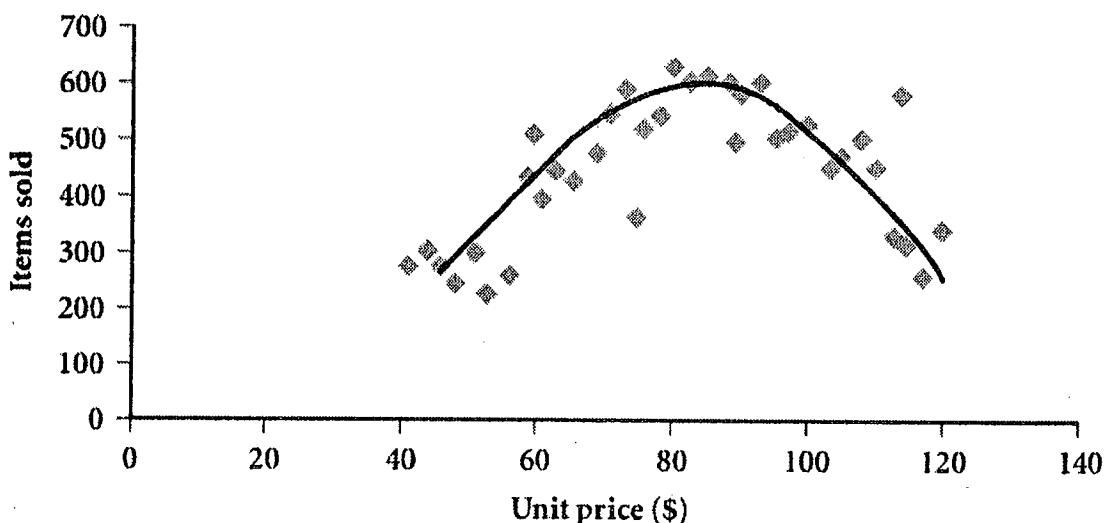
Hình 3.5. Đồ thị scatter plot cho thuộc tính giá



Hình 3.6. Đồ thị scatter có tồn tại sự tương quan giữa 2 thuộc tính

3.3. TIỀN XỬ LÝ DỮ LIỆU

Có nhiều cách tiền xử lý dữ liệu khác nhau nhằm mục tiêu tăng chất lượng dữ liệu và từ đó có thể làm tăng hiệu quả của các kỹ thuật khai phá dữ liệu. Mỗi một kỹ thuật cụ thể sẽ giúp cải thiện chất lượng dữ liệu theo hướng nhất định và hiệu quả của nó phụ thuộc rất nhiều vào đặc trưng của dữ liệu. Một số kỹ thuật tiền xử lý dữ liệu thường được áp dụng bao gồm:



Hình 3.7. Đồ thị loes curve biểu diễn quan hệ giữa 2 thuộc tính

- ✓ *Kỹ thuật làm sạch dữ liệu* (data cleaning) thường được sử dụng để thêm những giá trị bị thiếu, loại bỏ nhiễu, xác định và loại bỏ các giá trị ngoại lai và giải quyết vấn đề không nhất quán của dữ liệu. Hiển nhiên ta sẽ không thể tin tưởng vào kết quả thu được từ bất kỳ thuật toán khai phá dữ liệu nào nếu ta biết chắc rằng dữ liệu còn chưa được làm sạch và có chất lượng tốt. Một số kỹ thuật khai phá dữ liệu đã tích hợp sẵn các môđun để loại bỏ nhiễu và xử lý dữ liệu thiếu, tuy nhiên phần lớn chúng hoạt động không thực sự hiệu quả. Vì vậy, thay vì tập trung vào việc làm sạch dữ liệu các thuật toán khai phá dữ liệu có thể tập trung vào việc xây dựng các mô hình hiệu quả hơn. Nhiệm vụ làm sạch dữ liệu sẽ được thực hiện trong quá trình tiền xử lý dữ liệu trước khi sử dụng bất kỳ thuật toán khai phá dữ liệu nào (xem phần 3.4).

- ✓ *Kỹ thuật tích hợp dữ liệu* (data integration): cho phép trộn (lắp ghép/ tích hợp) dữ liệu từ nhiều nguồn khác nhau về một kho chứa đồng nhất và có tính gắn kết chặt chẽ phục vụ cho quá trình khai phá dữ liệu tiếp theo. Như chúng ta đã biết, các nguồn dữ liệu khác nhau thì tổ chức và định nghĩa dữ liệu hoàn toàn có thể khác nhau. Ví dụ: để chỉ cùng một thuộc tính tên người có nguồn định nghĩa là Name, nguồn khác đặt là TEN, hoặc chia ra là TEN, HO và DEM. Ngay cả trong miền giá trị của từng thuộc tính cũng có thể được định nghĩa khác nhau, ví dụ như thuộc tính TUOI = {(0...3), (4-18), (19-39), (40,59), (60, ...)} tương đương với {"sơ sinh", "trẻ em", "thanh niên", "trung niên", "người già"}. Quá trình khai phá tri thức sẽ không thể thực hiện, thực hiện chậm hoặc thực hiện không chính xác khi dữ liệu có càng nhiều dữ liệu dư thừa. Hiển nhiên ta thấy trong khi tích hợp dữ liệu các kỹ thuật làm sạch dữ liệu phải được áp dụng nhằm tránh sự dư thừa dữ liệu. Không những thế các kỹ thuật làm sạch còn được áp dụng để phát hiện và loại bỏ các dữ liệu dư thừa sau khi tích hợp dữ liệu từ nhiều nguồn khác nhau.
- ✓ *Thu gọn (làm giảm) dữ liệu* (data reduction) nhằm giảm kích cỡ của dữ liệu nhiều nhất có thể mà không làm ảnh hưởng (hoặc ảnh hưởng ở mức chấp nhận được) tới kết quả phân tích. Việc thu gọn dữ liệu thường xảy ra trong trường hợp dữ liệu quá lớn tới mức làm giảm hiệu năng của các kỹ thuật khai phá dữ liệu như thời gian chạy quá lâu hoặc không đủ bộ nhớ để thực hiện... Có hai chiến lược thu gọn dữ liệu là *giảm chiều dữ liệu* (dimensionality reduction) và *giảm số lượng dữ liệu* (numerosity reduction).
- ✓ *Kỹ thuật chuyển dạng dữ liệu* (data transformation) có thể ứng dụng với dữ liệu có phân bổ không phù hợp với các thuật toán phân tích dữ liệu dựa trên khoảng cách như mạng nơron, phân lớp K-lắng riêng gần nhất,... Với những

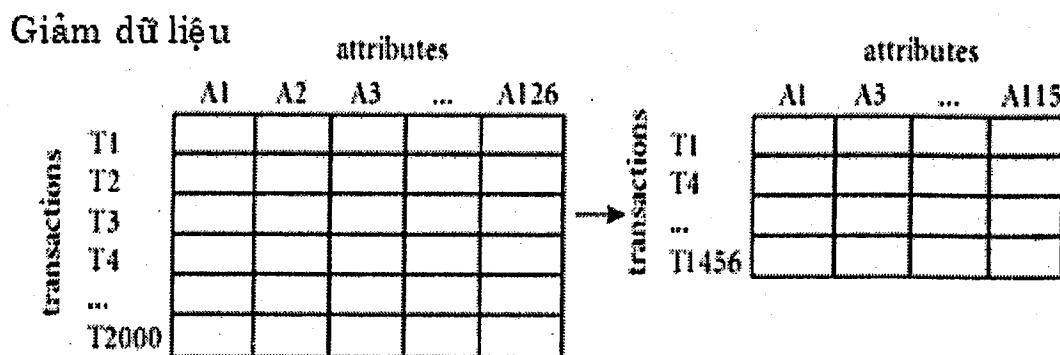
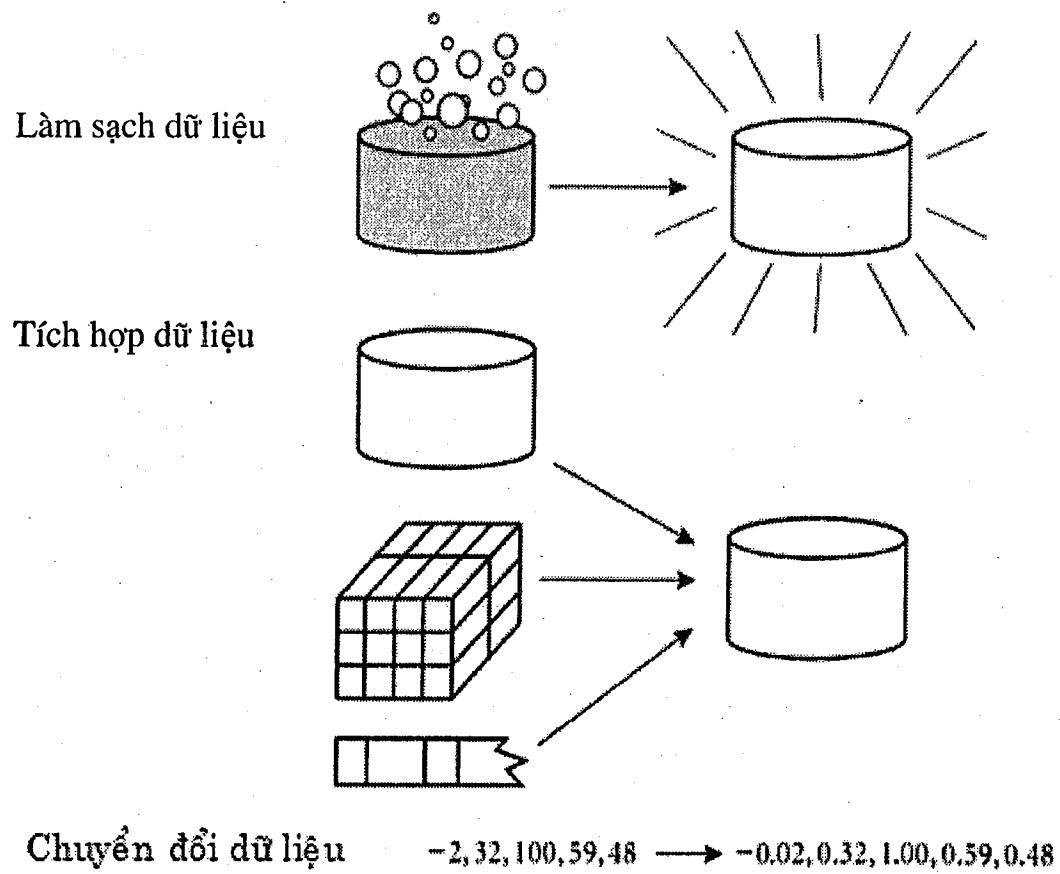
kỹ thuật khai phá dữ liệu này, thông thường dữ liệu cần được chuẩn hóa về cùng một miền dữ liệu thì các độ đo khoảng cách mới được áp dụng một cách hiệu quả. Phương pháp rời rạc hóa (discretization) và tạo cây phân cấp khái niệm (concept hierarchy generation) dữ liệu cũng là những kỹ thuật rất hiệu quả trong việc chuyển dạng dữ liệu. Ví dụ thay vì biểu diễn tuổi bằng các con số, ta có thể biểu diễn bằng tập hợp các từ “trẻ em”, “thanh niên”, “trung niên” và “người già”.

Những kỹ thuật và nhóm kỹ thuật trên đây có thể được áp dụng đồng thời với nhau để tăng hiệu quả sử dụng và chúng hoàn toàn không loại trừ lẫn nhau. Hình vẽ 3.8 tổng kết các kỹ thuật tiền xử lý dữ liệu được trình bày ở trên.

Nói chung, dữ liệu thực tế thường chứa nhiều nhiễu, không đầy đủ và không nhất quán. Tiền xử lý dữ liệu giúp tăng chất lượng của dữ liệu, từ đó có thể cải tiến được độ chính xác và hiệu quả của các quá trình khai phá dữ liệu ở các bước tiếp theo. Tiền xử lý dữ liệu là một trong những bước rất quan trọng trong quá trình khai phá tri thức bởi tính đúng đắn của các quyết định phụ thuộc rất nhiều vào chất lượng của dữ liệu. Phát hiện ra sự bất thường và sửa chữa sớm dữ liệu cũng như giảm dữ liệu phân tích có thể thu được lợi ích vô cùng lớn trong quá trình xử lý tri thức để đưa ra quyết định.

3.4. LÀM SẠCH DỮ LIỆU

Như trình bày ở mục 3.1, dữ liệu thường không đầy đủ, chứa nhiều giá trị nhiễu và không ổn định. Kỹ thuật này tìm cách tính toán các giá trị còn thiếu, loại bỏ và làm mịn các giá trị nhiễu trong quá trình xác định đặc trưng, cũng như chỉnh sửa sự nhất quán của dữ liệu. Ở phần này, chúng tôi chỉ trình bày một số phương pháp cơ bản để làm sạch dữ liệu bao gồm cách khôi phục dữ liệu bị thiếu, các kỹ thuật làm mịn, quy trình làm sạch dữ liệu.



Hình 3.8. Các kỹ thuật tiền xử lý dữ liệu

3.4.1. Các giá trị bị thiếu

Trong trường hợp dữ liệu có rất nhiều bản ghi có các thuộc tính không có dữ liệu. Liệu có cách nào để lấp đầy những vị trí thiếu dữ liệu như vậy không? Có một số phương pháp như sau:

1. Bỏ qua những bản ghi thiếu dữ liệu: kỹ thuật này thường được áp dụng khi thuộc tính nhãn bị thiếu (trong trường hợp phân lớp). Cách thức này thường không hiệu quả, trừ

trường hợp bản ghi có nhiều thuộc tính thiếu giá trị. Kỹ thuật này đặc biệt tồi trong trường hợp số lượng bản ghi có các thuộc tính không có giá trị chiếm một số lượng đáng kể so với các bản ghi đầy đủ. Trong một số trường hợp thì dữ liệu trong các bản ghi không đầy đủ lại có thể có một ý nghĩa nào đó trong quá trình phân tích dữ liệu.

2. Xác định các giá trị còn thiếu một cách thủ công: nói chung đây là một kỹ thuật tốn kém về mặt thời gian và nó thực sự không khả thi trong trường hợp dữ liệu lớn với nhiều giá trị bị thiếu.
3. Sử dụng hằng số toàn cục: thay thế toàn bộ các giá trị còn thiếu bằng một hằng số được định nghĩa trước. Phương pháp này thực hiện khá đơn giản tuy nhiên hiệu quả của nó không được chứng minh một cách rõ ràng.
4. Sử dụng các độ đo hướng trọng tâm của dữ liệu (ví dụ như tính trung bình cộng hoặc tính trung vị,...). Với các dữ liệu đối xứng thông thường áp dụng kỹ thuật tính trung bình, còn với dữ liệu không đối xứng thì tính trung vị phù hợp hơn.
5. Sử dụng giá trị bình quân hay trung vị của một thuộc tính cho tất cả các giá trị của cùng một lớp.
6. Sử dụng giá trị có khả năng cao nhất để thay thế cho giá trị thiếu: điều này có thể xác định được thông qua kỹ thuật hồi quy hoặc, sử dụng các công cụ suy diễn dựa trên lý thuyết Bayes hay quy nạp dựa trên cây quyết định.

Các phương pháp từ 3 đến 6 có thể bị ảnh hưởng bởi dữ liệu, do đó giá trị được thay thế có thể không chính xác. Tuy vậy, kỹ thuật số 6 lại được sử dụng khá phổ biến. Chúng ta cần lưu ý, trong nhiều trường hợp, các giá trị bị thiếu không có nghĩa là dữ liệu bị lỗi. Ví dụ khi chúng ta đi khám bệnh, người bệnh không có thẻ bảo hiểm y tế sẽ được để trống ở mục thẻ BHYT. Nói chung, mặc dù chúng ta có thể sử dụng kỹ thuật để làm sạch dữ liệu sau khi nhận được, tuy nhiên các kỹ thuật thu thập dữ liệu cũng cần phải được cải tiến để giảm số lượng các giá trị bị thiếu cũng như lỗi ngay tại bước thu thập dữ liệu ban đầu.

3.4.2. Dữ liệu bị nhiễu

Nhiễu là những lỗi ngẫu nhiên hoặc những sai lệch trong các giá trị đo đạc được. Có nhiều phương pháp đã được sử dụng để loại bỏ nhiễu, dưới đây xin giới thiệu một số phương pháp thông dụng.

+ *Phương pháp binning*: phương pháp này sẽ gán giá trị cho nhóm dữ liệu đã được sắp xếp bằng cách tham khảo các giá trị lân cận. Các giá trị đã được sắp xếp được phân phối vào các nhóm số tương ứng. Tiếp theo ta áp dụng phương pháp làm mịn phù hợp với từng kiểu dữ liệu. Hình 3.9 mô tả một số phương pháp làm mịn. Trong ví dụ này ta có thuộc tính *price* có giá trị từ 4 cho đến 34 và được đánh giá là dữ liệu nhiễu. Để khử nhiễu ta sắp xếp danh sách giá trị của thuộc tính này rồi chia thành 3 nhóm (trong trường hợp này ta chia sao cho số lượng trong mỗi nhóm là đều nhau), sau đó ta gán lại giá trị cho các phần tử trong từng nhóm các giá trị mới. Trường hợp thứ nhất các giá trị mới này là giá trị trung bình, trường hợp thứ 2 là gán giá trị cho các phần tử ở giữa bằng giá trị của phần tử ngoài biên.

Bảng 3.1. Phương pháp làm mịn dữ liệu Binning

Trường giá trị của thuộc tính *price* sau khi sắp xếp: 4, 8, 15, 21, 21, 24, 25, 28, 34

Phân chia dữ liệu trên thành các nhóm (bin) dựa theo số lượng

4, 8, 15

21, 21, 24

25, 28, 34

Làm mịn bằng giá trị trung bình của từng nhóm

9, 9, 9

22, 22, 22

29, 29, 29

Làm mịn bằng giá trị biên của từng nhóm

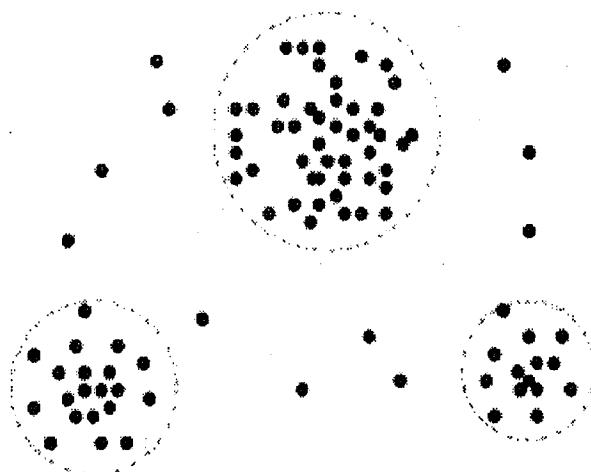
4, 4, 15

21, 21, 24

25, 25, 34

Tương tự, ta có thể áp dụng phương pháp làm mịn dựa trên giá trị trung vị. Trong phương pháp làm mịn dựa trên các giá trị biên, thì giá trị lớn nhất và nhỏ nhất được sử dụng. Mỗi giá trị trong các nhóm số tương ứng sẽ được thay thế bằng giá trị lớn nhất hay nhỏ nhất tương ứng tùy thuộc vào giá trị nào gần nó hơn. Phương pháp này cũng được sử dụng như là một phương pháp rời rạc hóa dữ liệu sẽ được trình bày trong mục 3.7.

+ Phương pháp hồi quy (regression): hồi quy là phương pháp tìm ra một hàm số biểu diễn dữ liệu, có nhiều phương pháp hồi quy. Hồi quy tuyến tính (linear regression) là phương pháp tìm ra đường thẳng tốt nhất biểu diễn quan hệ giữa hai thuộc tính, bằng cách này thì một thuộc tính có thể suy diễn ra thuộc tính còn lại. Hồi quy tuyến tính đa trị (multiple linear regression) là trường hợp mở rộng của hồi quy tuyến tính trong đó có nhiều hơn hai thuộc tính tham gia và dữ liệu được biểu diễn trên không gian đa chiều.



Hình 3.9. Ví dụ về phân cụm và giá trị ngoại lai

+ *Phương pháp phân tích ngoại lai* (outlier analysis): các giá trị ngoại lai có thể được phát hiện thông qua phương pháp phân cụm (clustering), các giá trị tương đồng với nhau sẽ được gom lại thành các nhóm có cùng tính chất. Một cách trực quan, ta có thể nhận thấy các giá trị nằm ngoài các cụm có thể được coi là các giá trị ngoại lai như mô tả trên Hình 3.9.

3.4.3. Làm sạch dữ liệu phải là một quy trình

Giá trị bị thiếu, nhiễu và không nhất quán làm cho dữ liệu không còn chính xác. Như đã trình bày ở các phần trước, chúng ta đã tìm kiếm các phương pháp để thực hiện việc loại bỏ các giá trị bị thiếu và làm mịn dữ liệu.

Có thể dễ dàng thấy đây là một công việc không hề đơn giản, có thể coi nó là một công việc rất lớn tương đương với một quy trình. Bước đầu tiên trong quy trình làm sạch dữ liệu là phát hiện ra các bất thường trong dữ liệu. Sự bất thường này có thể đến từ nhiều nguồn khác nhau như do thiết kế mẫu nhập liệu với quá nhiều trường tùy chọn, hay do lỗi người nhập liệu, lỗi do cố ý, hay thông tin không được cập nhật. Sự bất thường cũng có thể bắt nguồn từ thể hiện của dữ liệu không nhất quán hoặc cách sử dụng các định dạng biểu diễn khác nhau. Một nguyên nhân gây sự bất thường hay gấp khác đó là do hỏng hóc ngay trong thiết bị thu nhận dữ liệu hoặc lỗi hệ thống. Lỗi cũng có thể gấp phải khi dữ liệu được sử dụng sai với mục đích ban đầu. Sự bất thường dữ liệu cũng có thể được sinh ra trong quá trình tích hợp dữ liệu.

Câu hỏi đặt ra là làm thế nào để thực hiện việc phát hiện ra sự bất thường trong dữ liệu? Thông thường ta có thể bắt đầu bằng việc sử dụng tất cả những tri thức sẵn có đối với các tính chất của dữ liệu. Những hiểu biết này có thể được hiểu như là siêu dữ liệu (metadata) hay bản chất nó là “dữ liệu về dữ liệu”.

Ví dụ về siêu dữ liệu là: kiểu dữ liệu và miền giá trị cho từng thuộc tính; giá trị có thể cho mỗi thuộc tính. Sử dụng một số phương pháp phân tích thống kê đơn giản như tìm trung bình, trung vị, độ lệch chuẩn,... có thể giúp tìm ra xu hướng của dữ liệu và xác định được những dị thường trong dữ liệu; kiểm tra xem dữ liệu là đối xứng hay bất đối xứng; tìm khoảng biến thiên của các giá trị; tìm độ lệch chuẩn của mỗi thuộc tính; tìm sự phụ thuộc giữa hai thuộc tính bất kỳ... Trong bước này, ta có thể tự viết chương trình hoặc sử dụng các công cụ có sẵn để thực hiện. Từ đó ta có thể phát hiện ra nhiễu, sự bất thường, các giá trị không bình thường cần được nghiên cứu.

Là người phân tích dữ liệu, ta cần phân tích tìm hiểu mọi sự mâu thuẫn trong việc sử dụng chuẩn hoặc định dạng dữ liệu. Ví dụ như thuộc tính thời gian có thể được biểu diễn bằng các chuẩn khác nhau như năm trước, tháng rồi mới đến ngày, có chuẩn lại biểu diễn ngày trước. Hoặc cùng một chuẩn lại được biểu diễn bằng nhiều định dạng khác nhau như “YYYY/MM/DD” và “YY/MM/DD”.

Dữ liệu cũng nên được kiểm tra dựa trên một số luật bao gồm quy tắc duy nhất, liên tục và quy tắc NULL. **Quy tắc duy nhất** phát biểu như sau: mỗi giá trị của một thuộc tính bất kỳ phải khác với tất cả các giá trị còn lại của thuộc tính đó ví dụ như số chứng minh thư. Do đó ta có thể phát hiện ra dữ liệu lỗi nếu có 2 bản ghi có cùng giá trị cho thuộc tính này. **Quy tắc liên tục** không có giá trị nào bị mất giữa giá trị lớn nhất và nhỏ nhất trong cùng một thuộc tính, và các giá trị này là duy nhất (ví dụ như số thẻ sinh viên). **Quy tắc NULL** chỉ rõ cách sử dụng của các ký tự trống, dấu hỏi (?), ký tự đặc biệt hoặc bất kỳ ký hiệu nào khác được dùng để thể hiện trạng thái không có dữ liệu và cách sử dụng giá trị này.

Như đã trình bày ở phần trước, các giá trị bị thiếu có thể bao gồm (1) người được hỏi các giá trị này từ chối cung cấp hoặc không có thông tin để cung cấp (ví dụ như người không có hộ chiếu sẽ không thể điền số hộ chiếu và ngày cấp), (2) người nhập liệu không biết giá trị chính xác của dữ liệu, hoặc (3) dữ liệu sẽ được cung cấp sau. Quy tắc NULL sẽ chỉ ra cách thức lưu dữ liệu trong trường hợp không có dữ liệu.

Ngoài ra ta còn có rất nhiều công cụ có thể sử dụng hỗ trợ cho việc phát hiện sự bất thường trong dữ liệu (độc giả có thể tham khảo các công cụ này ở mục 2.3 trong tài liệu [Han06]).

Một số trường hợp dữ liệu không nhất quán có thể được sửa chữa thủ công bằng việc sử dụng các tham khảo từ dữ liệu gốc. Ví dụ như lỗi nhập liệu có thể được sửa bằng cách tham khảo lại văn bản gốc. Phần lớn các lỗi thường sẽ yêu cầu chuyển đổi dạng dữ liệu. Điều này có nghĩa là khi ta phát hiện ra sự bất thường của dữ liệu, thông thường ta sẽ phải định nghĩa và sử dụng một hoặc nhiều phép biến đổi để hiệu chỉnh chúng.

Có một quy trình gồm hai bước: phát hiện bất thường và chuyển đổi dữ liệu để sửa chữa bất thường này, hai bước này được lặp đi lặp lại. Tuy vậy quy trình này không thể tránh được sai sót và tốn kém thời gian. Một số phép biến đổi còn làm tăng sự bất thường của dữ liệu. Một số bất thường chỉ được phát hiện sau khi được sửa chữa, ví dụ lỗi nhập dữ liệu năm nhầm thành “20004” chỉ có thể phát hiện ra được khi ta chuyển nó về định dạng ngày tháng. Các phép biến đổi thường được thực hiện thành một dãy các chỉ thị lệnh. Người dùng chỉ có thể kiểm tra kết quả thực hiện sau khi các phép biến đổi được thực hiện xong. Thông thường thì các phép biến đổi này được thực hiện lặp đi lặp lại nhiều lần cho đến khi thỏa mãn yêu cầu. Các bộ dữ liệu không thể tự động thực hiện được trên các phép biến đổi sẽ được ghi vào tệp mà không có giải thích gì về lý do phép biến đổi không thực hiện được. Kết quả là toàn bộ quy trình làm sạch dữ liệu cũng chịu thiệt hại do thiếu sự phối hợp giữa các bước.

Các cách tiếp cận mới trong việc làm sạch dữ liệu nhấn mạnh vào việc cải thiện sự phối hợp giữa hai bước này. Ví dụ: bộ công cụ Potter's Wheel.

Một cách tiếp cận khác cũng tăng sự tương tác này là phát triển bộ ngôn ngữ tập trung vào các phép biến đổi dữ liệu. Công việc này tập trung chủ yếu vào các định nghĩa mở rộng của ngôn ngữ SQL và các thuật toán cho phép người sử dụng thực hiện phương pháp làm sạch dữ liệu hiệu quả hơn.

Đồng thời với việc phát hiện ra những đặc tính của dữ liệu, chúng ta cũng phải cập nhật những phát hiện này vào metadata. Những thông tin bổ sung này sẽ giúp cho quá trình làm sạch dữ liệu ngày càng hiệu quả hơn với dữ liệu đã cho.

3.4. TÍCH HỢP DỮ LIỆU

Tích hợp dữ liệu là một bước thường được sử dụng trong khai phá dữ liệu, đây là phương pháp hợp nhất dữ liệu từ nhiều nguồn khác nhau về một nơi, thông thường là kho dữ liệu (data warehouse).

Việc tích hợp dữ liệu một cách cẩn trọng sẽ giúp giảm và tránh được dư thừa cũng như sự không nhất quán của dữ liệu kết quả. Tích hợp dữ liệu sẽ giúp cải tiến hiệu năng và tốc độ của quá trình khai phá dữ liệu. Có rất nhiều vấn đề cần phải giải quyết trong quá trình tích hợp dữ liệu. Dưới đây sẽ giới thiệu một số vấn đề và cách giải quyết phổ biến.

3.4.1. Nhận diện thực thể

Vấn đề đầu tiên là sự không nhất quán về mặt ngữ nghĩa và cấu trúc của dữ liệu đặt ra những thách thức rất lớn trong tích hợp dữ liệu. Tích hợp lược đồ và đối sánh các đối tượng có thể rất phức tạp. Làm thế nào để có thể so khớp, lắp ghép được tập các thực thể từ nhiều nguồn dữ liệu khác nhau? Đây là bài toán nhận diện thực thể (entity identification). Ví dụ có 2 nguồn dữ liệu, làm thế nào để xác định được trường customer_id trong một bảng của CSDL thứ nhất và trường cus_number trong một bảng của CSDL thứ 2 là hai tên khác nhau của cùng một thuộc tính? Để trả lời được câu hỏi này ta có thể tham khảo các thông tin metadata mô tả 2 trường này, bao gồm: tên trường, kiểu dữ liệu; ý nghĩa (mục đích) của trường; miền giá trị cho phép; quy tắc xử lý giá trị null. Sau khi xem xét đầy đủ các thông tin trên và thấy thông tin đồng nhất thì ta có thể ghép thuộc tính customer_id và cus_number làm một thuộc tính duy nhất và chúng ta xử lý xong một trường trong quá trình nhận diện thực thể. Các trường khác trong bảng của 2 CSDL trên cũng được xử lý tương tự.

Trong quá trình so sánh để lắp ghép tập thuộc tính của bộ dữ liệu này với tập thuộc tính của bộ dữ liệu kia, cần đặc biệt quan tâm tới cấu trúc của dữ liệu. Điều này giúp đảm bảo tập các phụ thuộc hàm và các ràng buộc toàn vẹn sẽ được kế thừa vào trong bộ dữ liệu sau khi tích hợp. Ví dụ: hệ thống A thì khách hàng được giảm giá trên mỗi hóa đơn, còn hệ thống B, khách hàng sẽ được giảm giá trên từng mặt hàng trong hóa đơn. Dữ liệu cần được tích hợp bao gồm cả dữ liệu của A và B, nếu ràng buộc này không được kế thừa một cách phù hợp trước khi tích hợp, các mặt hàng trên hệ thống mới sẽ không được giảm giá một cách phù hợp.

Một trường hợp nữa có thể xảy ra trong quá trình tích hợp dữ liệu là có thể không có sự tương đồng về số lượng trường (thuộc tính) giữa 2 bảng trong 2 CSDL chúng ta đang tích hợp. Điều này cũng không có gì ngạc nhiên vì lý do trong CSDL này ta chỉ quan tâm (và cần) một số thuộc tính của thực thể nào đó, ngược lại trong CSDL kia ta lại quan tâm đến một tập hợp thuộc tính khác. Khi đó cách giải quyết đơn giản có thể là tạo ra một bảng mới chứa đầy đủ cả tập thuộc tính của thực thể và tìm cách điền vào các giá trị thiếu.

3.4.2. Sự dư thừa và phân tích độ tương quan

Sự dư thừa là một vấn đề quan trọng khác thường xảy ra trong quá trình tích hợp dữ liệu. Một thuộc tính có thể được coi là dư thừa nếu như nó có thể suy diễn được từ một hoặc một nhóm các thuộc tính khác (ví dụ như: điểm trung bình hoặc, tổng thu nhập...). Sự không nhất quán trong việc đặt tên các thuộc tính có thể gây ra sự dư thừa trong tập dữ liệu.

Một số loại dư thừa có thể được phát hiện nhờ phương pháp phân tích độ tương quan. Với hai thuộc tính cho trước, phương pháp này có thể cho biết mức độ liên quan giữa chúng dựa trên dữ liệu có được. Với những thuộc tính có giá trị thuộc dạng ký tự, ta có thể sử dụng phương pháp χ^2 (chi-square). Với những thuộc tính có tập giá trị có dạng số thì có thể áp dụng phương pháp *Hệ số tương quan* (correlation coefficient) hoặc *Hiệp phương sai* (covariance) để phát hiện ra mức độ phụ thuộc giữa hai thuộc tính bất kỳ.

a) Phương pháp χ^2 (chi-square)

Với dữ liệu số, độ tương quan giữa 2 thuộc tính A và B có thể được tìm thông qua phương pháp *Khi bình phương*. Giả sử thuộc tính A có c giá trị lần lượt a_1, a_2, \dots, a_c , và thuộc tính B có r giá trị tương ứng b_1, b_2, \dots, b_r . Dữ liệu được mô tả bởi A và B có thể được xếp thành 1 bảng với c giá trị của A xếp thành cột và r giá trị của B xếp thành hàng. Gọi (A_i, B_j) là sự kiện đồng thời $A = a_i$ và $B = b_j$.

Mỗi một khả năng có thể của (A_i, B_j) đều được ghi lại trong 1 ô của bảng này. Giá trị χ^2 được tính như sau

$$\chi^2 = \sum_{i=1}^c \sum_{j=1}^r \frac{o_{ij} - e_{ij}}{e_{ij}} \quad (3.5)$$

Với o_{ij} là giá trị quan sát (giá trị thực tế) và e_{ij} là giá trị kỳ vọng của sự kiện (A_i, B_j) được tính theo công thức sau:

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{n} \quad (3.6)$$

Với n là số mẫu dữ liệu, $\text{count}(A = a_i)$ là số lượng mẫu có giá trị a_i trong thuộc tính A , tương tự với $\text{count}(B = b_j)$. Công thức (3.5) sử dụng để tính tất cả các giá trị trên $r \times c$ ô của bảng.

Phương pháp này dùng để kiểm tra giả thiết A và B là độc lập với nhau (không có mối liên hệ nào giữa chúng) hay không. Kiểm tra này dựa trên mức độ quan trọng với $(r-1) \times (c-1)$ mức độ tự do. Nếu phép kiểm tra là không đúng thì điều đó có nghĩa là A và B là có tương quan với nhau về mặt thống kê.

Ví dụ, ta có điều tra 1500 người xem họ có thích đọc truyện viễn tưởng (fiction) hay không. Kết quả của cuộc điều tra được liệt kê trong bảng 3.2, trong đó ta có 2 thuộc tính là *giới tính*, và *sở thích đọc truyện*:

Bảng 3.2. Dữ liệu điều tra về sở thích đọc truyện

	Nam	Nữ	Tổng số
Fiction	250 (90)	200 (360)	450
Nonfiction	50 (210)	1000 (840)	1050
Tổng số	300	1200	1500

Trong bảng này giá trị kỳ vọng e_{ij} (tính theo công thức (3.2)) được ghi ở trong ngoặc đơn, ví dụ:

$$e_{11} = \frac{\text{count}(\text{nam}) \times \text{count}(\text{fiction})}{n} = \frac{300 \times 450}{1500} = 90$$

Do đó ta có thể tính ra được giá trị χ^2 như sau:

$$\begin{aligned}\chi^2 &= \frac{(250 - 90)^2}{90} + \frac{(50 - 210)^2}{210} + \frac{(200 - 360)^2}{360} + \frac{(1000 - 840)^2}{840} \\ &= 507.93\end{aligned}$$

Mức độ tự do của phép kiểm tra đối với bảng 2×2 là $(2-1) \times (2-1) = 1$, ở mức này giá trị phủ định giả thiết ở mức độ quan trọng 0.001 là 10.827 (giá trị này có thể tra bảng ở các sách thống kê, bảng 3.3 liệt kê một số giá trị này). Giá trị $507.93 > 10.827$ có nghĩa là giá trị này đã phủ định giả thiết *giới tính và sở thích đọc truyện viễn tưởng* là độc lập nhau. Hay ta có thể nói hai thuộc tính trên là có độ tương quan cao trong tập dữ liệu ở trên.

Bảng 3.3. Giá trị mức xác suất χ^2

	0.5	0.10	0.05	0.02	0.01	0.001
1	0.455	2.706	3.841	5.412	6.635	10.827
2	1.386	4.605	5.991	7.824	9.210	13.815
3	2.366	6.251	7.815	9.837	11.345	16.268
4	3.357	7.779	9.488	11.668	13.277	18.465
5	4.351	9.236	11.070	13.388	15.086	20.51

b) Phương pháp hệ số tương quan

Với các thuộc tính số, ta có thể tính toán độ phụ thuộc giữa chúng bằng phương pháp *hệ số tương quan* do Karl Pearson đề xuất:

$$r_{A,B} = \frac{\sum_{i=1}^N (a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^N a_i b_i - N\bar{A}\bar{B}}{N\sigma_A\sigma_B} \quad (3.7)$$

trong đó, N là số lượng mẫu dữ liệu, a_i và b_i lần lượt là các giá trị tương ứng của thuộc tính A và B ở mẫu thứ i , \bar{A} và \bar{B} là giá trị trung bình của A và B , σ_A và σ_B là các độ lệch chuẩn tương ứng của A và B .

$$\bar{A} = \frac{\sum_{i=1}^N a_i}{N} \text{ và } \bar{B} = \frac{\sum_{i=1}^N b_i}{N}$$

Lưu ý rằng $-1 \leq r_{A,B} \leq +1$, nếu $r_{A,B}$ nhận giá trị dương thì A và B là đồng biến, tức là nếu giá trị của A tăng thì giá trị của B cũng tăng theo. Giá trị của $r_{A,B}$ càng lớn thì độ phụ thuộc càng mạnh. Do đó nếu $r_{A,B}$ có giá trị đủ lớn thì ta có thể loại bỏ thuộc tính A hoặc B . Nếu kết quả của công thức (3.3) là 0 điều này có nghĩa là A độc lập với B (hay không có sự liên hệ giữa chúng). Còn nếu giá trị này là âm thì giá trị của thuộc tính A nghịch biến với B . Lưu ý rằng sự tương quan này không có hàm ý nhân quả. Nếu A và B tương quan với nhau thì vai trò của A và B ở đây là tương đương. Ví dụ: nếu chúng ta tìm ra sự liên quan giữa số lượng bệnh viện và số ô tô bị mất cắp trong vùng. Điều đó không có nghĩa là số lượng bệnh viện là nguyên nhân gây ra số vụ đánh cắp xe.

c) Phương pháp hiệp phương sai

Trong lý thuyết xác suất và thống kê, *độ đo tương quan* và *hiệp phương sai* là hai độ đo có cùng ý nghĩa nhằm ước lượng xem hai thuộc tính ảnh hưởng lẫn nhau như thế nào. Xét hai thuộc tính A và B , với N là mẫu dữ liệu có dạng $\{(a_1, b_1), \dots, (a_N, b_N)\}$. Hiệp phương sai (covariance) giữa A và B được định nghĩa như sau:

$$Cov(A, B) = \frac{\sum_{i=1}^N (a_i - \bar{A})(b_i - \bar{B})}{N} \quad (3.8)$$

So sánh công thức (3.7) và công thức (3.8) ta có thể viết lại như sau (để ta có thể thấy được sự tương đồng của 2 công thức):

$$r_{A,B} = \frac{Cov(A, B)}{\sigma_A \sigma_B} \quad (3.9)$$

Với hai thuộc tính A và B có khuynh hướng thay đổi với nhau. Nếu A lớn hơn \bar{A} thì B cũng có xu hướng lớn hơn \bar{B} thì

$Cov(A, B)$ có giá trị dương. Ngược lại thì $Cov(A, B)$ có giá trị âm nếu một thuộc tính có khuynh hướng cao hơn giá trị trung bình thì thuộc tính còn lại có khuynh hướng nhỏ hơn giá trị này của nó.

3.4.3. Phát hiện các bộ lặp

Trong quá trình phát hiện sự dư thừa giữa các thuộc tính, việc phát hiện và loại bỏ các bản ghi lặp lại (hay trùng nhau) cũng là vấn đề đáng được quan tâm. Sự không nhất quán thường gia tăng khi có nhiều bản ghi bị lặp do sai sót trong quá trình đưa dữ liệu vào hoặc cập nhật dữ liệu không đúng cách. Ví dụ: trong CSDL bán hàng, nếu ta dùng tên khách hàng và địa chỉ khách hàng làm khóa, khi đó do lỗi nhập liệu, trường địa chỉ của cùng 1 khách hàng có thể được nhập không đúng theo một định dạng duy nhất, lúc đó cùng một khách hàng ta lại có thể tồn tại nhiều bản ghi khác nhau trong bảng (nhưng về bản chất nó là một).

3.4.4. Phát hiện xung đột trong dữ liệu và mức độ trừu tượng

Tích hợp dữ liệu cũng bao hàm việc phát hiện xung đột và thay đổi mức độ trừu tượng trong giá trị của dữ liệu. Ví dụ: với cùng một đối tượng trong thế giới thực thì mỗi CSDL sẽ có cách biểu diễn, mô tả hay mã hóa khác nhau. Chẳng hạn như nhiệt độ có thể biểu diễn theo nhiều độ đo khác nhau (độ C hoặc độ F). Thông tin về điểm số của học sinh ở mỗi trường cũng có thể khác nhau như theo thang điểm 10, thang điểm 4, hoặc thang điểm chữ... Cách tính điểm khác nhau như vậy sẽ gây khó khăn trong việc chuyển đổi dữ liệu từ hệ thống nọ sang hệ thống kia và ngược lại. Trong quá trình tích hợp dữ liệu ta cũng cần phải xử lý trường hợp này, cụ thể là phải chọn một định dạng dữ liệu duy nhất và chuyển đổi các kiểu dữ liệu khác sang.

Các thuộc tính cũng có thể có mức độ trừu tượng khác nhau giữa các hệ thống tùy theo nhu cầu của hệ thống đó. Ví dụ như tổng số sinh viên có thể là sinh viên của một lớp trong hệ thống niên chế nhưng sẽ là tổng số sinh viên của một lớp-môn học trong hệ thống tín chỉ.

3.5. CHUYỂN ĐỔI DỮ LIỆU

3.5.1. Các chiến lược chuyển đổi dữ liệu

Trong các phép biến đổi dữ liệu, dữ liệu sẽ được chuyển đổi hoặc hợp nhất vào các định dạng phù hợp cho việc khai phá dữ liệu. Chiến lược cho các phép biến đổi này bao gồm:

- ✓ **Làm mịn (smoothing):** loại bỏ nhiễu trong dữ liệu (trình bày trong phần 3.4.1)
- ✓ **Tổng hợp (aggregation):** thực hiện các thao tác tổng hợp (chẳng hạn như phép tính tổng) trên dữ liệu. Ví dụ ta có thể tính tổng doanh thu theo ngày, tháng hoặc năm. Thao tác này thường được sử dụng trong quá trình xây dựng khối dữ liệu (data cube) để phân tích dữ liệu ở nhiều mức chi tiết khác nhau.
- ✓ **Khái quát hóa (generalization) dữ liệu:** dữ liệu thô ban đầu sẽ được thay thế bằng các khái niệm ở mức cao hơn (trong cây phân cấp khái niệm). Ví dụ như thuộc tính *phố* (street) có giá trị rời rạc, thuộc tính này có thể được khái quát hóa lên bằng thuộc tính *thành phố* (city) hay *đất nước* (country) tùy theo mục đích. Thuộc tính có giá trị số là *tuổi* (age) cũng có thể được khái quát hóa thành thuộc tính ở mức cao hơn như *trẻ* (youth), trung niên (middle-age) và cao niên (senior).
- ✓ **Xây dựng các thuộc tính (attribute construction):** các thuộc tính được xây dựng thêm từ dữ liệu gốc nhằm hỗ trợ cho quá trình khai phá dữ liệu.
- ✓ **Chuẩn hóa:** biến đổi miền giá trị của các thuộc tính về những miền giá trị nhỏ hơn ví dụ như [0.0, 1.0] hoặc [-1.0, 1.0] nhằm làm cho các giải thuật khai phá hoạt động hiệu quả hơn.
- ✓ **Rời rạc hóa:** khi những giá trị số trong miền liên tục sẽ được chuyển về các khoảng số được gán nhãn (VD: như 0-10, 11-20, ...) hoặc các khoảng gán nhãn (thiếu niên,

thanh niên và trung niên, ...). Các mức này hoàn toàn có thể được gom nhóm lại với mức cao hơn tùy yêu cầu để tạo nên khái niệm phân cấp cho các thuộc tính.

Rời rạc hóa dữ liệu có thể phân loại dựa trên cách thức thực hiện rời rạc hóa, như có sử dụng thông tin phân lớp hoặc cách thức thực hiện từ trên xuống (topdown), hay từ dưới lên (bottom-up). Nếu phương pháp rời rạc hóa dùng thông tin phân lớp nó được gọi là rời rạc hóa có giám sát, nếu không thì được gọi là không có giám sát.

3.5.2. Chuẩn hóa dữ liệu

Các đại lượng đo đạc có thể ảnh hưởng tới phân tích dữ liệu. Ví dụ sự thay đổi các đại lượng giữa độ C và độ F trong đo đạc nhiệt độ, giữa mét và inch trong đo độ dài, có thể dẫn tới các kết quả khác nhau. Nói chung khi chia nhỏ một thuộc tính thì đồng nghĩa với việc mở rộng miền giá trị của thuộc tính đó, sẽ dẫn tới việc thuộc tính đó có ảnh hưởng lớn hơn các thuộc tính khác. Để tránh sự phụ thuộc vào cách chọn đại lượng đo lường, dữ liệu nên được *chuẩn hóa* trước khi sử dụng. Một trong số các cách thường dùng là chuyển miền dữ liệu về nằm trong khoảng [-1, 1] hoặc [0.0, 1.0].

Có rất nhiều phương pháp chuẩn hóa dữ liệu. Ở đây chúng ta chỉ trình bày một số phương pháp cơ bản bao gồm chuẩn hóa min-max, z-score,...

Gọi A là thuộc tính có kiểu số với n giá trị v_1, v_2, \dots, v_n .

Chuẩn hóa min-max: thực hiện một phép biến đổi tuyến tính trên dữ liệu gốc. Giả sử \min_A và \max_A là giá trị nhỏ nhất và lớn nhất của thuộc tính A . Phương pháp này chuyển một giá trị v_i thành giá trị v'_i trong miền $[\min'_A, \max'_A]$ được tính như sau:

$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\max'_A - \min'_A) + \min'_A \quad (3.10)$$

Phương pháp này bảo tồn được quan hệ giữa các giá trị trong dữ liệu gốc. Dữ liệu sau khi chuẩn hóa có thể bị lỗi nếu dữ liệu mới nằm ngoài khoảng giá trị của dữ liệu gốc.

Ví dụ: giả sử giá trị min và max của thuộc tính thu nhập (income) là 12000\$ và 98000\$, ta muốn chuyển đổi về khoảng [0, 1]. Khi đó thu nhập có giá trị 73600\$ sẽ có giá trị mới là

$$v_i = \frac{73600 - 12000}{98000 - 12000} (1 - 0) + 0 = 0,716$$

Chuẩn hóa z-score: các giá trị của thuộc tính A sẽ được chuẩn hóa dựa trên giá trị trung bình và độ lệch chuẩn của A. Giá trị v'_i sẽ được tính toán dựa trên v_i như sau:

$$v'_i = \frac{v_i - \bar{A}}{\sigma_A} \quad (3.11)$$

với \bar{A} là giá trị trung bình và σ_A là độ lệch chuẩn. Phương pháp chuẩn hóa này thường được sử dụng trong trường hợp không xác định được chính xác giá trị lớn nhất và giá trị nhỏ nhất của thuộc tính hoặc trong trường hợp các giá trị ngoại lai chi phối phương pháp chuẩn hóa min-max.

$$\sigma_A^2 = \frac{\sum_{i=1}^n (v_i - \bar{A})^2}{n} \quad (3.12)$$

Một biến thể thường được sử dụng trong chuẩn hóa này là thay thế σ_A bằng giá trị trung bình tuyệt đối. Giá trị này, ký hiệu là S_A được tính như sau:

$$S_A = \frac{1}{n} \sum_{i=1}^n |v_i - \bar{A}| \quad (3.13)$$

Chuẩn hóa thay đổi tỉ lệ (scaling): Giá trị mới $v' = v/10^j$ với j là số nguyên nhỏ nhất thỏa mãn điều kiện $\max(|v'|) < 1$. Ví dụ: giả sử thuộc tính A có giá trị từ -986 đến 917. Giá trị tuyệt đối lớn nhất của thuộc tính này là 986 (xấp xỉ 10^3), để chuẩn hóa sang tỉ lệ mới ta có thể chia cho 10^3 , khi đó giá trị 917 sẽ có giá trị mới là 0,917.

3.6. PHƯƠNG PHÁP THU GỌN DỮ LIỆU

Phương pháp thu gọn dữ liệu có thể được áp dụng nhằm giảm lượng dữ liệu nhiều nhất có thể mà vẫn giữ được tính toàn vẹn của dữ liệu gốc. Điều này có nghĩa rằng các phương pháp phân tích dữ liệu khi thực hiện một cách hiệu quả hơn trên dữ liệu đã thu gọn mà vẫn trả lại kết quả phân tích như khi thực hiện trên dữ liệu gốc (hoặc gần tốt như thực hiện trên dữ liệu gốc).

Các chiến lược thu gọn dữ liệu bao gồm giảm số chiều của dữ liệu, giảm số lượng dữ liệu (numerosity reduction) và nén dữ liệu.

Giảm chiều dữ liệu: là quá trình làm giảm bớt số lượng các thuộc tính theo một chiến lược nào đó. Các phương pháp giảm chiều dữ liệu bao gồm phép biến đổi wavelet, PCA. Trích chọn tập các thuộc tính là một trong các phương pháp giảm chiều dữ liệu dựa trên việc phát hiện và loại bỏ các thuộc tính thừa, thuộc tính ít phù hợp và không phù hợp.

Giảm số lượng dữ liệu: cho phép thay thế dữ liệu gốc bằng một cách thể hiện khác với không gian nhỏ hơn dữ liệu gốc. Phương pháp này có thể có hoặc không có tham số. Với phương pháp có tham số, mô hình được sử dụng để ước lượng dữ liệu, vì vậy thông thường ta chỉ cần lưu trữ các tham số của dữ liệu mà không cần lưu toàn bộ dữ liệu gốc (ví dụ như mô hình hồi quy). Phương pháp không dùng tham số bao gồm phân phối theo tần suất, phân cụm, phương pháp lấy mẫu.

3.6.1. Giảm chiều dữ liệu

Lựa chọn tập con thuộc tính (attribute subset selection): có nhiều trường hợp tập dữ liệu chúng ta cần khai phá có chứa hàng trăm (thậm chí hàng nghìn) thuộc tính. Đặc biệt là có nhiều thuộc tính không có ý nghĩa, hoặc dư thừa trong quá trình khai phá dữ liệu, chẳng hạn như thuộc tính số chứng minh thư hay số điện thoại. Nên nếu bỏ đi được các thuộc tính này không những làm giảm được chiều dữ liệu, làm giảm được thời gian xử lý mà còn có thể làm tăng được hiệu quả của các giải thuật khai phá.

Mục đích của phương pháp lựa chọn tập con thuộc tính là tìm ra được tập con thuộc tính nhỏ nhất mà vẫn biểu diễn được sự phân bố của dữ liệu gốc ban đầu.

Cho n thuộc tính, ta có 2^n tập con, nên việc tìm ra tập con tốt nhất là một bài toán có chi phí rất cao nếu ta xét từng tập con một. Thay vì tìm kiếm tập con thuộc tính tốt nhất, ta có thể sử dụng các thuật toán dựa trên kinh nghiệm, ví dụ là thuật toán tham lam (greedy), để lựa chọn thuộc tính tốt nhất tại mỗi bước. Đây là giải pháp dựa vào tối ưu cục bộ để hy vọng tìm ra tối ưu toàn cục. Các phương pháp này là một lựa chọn tốt trong thực tế khi số lượng các thuộc tính trong tập dữ liệu là lớn. Các thuộc tính tốt nhất và xấu nhất có thể xác định được thông qua các độ đo dựa trên thống kê. Có rất nhiều độ đo, ví dụ là độ đo *độ lợi thông tin* (information gain) được đề cập ở mục 6.2 chương 6. Các phương pháp dựa trên kinh nghiệm có các chiến lược sau:

- **Lựa chọn dần từng thuộc tính** (stepwise forward selection): Thủ tục bắt đầu từ tập rỗng, tại mỗi bước nó lựa chọn thêm một thuộc tính được đánh giá là tốt nhất (trong những thuộc tính còn lại) và cho vào tập. Quá trình này lặp lại cho đến hết các thuộc tính. Sau khi thủ tục hoàn thành ta có một danh sách các thuộc tính đã được xếp hạng giảm dần theo độ “tốt”, tùy vào trường hợp cụ thể ta có thể quyết định lấy tập con với số lượng là bao nhiêu từ đầu danh sách đã được sắp xếp này.
- **Loại bỏ dần từng thuộc tính** (stepwise backward elimination): Thủ tục bắt đầu từ tập toàn bộ các thuộc tính, tại từng bước lặp nó loại bỏ một thuộc tính được đánh giá là xấu nhất. Kết quả ta cũng thu được một danh sách đã sắp xếp các thuộc tính theo thứ tự giảm dần của độ “tốt” và việc lựa chọn lại giống như trường hợp ở trên.
- **Kết hợp cả lựa chọn và loại bỏ thuộc tính**: Giải thuật kết hợp cả 2 giải thuật trên lại để vừa chọn thuộc tính tốt nhất vừa loại bỏ thuộc tính xấu nhất tại mỗi bước.

- **Sử dụng cây quyết định:** Ta cũng có thể sử dụng cây quyết định như ID3, C4.5 hay CART (xem ở chương 6) để lựa chọn danh sách các thuộc tính tốt. Điểm mạnh của cây quyết định là nó sẽ không đưa vào cây các thuộc tính được đánh giá là “không liên quan”, do đó ta có thể sử dụng luôn các thuộc tính xuất hiện ở trên cây làm tập thuộc tính con tốt nhất mà không cần phải xử lý thêm như ở các phương pháp ở trên.

Giảm số chiều bằng phương pháp biến đổi: Một phương pháp khác để giảm số chiều là biến đổi (hay mã hóa) dữ liệu sang một dạng khác. Nếu dữ liệu sau khi biến đổi có thể tái xây dựng lại được thành dữ liệu gốc thì phương pháp biến đổi đó được gọi là không mất mát (lossless), nếu không thì phương pháp đó được gọi là biến đổi có mất mát (lossy). Dưới đây sẽ trình bày sơ lược 2 phương pháp biến đổi (có mất mát) thông dụng là phép biến đổi rời rạc dạng sóng (wavelet), và phương pháp phân tích thành phần chính (Principal Component Analysis).

- **Phép biến đổi rời rạc dạng sóng** (Discrete Wavelet Transform – DWT): là một phương pháp xử lý tín hiệu số, được sử dụng khi biến đổi một véctơ X thành một véctơ X' khác (có cùng kích thước) theo hệ số wavelet. Tuy rằng véctơ kết quả X' sau khi biến đổi có cùng kích thước với véctơ ban đầu, nhưng ta có thể làm giảm số chiều của X' bằng cách chỉ giữ lại các hệ số wavelet có trọng số lớn. Chẳng hạn với một ngưỡng đầu vào, ta chỉ giữ lại các thành phần véctơ có trọng số wavelet lớn hơn ngưỡng và loại bỏ những thành phần còn lại. Sau khi biến đổi không gian mới có thể rất thưa so với không gian ban đầu, do đó các giải thuật chuyên xử lý dữ liệu thưa sẽ rất phù hợp với phép biến đổi wavelet. Khi đã có một tập các hệ số wavelet, ta có thể xây dựng lại dữ liệu xấp xỉ với dữ liệu ban đầu từ dữ liệu sau khi được biến đổi. Phương pháp biến đổi này khá giống với phương pháp biến đổi Fourier (Discrete Fourier Transform -DFT), nhưng ưu điểm của DWT là tạo ra một véctơ xấp xỉ với dữ liệu gốc tốt hơn

DFT. Hơn nữa nếu biến đổi Fourier chỉ có một thì biến đổi wavelet lại có rất nhiều họ hàm.

- **Phân tích thành phần chính:** Giả sử các phần tử dữ liệu được biểu diễn bằng một véctơ n chiều, phương pháp phân tích thành phần chính (hay còn gọi là phương pháp Karhunen-Loeve hay K-L cho gọn) sẽ tìm k véctơ trực giao n chiều có thể dùng để biểu diễn dữ liệu, với $k \leq n$. Do vậy dữ liệu ban đầu có thể được biểu diễn bằng một không gian nhỏ hơn bằng phép chiếu trên không gian k chiều. Khác với phương pháp lựa chọn tập con thuộc tính – nó chọn ra một tập con thuộc tính từ tập thuộc tính ban đầu, phương pháp phân tích thành phần chính *kết hợp* bản chất của các thuộc tính lại với nhau để tạo ra thuộc tính mới để thay thế. Chi tiết của phương pháp này nằm ngoài phạm vi của cuốn giáo trình này.

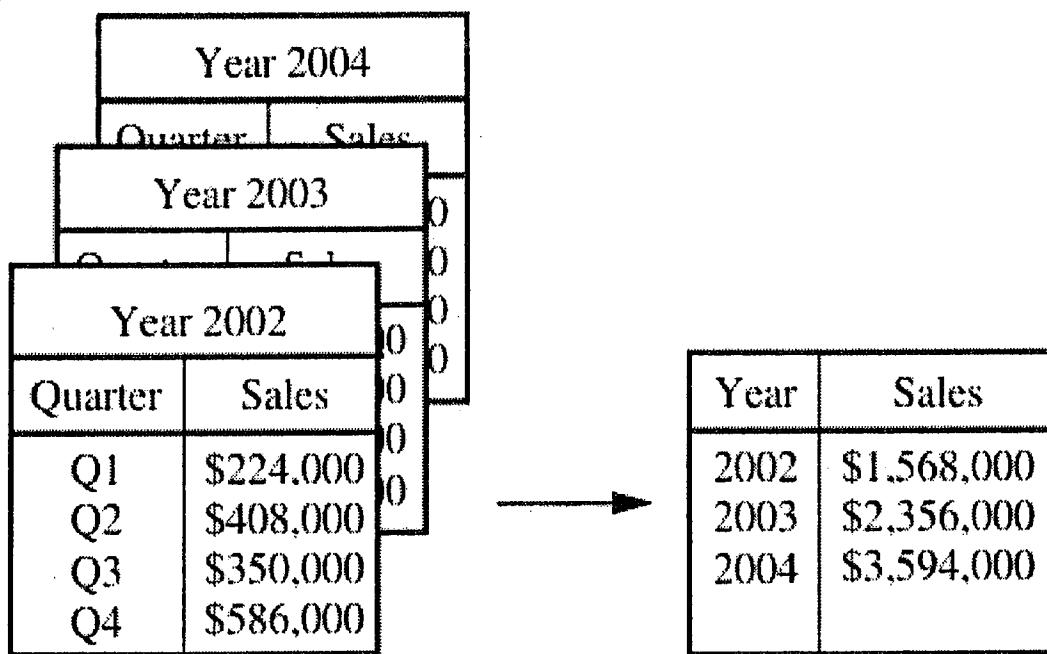
3.6.2. Giảm số lượng dữ liệu

Giảm số lượng dữ liệu là phương pháp thay thế dữ liệu gốc bằng một cách thể hiện khác với không gian nhỏ hơn dữ liệu gốc. Dưới đây sẽ trình bày một số phương pháp phổ biến.

- **Tổng hợp khối dữ liệu** (data cube aggregation): Thực hiện các phép toán tổng hợp (aggregation) trên dữ liệu trong quá trình xây dựng khối dữ liệu. Giả sử trong một cơ sở dữ liệu bán hàng ta có cột *tiền* (sales) cho từng mặt hàng mà khách hàng đã mua. Nếu chỉ muốn xem dữ liệu doanh số bán hàng của một ngày ta có thể tổng hợp tổng tiền của tất cả các mặt hàng trong ngày lại. Tương tự ta có thể tổng hợp doanh số theo tháng, quý, năm tùy theo nhu cầu phân tích số liệu. Như vậy cứ mỗi lần tổng hợp thì số lượng dữ liệu lại giảm đi rất nhiều, đặc biệt tuy dữ liệu giảm đi nhưng số liệu cuối cùng lại không hề bị ảnh hưởng. Phương pháp này được áp dụng vào trong quá trình xây dựng khối dữ liệu. Hình 3.11 và 3.12 minh họa về phương pháp này.

- **Mô hình hồi quy và tuyến tính logarit:** Trong trường hợp hồi quy tuyến tính đơn giản, dữ liệu được mô hình hóa để có thể biểu diễn được bằng một đường thẳng. Cụ thể quan hệ giữa 2 biến có thể được biểu diễn bằng phương trình $y = wx + b$. Trong bài toán khai phá dữ liệu thì x và y là biến biểu diễn các thuộc tính, còn w và b được gọi là các hệ số hồi quy. Sau khi xây dựng được phương trình biểu diễn đường thẳng trên thì ta chỉ cần lưu lại các tham số hồi quy w và b mà không cần phải lưu trữ dữ liệu thực sự, kết quả là ta làm giảm được số lượng dữ liệu. Hồi quy tuyến tính logarit (log-linear) xấp xỉ phân bố xác suất đa chiều rời rạc. Cho một tập dữ liệu được biểu diễn bằng các vectơ n chiều (dữ liệu có n thuộc tính), ta có thể coi 1 phần tử dữ liệu là một điểm trong không gian n chiều. Mô hình tuyến tính logarit có thể ước lượng xác suất của tổng điểm trong không gian đa chiều cho một tập các thuộc tính được rời rạc hóa dựa trên một tập con chiều không gian nhỏ hơn. Sau khi ước lượng xong, ta cũng chỉ cần giữ lại các tham số ước lượng mà không cần phải lưu lại dữ liệu. Ngoài ra vì mô hình tuyến tính logarit có thể biểu diễn dữ liệu gốc bằng một không gian có số chiều nhỏ hơn nên nó cũng có thể sử dụng để làm giảm số chiều dữ liệu. Phương pháp hồi quy và tuyến tính logarit thuộc lớp phương pháp có tham số.
- **Biểu đồ tần suất (histogram):** Phương pháp này xấp xỉ phân bố dữ liệu bằng cách chia dữ liệu thành các nhóm (các tập không giao nhau) dựa trên một thuộc tính nào đó, nếu một nhóm chứa các phần tử dữ liệu có giá trị thuộc tính đang xét là bằng nhau thì ta gọi là nhóm duy nhất (singleton bucket). Sau khi chia xong thì ta dùng các nhóm dữ liệu này để biểu diễn dữ liệu thay vì dữ liệu gốc, do đó số lượng dữ liệu sẽ được giảm đi. Ví dụ ta có giá trị cho thuộc tính giá (price) sau khi sắp xếp tính bằng USD là: 1, 1, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20,

20, 20, 20, 20, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30. Nếu ta chia tập dữ liệu này thành các nhóm có giá trị bằng nhau thì ta thu được 13 nhóm như Hình 3.12. Do đó từ tập dữ liệu ban đầu là 52 phần tử ta chỉ còn 13 phần tử. Trong trường hợp dữ liệu là liên tục thì mỗi một nhóm sẽ có một miền giá trị.

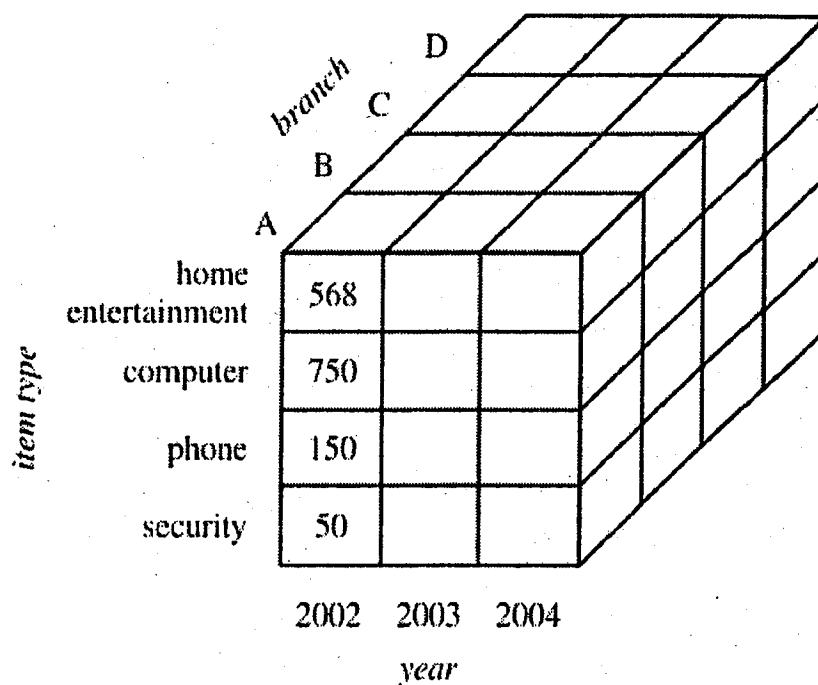


Hình 3.10. Dữ liệu năm được tổng hợp từ dữ liệu quý

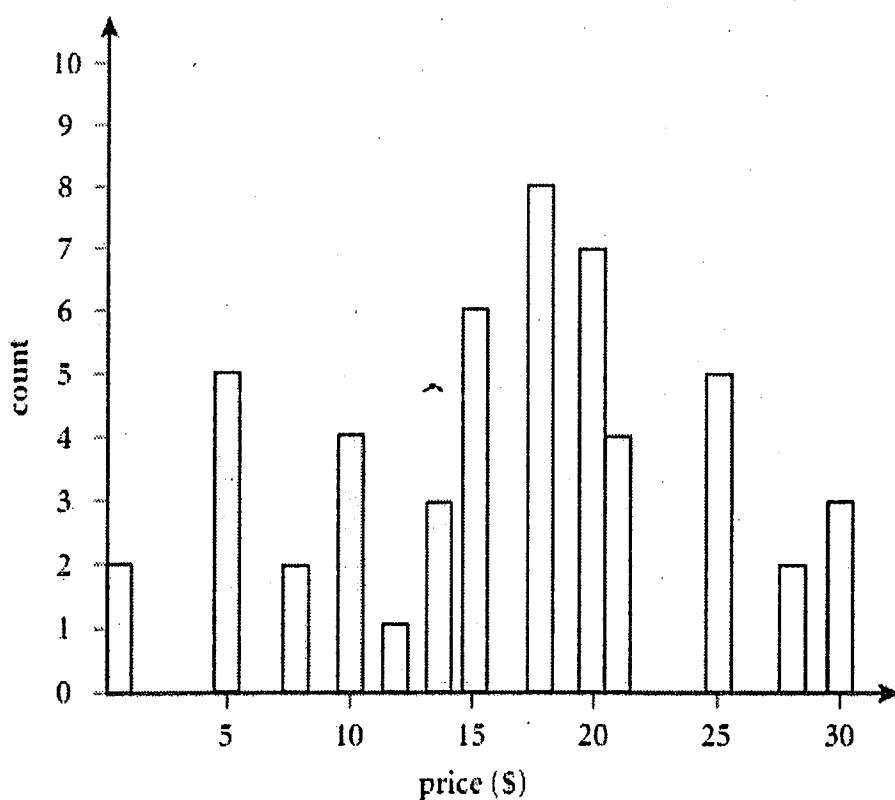
Có nhiều tiêu chí để chia dữ liệu thành các nhóm, dưới đây giới thiệu một số tiêu chí:

- ❖ Độ rộng bằng nhau (equal-width): miền giá trị cho mỗi nhóm là bằng nhau. Ví dụ một thuộc tính có giá trị từ 1 đến 100, nếu chia thành các nhóm có “độ rộng” là 10 thì ta có 10 nhóm ($100/10 = 10$).
- ❖ Bằng tần suất (equal-frequency): số lượng các phần tử dữ liệu trong từng nhóm là (xấp xỉ) bằng nhau.
- **Phân cụm** (clustering): Phương pháp này sử dụng các giải thuật phân cụm để nhóm dữ liệu lại thành các cụm, các cụm này sẽ được dùng làm đại diện cho dữ liệu gốc. Các giải thuật phân cụm sẽ được trình bày ở chương 5.

- **Lấy mẫu** (sampling): phương pháp này chỉ đơn giản là lấy ngẫu nhiên một tập con của dữ liệu. Giả sử ta có một tập dữ liệu lớn D gồm N phần tử dữ liệu, ta có các phương pháp lấy mẫu sau:
 - *Lấy mẫu ngẫu nhiên đơn giản không thay thế*: Ta lấy ngẫu nhiên s phần tử từ D ($s < N$), khi đó mỗi phần tử sẽ có xác suất được lấy là $1/N$.
 - *Lấy mẫu ngẫu nhiên đơn giản có thay thế*: Ta lấy ngẫu nhiên s phần tử từ D ($s < N$), điểm khác so với phương pháp trên là: một phần tử sau khi được lấy mẫu nó lại được bỏ vào tập D , do đó nó có khả năng được lấy mẫu nhiều hơn 1 lần.



Hình 3.11. Khối dữ liệu được tạo ra nhờ các thao tác tổng hợp



Hình 3.12. Các nhóm sau khi chia theo biểu đồ tần suất

- *Lấy mẫu cụm*: Khi D được phân thành M cụm không giao nhau, khi đó ta có thể lấy ngẫu nhiên s cụm ($s < M$).
- *Lấy mẫu theo tầng* (stratified sampling): Giả sử D được phân thành các phần không giao nhau gọi là các tầng (strata). Ta sẽ tiến hành lấy mẫu ngẫu nhiên đơn giản trên từng tầng. Phương pháp này đảm bảo tầng nào cũng được lấy mẫu, do đó dữ liệu mẫu thu được có khả năng đại diện tốt cho dữ liệu gốc.

Các phương pháp lấy mẫu có ưu điểm là chi phí thấp nên có thể ứng dụng trong những trường hợp cần tốc độ xử lý.

3.7. RỜI RẠC HÓA DỮ LIỆU VÀ SINH CÂY KHÁI NIỆM PHÂN CẤP

Rời rạc hóa và sinh cây phân cấp khái niệm là phương pháp làm giảm số lượng dữ liệu, đồng thời cho phép người dùng phân tích dữ liệu ở các mức trừu tượng khác nhau.

3.7.1. Phương pháp áp dụng cho dữ liệu số

a) *Phương pháp binning*

Phương pháp này là phương pháp phân tách từ trên xuống dựa trên các nhóm số. Phương pháp này được trình bày kỹ trong phần làm mịn dữ liệu (phần 3.4) ở trên. Phương pháp này cũng được sử dụng để làm rời rạc hóa dữ liệu. Ví dụ như giá trị của thuộc tính có thể được rời rạc hóa bằng cách nhóm theo các giá trị bằng nhau về độ rộng hoặc bằng nhau về tần số, sau đó có thể thay thế cả nhóm giá trị đó bằng giá trị trung bình hoặc trung vị. Phương pháp này có thể được áp dụng lặp lại để có thể thu được sự rời rạc hóa kiểu phân cấp.

Phương pháp này không sử dụng thông tin về phân lớp nên được gọi là phương pháp rời rạc hóa không có giám sát. Phương pháp này khá nhạy cảm với số lượng nhóm tạo ra cũng như sự xuất hiện của giá trị ngoại lai.

b) *Phương pháp phân tích biểu đồ tần suất*

Cũng giống phương pháp Binning, phương pháp này cũng là phương pháp rời rạc hóa không có giám sát. Phương pháp này phân chia các giá trị của thuộc tính thành từng nhóm không giao nhau. Có rất nhiều cách phân chia khác nhau có thể được dùng để định nghĩa biểu đồ. Phân tích biểu đồ có thể được áp dụng lặp đi lặp lại với từng khối để có thể tự động tạo ra phân cấp đa mức, vòng lặp này sẽ dừng lại khi đạt tới mức được định nghĩa trước. Biểu đồ tần suất cũng có thể được phân rã dựa trên phân tích nhóm dựa trên phân bố của dữ liệu.

c) *Phương pháp phân cụm, cây quyết định và phân tích tương quan*

Phân cụm, cây quyết định và phân tích tương quan cũng thường được sử dụng để rời rạc hóa dữ liệu. Trong phần này chúng tôi chỉ giới thiệu ngắn gọn từng phương pháp.

Phân cụm là một phương pháp rời rạc hóa khá phổ biến. Các phương pháp phân cụm có thể được sử dụng để rời rạc hóa các thuộc tính số bằng cách phân nhỏ giá trị của thuộc tính A vào các cụm hoặc các nhóm. Phương pháp phân cụm cho phép phân tích phân bố của thuộc tính A , từ đó có thể trả lại các kết quả rời rạc hóa có chất lượng tốt.

Phân cụm có thể được dùng để tạo thành các mức phân cấp dựa theo cả hai chiến thuật từ phân rã trên xuống và gom nhóm từ dưới lên. Về mặt lý thuyết thì mỗi một nhóm là một nút trong cây phân cấp, với mỗi nút cha sẽ được phân rã thành một số cụm con để tạo thành mức thấp hơn. Ngược lại, các cụm được hình thành từ việc nhóm một vài cụm gần nhau tạo thành nút có mức cao hơn.

Cây quyết định dùng trong phân lớp cũng có thể được dùng để rời rạc hóa dữ liệu. Phương pháp này thực hiện theo cách tiếp cận phân rã trên xuống. Không giống các phương pháp đã giới thiệu ở trên, cây quyết định rời rạc hóa bằng cách tiếp cận học có giám sát dựa trên thông tin của thuộc tính phân lớp. Ví dụ với tập dữ liệu về các triệu chứng của bệnh với mỗi người sẽ có kết quả chẩn đoán tương ứng. Phân bố của kết quả phân lớp sẽ được sử dụng để tính toán và xác định các vị trí phân tách nút. Một cách trực quan, ý tưởng chính là lựa chọn các điểm phân chia các nút sao cho có thể phân chia thành nhiều cụm với tập bộ dữ liệu có cùng thuộc tính nhãn. Lý thuyết về độ đo Entropy thường được sử dụng cho mục đích này. Để rời rạc hóa một thuộc tính số A , phương pháp sẽ lựa chọn giá trị của A sao cho tối thiểu hóa độ đo Entropy tại điểm phân tách, công việc này được lặp lại cho tới khi đạt mức rời rạc phù hợp.

Tính toán độ tương quan cũng có thể sử dụng để rời rạc hóa dữ liệu. Phương pháp từ trước tới giờ đã được trình bày thường sử dụng hướng tiếp cận phân rã từ trên xuống. Ngược lại phương pháp này là phương pháp tổng hợp từ dưới lên bằng cách tìm các láng giềng gần nhau nhất và ghép chúng lại với nhau thành nhóm lớn hơn. Cũng giống như cách tiếp cận dựa trên cây

quyết định, phương pháp này cũng là phương pháp học có giám sát dựa trên thông tin phân lớp. Quan điểm cơ bản của hướng tiếp cận này là để có rời rạc hóa tốt, tần suất của các lớp liên quan khá phù hợp trong cùng một khoảng. Vì vậy nếu hai khoảng liền kề có phân bố phân lớp giống nhau thì có thể được ghép lại với nhau, nếu không thì chúng không thể ghép được với nhau. Phương pháp dựa trên khi bình phương (được trình bày ở mục 3.4 ở trên) này được thực hiện như sau. Đầu tiên, mỗi giá trị riêng biệt của thuộc tính số A sẽ được coi là một cụm độc lập. Hàm χ^2 được thực hiện với tất cả các nhóm liền kề nhau. Các nhóm với giá trị χ^2 nhỏ nhất sẽ được ghép với nhau (do giá trị χ^2 cho biết độ tương đồng giữa hai thuộc tính, giá trị càng nhỏ có nghĩa là phân bố càng giống nhau). Quá trình gom nhóm được lặp lại tới khi tiêu chuẩn dừng được định nghĩa trước.

3.8. PHƯƠNG PHÁP ÁP DỤNG CHO DỮ LIỆU PHÂN LOẠI

Dữ liệu phân loại (categorical data) (ví dụ như dữ liệu về giới tính, màu sắc, vị trí địa lý) là dữ liệu rời rạc. Đặc điểm của loại dữ liệu này là hữu hạn nhưng trong một số trường hợp là rất lớn, và đặc biệt là không có thứ tự. Việc xác định thứ tự cho loại dữ liệu này cần phải có chuyên gia định nghĩa. Ví dụ trong kho dữ liệu ta có các thuộc tính *ngõ*, *phố*, *tỉnh* (*thành phố*), *quốc gia*, khi đó ta có thể định nghĩa một cây phân cấp (hay thứ tự) giữa các thuộc tính này là: *ngõ* < *phố* < *tỉnh* < *quốc gia*. Khi nhóm dữ liệu ta cũng cần định nghĩa những giá trị nào thuộc vào nhóm nào, ví dụ như phố nào thuộc thành phố nào, hay những phố nào thuộc phía bắc Hà Nội, những phố nào thuộc phía nam Hà Nội,... Sau khi định nghĩa được các nhóm và cây phân cấp như trên thì ta có thể áp dụng các thuật toán xử lý tương tự như thuộc tính số ở trên.

3.9. TỔNG KẾT

Chuẩn bị dữ liệu là một công việc có vai trò quan trọng trong quá trình phân tích dữ liệu. Bản chất của công tác chuẩn bị dữ

liệu là xử lý thô dữ liệu theo mục đích khai phá cụ thể nào đó của người phân tích. Theo cách tiếp cận này, việc xử lý dữ liệu này sẽ có ảnh hưởng lớn tới kết quả phân tích. Chuẩn bị dữ liệu được phân chia thành một số nhóm như làm sạch dữ liệu, tích hợp dữ liệu, thu gọn dữ liệu và biến đổi dữ liệu... Tuy nhiên các phương pháp này thường được sử dụng kết hợp với nhau nhằm làm tăng hiệu quả của giai đoạn khai phá dữ liệu theo nhu cầu cụ thể của người phân tích dữ liệu. Trong mỗi nhóm lại có rất nhiều phương pháp cụ thể khác nhau mà ở đây chúng tôi chỉ giới thiệu một số phương pháp điển hình. Tùy thuộc vào đặc tính dữ liệu và mục tiêu bài toán, phân tích viên cần lựa chọn phương pháp chuẩn bị dữ liệu tương ứng và công việc này thường rất khó đánh giá định lượng một cách chính xác. Tuy nhiên việc lựa chọn phương pháp thích hợp lại giúp cho quá trình phân tích, khai phá dữ liệu trở nên dễ dàng và hiệu quả hơn rất nhiều.

CÂU HỎI VÀ BÀI TẬP

- 3.1.** Dữ liệu trong thực tế thường không đầy đủ, một số mẫu có thể bị thiếu một hoặc một vài giá trị. Trình bày một số phương pháp giải quyết vấn đề này?
- 3.2.** Giả sử dữ liệu của thuộc tính tuổi được gom lại theo nhóm như sau:

Tuổi	Tần số
1-5	200
6-15	450
16-20	300
21-50	1500
51-80	700
81-100	44

Tính giá trị trung vị (xấp xỉ) của tập dữ liệu trên.

3.3. Giả sử giá trị của thuộc tính tuổi là như sau: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 36, 40, 45, 46, 52, 70

- Tính giá trị trung bình và trung vị của tập dữ liệu trên
- Tính giá trị mode và kết luận tập dữ liệu này có đặc điểm gì (unimodal, bimodal, trimodal,...)
- Tính giá trị midrange của tập dữ liệu.
- Tính giá trị (xấp xỉ) Q_1 và Q_3 .
- Tính bộ 5 tóm tắt của tập dữ liệu trên.
- Vẽ sơ đồ boxplot

3.4. Sử dụng tập dữ liệu ở bài 3.

- Sử dụng phương pháp làm mịn bin theo tần suất là 3. Bình luận về kết quả thu được.
- Làm cách nào để phát hiện được trường hợp ngoại lai trong tập dữ liệu này.
- Sử dụng một phương pháp làm mịn khác cho tập dữ liệu này.

3.5. Cho biết dữ liệu về tuổi có giá trị như sau: 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70, thực hiện một số yêu cầu sau

- Với kỹ thuật min-max, cho biết giá trị của tuổi 35 trong khoảng [0, 1].
- Dùng kỹ thuật z-score để chuyển giá trị 35 với độ lệch chuẩn là 12,94

3.6. Giả sử dữ liệu kiểm tra sự liên quan giữa tuổi và sự béo phì của bệnh viện trên 18 người chọn ngẫu nhiên:

age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2

age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

- 3.7.** Tính giá trị trung bình, trung vị, và độ lệch chuẩn của hai thuộc tính tuổi (age) và tỉ lệ béo (%fat) cho tập dữ liệu trên.
- Vẽ biểu đồ boxplot cho 2 thuộc tính trên.
 - Vẽ biểu đồ scatter plot và q-q plot cho 2 thuộc tính trên
 - Chuẩn hóa 2 thuộc tính trên bằng z-score.
 - Tính hệ số tương quan giữa 2 thuộc tính trên. Kết luận xem 2 thuộc tính này có quan hệ gì với nhau hay không?
- 3.8.** Sử dụng lược đồ để tóm tắt các kỹ thuật trích chọn thuộc tính
- Mở rộng dần tập con (forward selection).
 - Loại bỏ dần các thuộc tính (backward elimination).
 - Kết hợp hai phương pháp trên.

Chương 4.

PHÁT HIỆN LUẬT KẾT HỢP

4.1. GIỚI THIỆU VỀ LUẬT KẾT HỢP

Mẫu phổ biến (frequent patterns) là các mẫu (ví dụ: tập các mục, chuỗi con hoặc các thành phần cấu trúc con) mà xuất hiện một cách thường xuyên trong một tập dữ liệu. Ví dụ như: một tập các mục (như bánh mì và sữa), thường được mua cùng nhau trong các hóa đơn hàng siêu thị, thì ta gọi là *tập mục phổ biến (frequent Itemset)*. Mỗi chuỗi con thường xuất hiện lần lượt trong cơ sở dữ liệu thì được coi là một *mẫu tuần tự (sequential pattern)* ví dụ như khách hàng thường mua laptop trước sau đó có thể mua máy ảnh số rồi đến thẻ nhớ. Một thành phần cấu trúc con như đồ thị con, cây con, mạng con... xuất hiện thường xuyên thì lại được gọi là *mẫu phổ biến có cấu trúc (structured pattern)*. Trong bài toán khai phá luật kết hợp, chúng ta thường quan tâm đến các tập mục phổ biến nhiều hơn.

Khai phá luật kết hợp là tìm ra các mẫu có tần suất cao, các mẫu kết hợp, liên quan hoặc các cấu trúc tồn tại giữa các tập hợp đối tượng trong cơ sở dữ liệu các giao dịch, cơ sở dữ liệu quan hệ hoặc các kho chứa thông tin khác. Nói cách khác là chúng ta đi tìm tất cả các tập phổ biến từ trong dữ liệu.

Cho một tập các giao tác, khai phá luật kết hợp có nhiệm vụ tìm ra các luật mà dự đoán sự xuất hiện của một đối tượng dựa vào sự xuất hiện của các đối tượng khác trong giao tác. Nhưng nó không có khả năng khai phá ra các chuỗi đối tượng xảy ra tuần tự đảm bảo một điều kiện nào đó. Điều này sẽ được xử lý trong bài toán khai phá các mẫu tuần tự.

Một trong những ví dụ điển hình cho bài toán khai phá luật kết hợp là bài toán sắp xếp hàng hóa trong siêu thị. Giả sử bạn là một chủ cửa hàng. Để đưa ra chiến lược kinh doanh hiệu quả, bạn muốn quan tâm đến thói quen mua sắm của khách hàng. Một trong các câu hỏi đặt ra là “Nhóm những mặt hàng nào mà khách hàng thường mua cùng trong một lần ghé cửa hàng?” Sau khi xử lý trên khối dữ liệu hóa đơn từ xưa đến nay thì nhận ra rằng: có 30% hóa đơn có tính tiền cả bia và tã lót trẻ em, và cứ 100 người mua tã lót thì có đến 40 người mua thêm bia. Nó gợi ý cho bạn rằng nên để gian hàng bia và tã lót gần nhau để tiện cho khách hàng. Trong ví dụ này, tập {bia, tã lót} là một tập phổ biến với tần suất 30%, luật {40% người mua tã lót thì mua luôn cả bia} là một luật kết hợp.

Bài toán đặt ra như sau:

Cho biết $T = \{t_1, t_2, \dots, t_n\}$ là tập các *giao dịch* (transaction) với n là số các giao dịch có trong T . Tập $I = \{i_1, i_2, \dots, i_m\}$ là một tập gồm m *tập mục* khác nhau xuất hiện trong t_i . Mỗi giao dịch t_i là một tập các mục xuất hiện đồng thời. Ta có $t_i \subset I$. Với X và Y là các tập mục. **Luật kết hợp** có thể biểu diễn bởi công thức sau:

$$X \rightarrow Y, \text{ với } X \subset I, Y \subset I \text{ và } X \cap Y = \emptyset$$

Một giao dịch t_i thuộc T chứa một tập mục X nếu X là tập con của t_i .

Xét cơ sở dữ liệu bao gồm 8 giao dịch ($n = 8$) và có 5 mục dữ liệu khác nhau được gán giá trị lần lượt là a, b, c, d, e , vậy ta có $m = 5$ và $I = \{a, b, c, d, e\}$. Dữ liệu trong CSDL được phân bổ như trong bảng 4.1

Bảng 4.1. Cơ sở dữ liệu ví dụ gồm 5 giao dịch

TID	Tập mục trong giao dịch
1	{a, b, c, d, e}
2	{b, c}
3	{a, b, f}
4	{a, b, g}
5	{a, f, h}

Độ hỗ trợ của X trong T là số giao dịch chứa X trong T (viết tắt là $X.count$). Ví dụ: $a.count = 4$, $b.count = 4$

Hai khái niệm hết sức cơ bản để đo độ mạnh của một luật kết hợp là *độ hỗ trợ* và *độ tin cậy*.

Độ hỗ trợ của một luật $X \rightarrow Y$ là tỉ lệ % các giao dịch trong T mà chứa cả X và Y. Nó giúp xác định mức độ phổ biến của các giao dịch chứa tập mục $(X \cup Y)$ trong tổng số tất cả các giao dịch. Công thức tính *độ hỗ trợ (support)*:

$$support(X \rightarrow Y) = \frac{(X \cup Y).count}{n} = P(X \cup Y) \quad (4.1)$$

Độ tin cậy của luật $X \rightarrow Y$ lại là tỉ lệ % các giao dịch trong T chứa cả X và Y trên tổng số các giao dịch trong T chỉ chứa X. Nó là đại lượng xác định khả năng dự đoán của luật và được tính như sau:

$$confidence(X \rightarrow Y) = \frac{(X \cup Y).count}{X.count} = P(Y | X) \quad (4.2)$$

Bài toán cơ bản đặt ra: Cho một tập các giao dịch T, tìm ra tất cả các luật kết hợp trong T mà có độ hỗ trợ không nhỏ hơn một ngưỡng nào đó (*minsup*) và đồng thời cũng có độ tin cậy không nhỏ hơn một ngưỡng khác (*minconf*). Luật được sinh ra thỏa mãn không nhỏ hơn hai ngưỡng *minsup* và *minconf* được gọi là *luật mạnh*. Nhìn chung, bài toán khai phá luật kết hợp thường được chia làm 2 pha chính:

Pha 1 (Tìm tất cả các tập mục phổ biến): Mỗi tập mục sẽ được tính xác suất xuất hiện, các tập mục phổ biến phải thỏa mãn độ hỗ trợ không nhỏ hơn độ hỗ trợ tối thiểu *minsup*.

Pha 2 (Sinh ra các luật kết hợp mạnh từ các tập mục phổ biến ở pha 1): Các luật này phải có độ tin cậy không nhỏ hơn độ tin cậy nhỏ nhất *minconf*.

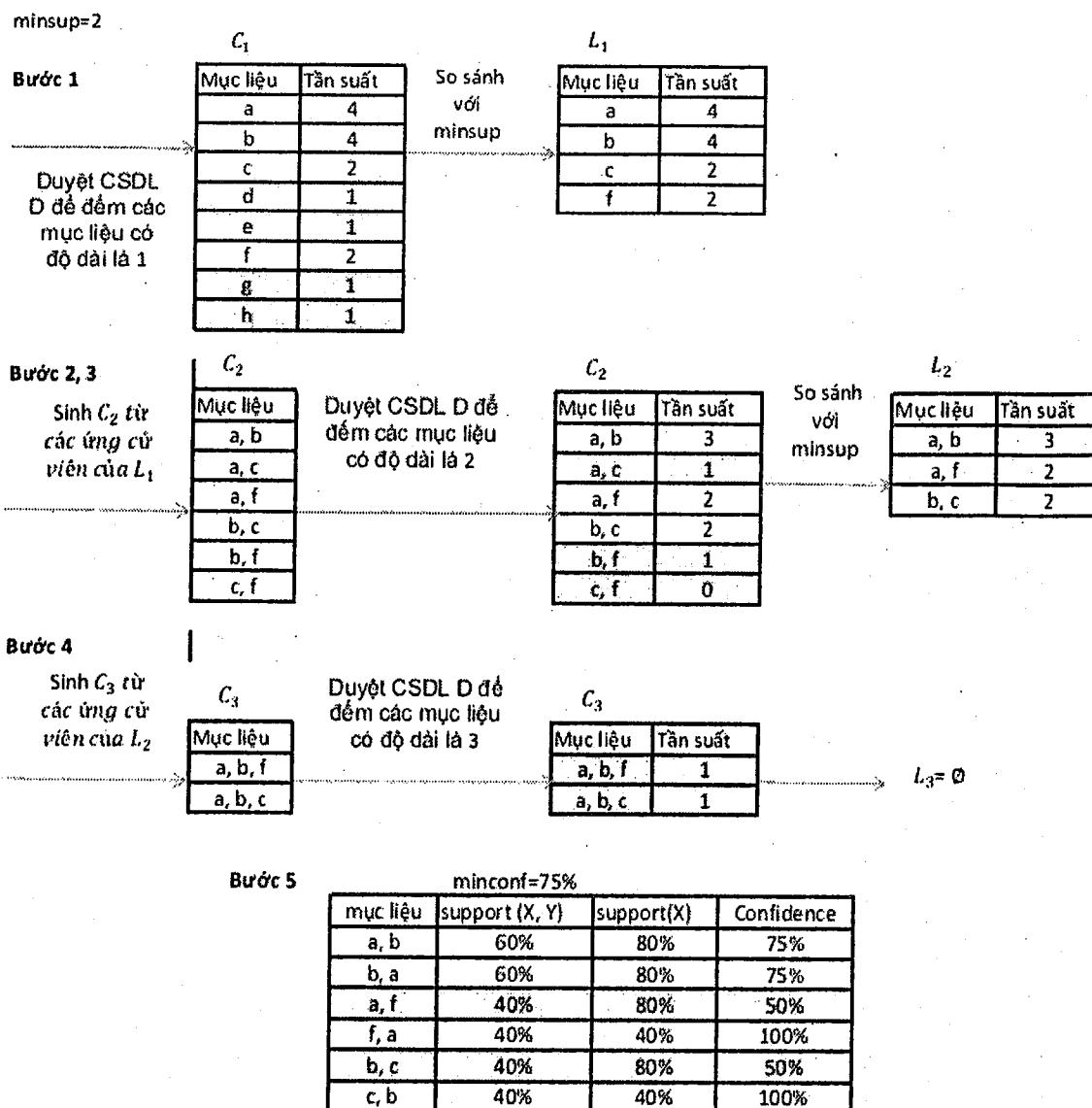
Khác với các phương pháp khai phá dữ liệu khác, dựa vào các ngưỡng tối thiểu, khai phá luật kết hợp luôn chỉ có duy nhất một tập kết quả cho dù áp dụng bất kì một giải thuật nào. Thách thức

lớn nhất của khai phá tập mục phổ biến là nó thường sinh ra một lượng vô cùng lớn các tập mục thỏa mãn ngưỡng $minsup$, đặc biệt khi $minsup$ khá nhỏ. Điều này do nếu một tập mục là thường xuyên thì mỗi tập con của nó cũng thường xuyên xảy ra. Ví dụ: ta có một tập mục thường xuyên có độ dài 10, sẽ chứa 10 tập mục thường xuyên có độ dài 1, đồng thời chứa $10!/(10-2)!*2! = 45$ tập mục thường xuyên có độ dài 2, ... Số lượng các tập thường xuyên như vậy là quá lớn để tính toán nên người ta đưa ra một số khái niệm để giải quyết vấn đề này bao gồm *tập thường xuyên đóng* và *tập thường xuyên cực đại*.

Một tập mục X được gọi là *tập đóng* trong cơ sở dữ liệu T nếu không tồn tại bất kỳ một tập Y sao cho $X \subset Y$ mà có cùng độ hỗ trợ trong T . Một tập mục được gọi là *tập thường xuyên đóng* nếu nó là tập vừa đóng thường xuyên trong cơ sở dữ liệu T . Một tập X được gọi là *tập thường xuyên cực đại* nếu X là tập thường xuyên và không tồn tại tập X mà $X \subset Y$ đồng thời Y là thường xuyên trong T .

4.2. PHƯƠNG PHÁP KHAI PHÁ TẬP MỤC PHỔ BIẾN

Apriori là một thuật giải được R. Agrawal, R. Srikant đề xuất lần đầu vào năm 1994 nhằm khai phá tập mục phổ biến nhị phân. Thuật toán này thực hiện lặp lại việc tìm kiếm theo mức, sử dụng thông tin ở mức k để duyệt mức $k+1$. Đầu tiên, tập các mục thường xuyên có độ dài là 1 được xây dựng bằng việc duyệt qua toàn bộ dữ liệu để đếm sự xuất hiện của từng phần tử và giá trị này phải lớn hơn hoặc bằng độ hỗ trợ nhỏ nhất ($minsup$). Kết quả của việc đếm này được ký hiệu là L_1 . Tiếp theo L_1 này được sử dụng để tìm L_2 là tập mục thường xuyên có độ dài 2. Tác vụ này được thực hiện lặp lại đến khi không tìm được tập mục thường xuyên có độ dài k thỏa mãn điều kiện $minsup$. Lưu ý rằng mỗi lần thực hiện việc tìm tập các mục thường xuyên L_k yêu cầu duyệt toàn bộ dữ liệu. Từ tập mục thường xuyên này ta sinh ra luật kết hợp mạnh bằng cách tìm các luật trong tập mục thường xuyên thỏa mãn ngưỡng $minconf$.



Apriori Algorithm

1. Duyệt toàn bộ CSDL giao dịch để tính giá trị hỗ trợ là phần tử của tập phổ biến tiềm năng C_1 của 1-itemset, so sánh với $minsup$, để có được 1-itemset (L_1)
2. L_1 nối (phép join) L_1 để sinh ra 2-itemset là tập phổ biến tiềm năng. Loại bỏ các tập mục không phải là tập phổ biến thu được 2-itemset C_2
3. Duyệt toàn bộ CSDL giao dịch để tính giá trị hỗ trợ của mỗi ứng viên 2-itemset, so sánh từng phần tử với $minsup$ để thu được tập mục thường xuyên 2-itemset (L_2)

4. Lặp lại từ bước 2 cho đến khi tập ứng cử tiềm năng $C = \emptyset$ (không tìm thấy tập mục phổ biến)

5. Với mỗi mục phổ biến I , sinh tất cả các tập con s không rỗng của I

6. Với mỗi tập con s không rỗng của I , sinh ra các luật $s \Rightarrow (I-s)$ nếu độ tin cậy (Confidence) của nó $\geq minconf$

Ví dụ:

Xét CSDL trong Bảng 4.1, tìm tất cả các luật kết hợp áp dụng thuật toán trên thỏa mãn điều kiện $minsup = 40\%$ và $minconf = 80\%$

Tập luật sinh ra sau khi thực hiện thuật toán có thể như sau

R1: $a \rightarrow b$ (support = 60%, confidence = 75%)

R2: $b \rightarrow a$ (support = 60%, confidence = 75%)

R3: $f \rightarrow a$ (support = 40%, confidence = 100%)

R4: $c \rightarrow b$ (support = 40%, confidence = 100%)

4.3. THUẬT TOÁN FP-GROWTH

4.3.1. Ý tưởng thuật toán

Thuật toán kinh điển Apriori tìm tập mục phổ biến thực hiện khá hiệu quả tốt bởi rút gọn kích thước các tập ứng cử nhờ kỹ thuật tinh nhánh như giới thiệu ở phần trước. Tuy nhiên, trong tình huống mà số các dữ liệu nhiều, độ dài của giao dịch dài hoặc độ hỗ trợ cực tiểu thấp, các thuật toán Apriori gặp phải 2 chi phí lớn:

- Chi phí cho số lượng khổng lồ các tập ứng cử. Ví dụ: nếu có 10^4 tập 1-mục phổ biến thì thuật toán Apriori sẽ cần sinh ra hơn 10^7 các ứng cử 2-itemset và thực hiện kiểm tra sự xuất hiện của chúng. Hơn nữa, để khám phá được một số mẫu phổ biến kích thước (độ dài) là l , thuật toán phải kiểm tra $(2^l - 2)$ các mẫu phổ biến tiềm năng. Ví dụ $l=100$, chẳng hạn là $\{a_1, a_2, \dots, a_{100}\}$, nó phải sinh ra tổng số $2^{100} \approx 10^{30}$ các ứng cử (đây chính là số tập con của tập có 100 phần tử).

- Đòi hỏi lặp lại nhiều lần duyệt CSDL để kiểm tra tập rất lớn các ứng cử. Số lần duyệt CSDL của thuật toán Apriori bằng độ dài của mẫu phổ biến dài nhất tìm được. Trong trường hợp mẫu phổ biến dài hơn và CSDL lớn, có nhiều bản ghi, điều này là không thể thực hiện được. Thuật toán Apriori chỉ thích hợp cho các CSDL thừa (sparse), với các CSDL có mật độ dày (dense) thì thuật toán thực hiện kém hiệu quả hơn.

Nhằm khắc phục các nhược điểm trên, thuật toán có tên là FP-growth được giới thiệu bởi Jiawei Hai Jian Pei và Yiwen Yin năm 2000. Thuật toán tìm các tập phổ biến hiệu quả hơn thuật toán Apriori bằng việc sử dụng một kỹ thuật khác không cần sinh các ứng cử. Sự hiệu quả của khai phá nhận được với 3 kỹ thuật chính:

Thứ nhất nó mở rộng của cấu trúc cây prefix (prefix tree), được gọi là cây mẫu phổ biến (*frequent pattern tree hoặc gọi tắt là FP-tree*) dùng để nén dữ liệu thích hợp. Chỉ có các mục độ dài l (l -itemset) ở trong cây và các nút của cây được sắp đặt để các nút xuất hiện thường xuyên hơn có thể dễ dàng chia sẻ với các nút xuất hiện ít hơn. CSDL lớn được nén chặt tới cấu trúc dữ liệu nhỏ hơn (FP-tree), tránh được chi phí lặp lại duyệt qua CSDL.

Thứ hai, phương pháp khai phá tăng trưởng (growth) từng đoạn dựa trên Fp-tree gọi là phương pháp *FP – growth* đã được thực hiện. Bắt đầu từ mẫu phổ biến độ dài 1, FP-growth chỉ xem xét cơ sở mẫu phụ thuộc của nó (condition pattern base) như là CSDL con (sub-database) bao gồm tập các mục phổ biến cùng xuất hiện với mẫu hậu tố (suffix pattern), xây dựng condition FP-tree tương ứng của nó và thực hiện khai phá đệ qui trên cây này. Mẫu tăng trưởng là nhận được qua việc nối mẫu hậu tố (suffix pattern) với một đoạn mẫu được sinh ra từ condition FP-tree. Khai phá dựa trên FP-tree được thực hiện theo cách tăng trưởng (growth) các đoạn mẫu để tránh chi phí cho việc sinh ra số lượng lớn các tập ứng cử.

Thứ ba, kỹ thuật tìm kiếm được dùng ở đây là dựa vào kỹ thuật chia để trị (*divide-and-conquer method*) để phân rã nhiệm vụ khai phá thành tập các nhiệm vụ nhỏ hơn với giới hạn các mẫu trong các CSDL nhằm thu gọn không gian tìm kiếm.

Phương pháp FP-growth đã chứng tỏ được tính hiệu quả của nó và thể hiện khai phá cho cả các mẫu ngắn và dài, nhanh hơn thuật toán Apriori, luôn chỉ cần duyệt CSDL hai lần

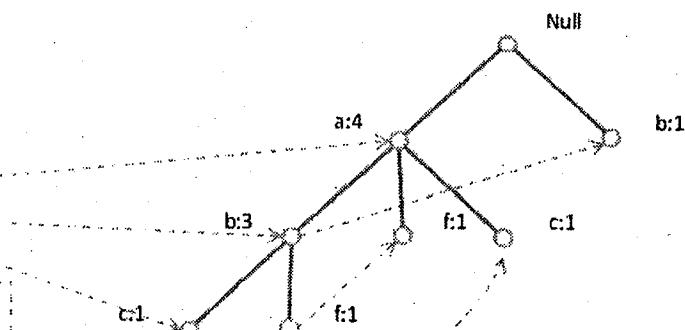
4.3.2. Thuật toán FP-growth

1. Duyệt CSDL lần thứ nhất để tính độ hỗ trợ của tất cả 1-itemset. Loại bỏ những mục có độ hỗ trợ nhỏ hơn $minsup$. Các mục còn lại được sắp theo thứ tự giảm dần của độ hỗ trợ (cũng tức là giảm dần theo số lần xuất hiện trong CSDL), ta nhận được danh sách L các mục đã sắp.
2. Duyệt CSDL lần thứ hai, với mỗi tác vụ t , loại các mục không đủ độ hỗ trợ, các mục còn lại theo thứ tự giống như xuất hiện trong L (tức là thứ tự giảm dần theo độ hỗ trợ) được đưa vào cây FP-tree.
3. Tìm các tập mục phổ biến trên cây FP-tree đã xây dựng mà không cần duyệt lại CSDL nữa.

Định nghĩa 4.1: Cấu trúc cây FP-tree được định nghĩa như sau:

- Gốc của cây nhãn *null*, các đường đi trên cây biểu diễn một tập các tiền tố của một tập mục
- Mỗi nút trong cây có chứa ba thành phần: tên mục, số lần xuất hiện (*count*), con trỏ. Trong đó, *count* là số lượng xuất hiện của nhánh con (từ *NULL* đến nút này) trong các giao dịch, còn con trỏ liên kết (mũi tên nét đứt) đến nút có cùng tên tiếp theo của nó.
- Mỗi dòng trong bảng header chứa hai trường: tên mục và nút rỗng trỏ tới đến nút đầu tiên cùng một mục trên cây FP

Mục liệu	Tần suất	Node link
a	4	
b	4	
c	2	
f	2	
d	1	
e	1	
g	1	
h	1	



Hình 4.2. Ví dụ về cây FP (xây dựng từ dữ liệu ở bảng 4.1)

Ta xây dựng hàm $insert_tree((p, P), T)$ với T là gốc của một nhánh con ta đang duyệt đến. Nếu T có một nút con là N thỏa mãn $N.\text{tên} = p.\text{tên}$ thì $N.\text{count}$ tăng lên 1. Ngược lại, ta tạo một nút con mới Q với $Q.\text{tên} = P.\text{tên}$ với $Q.\text{count} = 1$. Gọi tiếp hàm đệ quy $insert_tree$ cho tập con P và N hoặc Q ứng với từng trường hợp. Thủ tục thêm một dãy các mục (đã sắp giảm dần theo độ hỗ trợ) của một tác vụ vào cây thực hiện đệ quy như sau:

```

Procedure insert_tree(string[ p|P] ,tree có gốc T)
If T có nút con N mà N.itemname=p
Then N.count++
else
    Tạo một nút mới N;
    N.itemname:=p;
    N.count:=1
    Thay đổi nút liên kết cho p bao gồm N;
End if
If p # rỗng
    insert_tree(P,N);
  
```

Ví dụ:

Xây dựng lại cây FP tuân tự từng bước một

Tìm tập mục phổ biến trên cây FP-tree:

Sau khi xây dựng xong FP-tree cho CSDL, việc khai phá tìm các mẫu phổ biến chỉ thực hiện trên cây FP-tree mà không cần

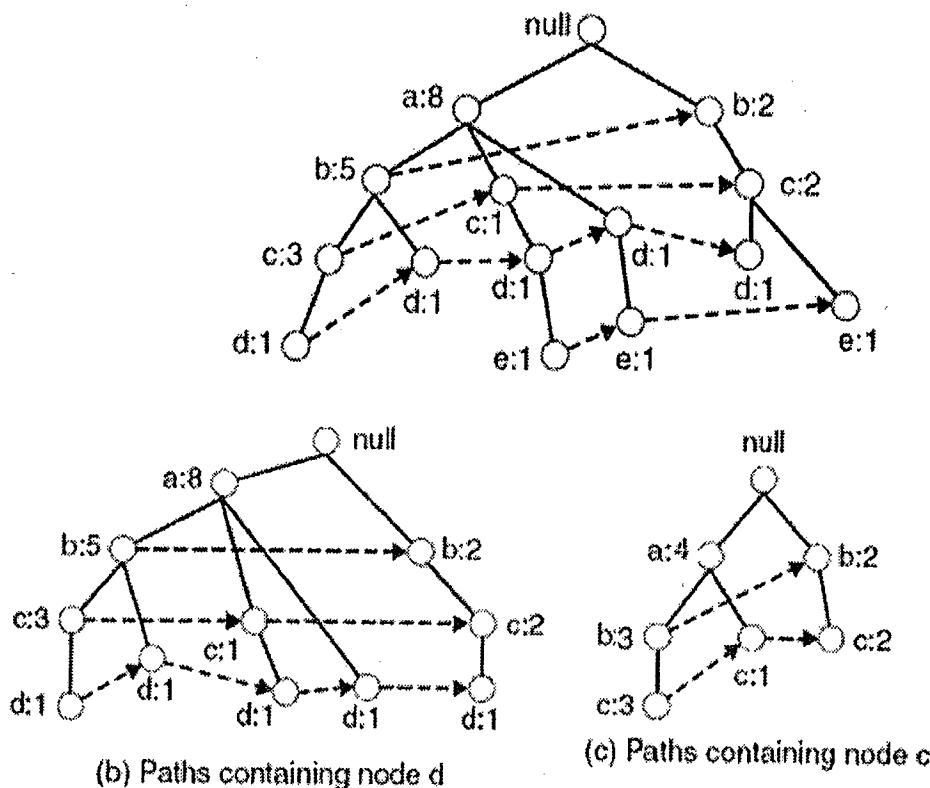
duyệt CSDL nữa. Kiến trúc của cây FP đảm bảo một kiến trúc dữ liệu khả bền vững. Tuy nhiên nó không mặc định đảm bảo chắc chắn rằng nó có độ hiệu quả cao hơn nhiều bởi vì nếu vẫn đơn giản sử dụng cây FP để sinh và kiểm tra tất cả các mẫu ứng viên thì chúng ta vẫn phải đối mặt với một lượng tổ hợp lớn các ứng viên được sinh ra.

Một giải thuật chia-để-trị được xây dựng giúp giải quyết điểm hạn chế trên. Trước hết, cần làm rõ một số bổ đề, tính chất hỗ trợ cho giải thuật.

Tính chất 1 (Tính chất liên kết nút)

Với bất cứ mục phổ biến a_i nào, tất cả các mẫu phổ biến có phần tử cuối cùng là a_i đều có thể được tìm ra thông qua các liên kết nút của a_i , bắt đầu từ nút đầu trên bảng tiêu đề các mục.

Thật vậy, nếu ta loại bỏ các nhánh không chứa d và các nút con đằng sau d như trong Hình 4.3(a) dựa vào đường liên kết nút. Tất cả thông tin về các mẫu phổ biến có phần tử cuối là d đều chứa trong đồ thị bên phải.



Hình 4.3. Mô tả tính chất liên kết nút

Ta có các mẫu điều kiện cơ sở (*conditional pattern base*) cho nút d là $\{(a:8, b:5, c:3), (a:8, b:5), (a:8, c:1), (a:8), (b:2, c:2, d:1)\}$. Đó là các đường kéo từ nút gốc đến d. Qua đó ta tìm ra được cây điều kiện gọi là *cây FP điều kiện* (*conditional FP-tree*) bao gồm các nút trên Hình 4.3(b) mà có tổng số lần xuất hiện trên các mẫu điều kiện cơ sở của a_i lớn hơn độ hỗ trợ tối thiểu. Ví dụ trong Hình 4.3(c), các biến count đã được điều chỉnh ứng với $c.count$, nếu $minsup = 4$ thì ta có cây FP điều kiện là $(a:4, b:5) | c$, còn nếu $minsup = 5$ thì ta có cây FP điều kiện là $\{(b:5)\} | c$

Tính chất 2 (Tính chất Đường tiền tố - Prefix path)

Để tính các mẫu phổ biến cho nút a_i trên đường đơn P , chỉ cần quan tâm đến các nút đi trước nút a_i trên đường P , và các nút đó có cùng giá trị count với nút a_i nếu coi như cây chỉ bao gồm duy nhất đường P .

Thật vậy, với nhánh đầu tiên trong Hình 4.3(c), ta có đường $\{(a:3, b:3)\}$ cho nút $(c:3)$. Đường này gọi là transform prefixed path của a_i trên P .

Bổ đề Fragment Growth

Cho α là một tập mục trong dữ liệu giao dịch D , B là mẫu điều kiện của α , và β là một tập mục trong B . Khi đó, độ hỗ trợ của $(\alpha \cup \beta)$ trong D tương đương với độ hỗ trợ của β trong B .

Thật vậy, trong Hình 4.3 (b)(c) thì nút a_i ở đây là c hoặc d đều nằm ở nút lá và $\sum a_i.count$ là lớn nhất. Do đó độ hỗ trợ của $(\alpha \cup \beta)$ bằng độ hỗ trợ của β . Ngoài ra, ta rút ra được nhận xét là để $(\alpha \cup \beta)$ là tập mục phổ biến khi và chỉ khi β cũng là tập mục phổ biến.

Tính chất 3 (Sinh ra mẫu trên đường FP-tree đơn)

Giả sử một cây FP là T (như Hình 4.3 (a)) có một đường đơn P . Tập tất cả các mẫu phổ biến của T có thể tìm được bằng cách tổ hợp các nút trên P mà có độ hỗ trợ không nhỏ hơn $minsup$

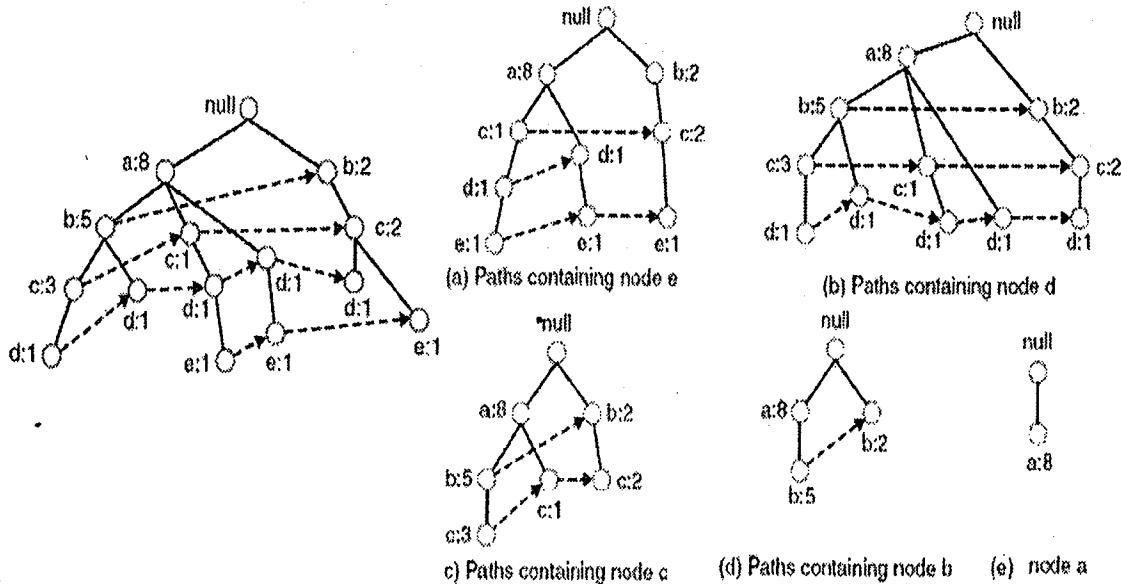
Ví dụ, giả sử ta có cây FP điều kiện là $((a:4, b:5) | c$ thì ta có thể kiểm tra các tổ hợp $\{ac, bc, abc\}$.

Procedure FP-Growth(Tree, α) { // α là 1 itemset

- (1) If Tree chứa một đường đơn P
- (2) then for each tổ hợp của các nút trên P do
- (3) sinh ra mẫu $\alpha \cup \beta$ có độ hỗ trợ bằng độ hỗ trợ nhỏ nhất của các nút trong β ;
- (4) End for;
- (5) else for each a_i trên hàng đầu tiên của Tree do
- (6) sinh ra mẫu $\beta = a_i \cup \alpha$ với độ hỗ trợ = độ hỗ trợ của a_i ;
- (7) xây dựng mẫu điều kiện của β ;
- (8) xây dựng cây FP điều kiện $Tree_\beta$;
- (9) if $Tree_\beta = \emptyset$
- (10) then gọi $FP-growth(Tree_\beta, \beta)$
- (11) end for;
- (12) end if;

Ví dụ minh họa:

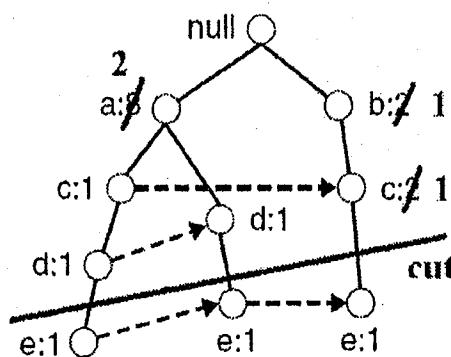
Đối với mỗi item ta tạo ra các cây con đường cha (như Hình 4.4) dựa vào các đường *mẫu điều kiện cơ bản* của nó.



Hình 4.4. Đường prefix path cho mỗi mục

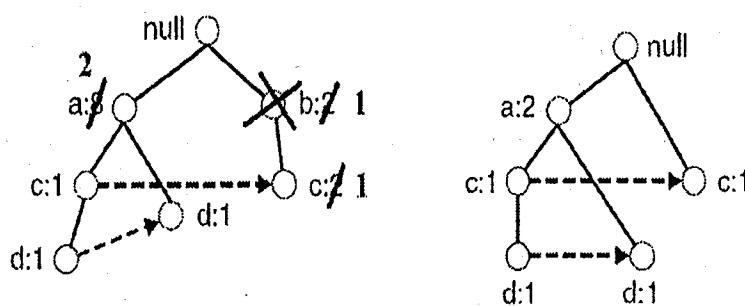
Đối với mỗi cây con (ứng với một item) ta sinh ra cây FP điều kiện (ví dụ mẫu với cây cả e trong Hình 4.4(a)) bằng cách cập nhật

lại các con đếm *count*. Lúc này chỉ có 2 đường qua *a*, như vậy *a.count* = 2. Tương tự ta có *b.count* = 1 và *c.count* = 1. Loại bỏ các nút *e* như Hình 4.5, ta thực hiện tiếp tục tìm các tập phổ biến mức 3 chứa *de* ở cuối như Hình 4.7 (làm tương tự với *ae*, *ce*). Trong cây FP điều kiện của *e*, nhánh bên phải NULL không chứa lá *d*, do vậy nhánh này bị cắt đi khi xem xét đến các tập phổ biến có chứa *de*. Sau khi cắt ta được hình 4.7(b) ở giữa chứa các đường tiền tố của *de*.

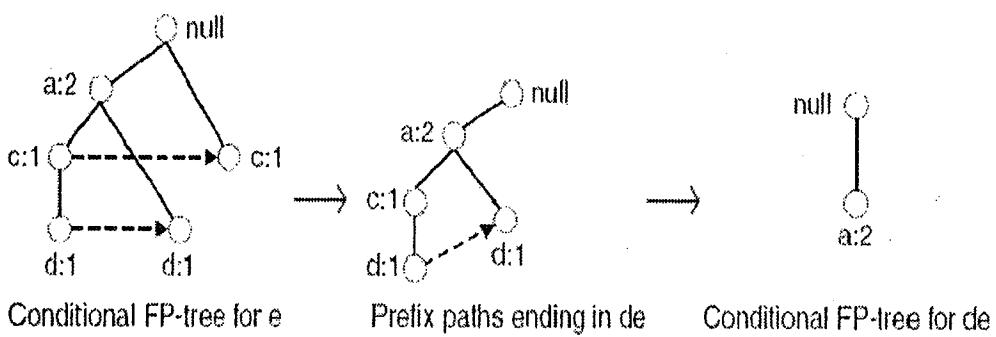


Hình 4.5. Cây sau khi loại bỏ nút *e*

Tiếp tục xem xét đến độ hỗ trợ của các mục trong cây 4.7b, *c.count*=1 nhỏ hơn độ hỗ trợ tối thiểu nên mục *c* bị cắt đi. Ta còn lại hình 4.7c chứa cây FP điều kiện cho *de* và nó chỉ có duy nhất đỉnh *a* (không tính gốc). Vậy ta có thêm tập mục phổ biến $\{a, d, e\}$. Loại bỏ nút có độ hỗ trợ nhỏ hơn độ hỗ trợ tối thiểu trong cây (Hình 4.6). Ở đây, độ hỗ trợ của *B* là 1, trong khi *minsup* = 2. Từ cây này ta tìm được các tập phổ biến mức 2 chứa *e* ở cuối: *ae*, *de*, *ce*



Hình 4.6. Cây sau khi loại bỏ nút *e* và tính toán lại độ hỗ trợ



Hình 4.7. Tập phổ biến mức 3

4.4. MỘT SỐ THUẬT TOÁN SONG SONG

Một số thuật toán song song đã được đề xuất và thử nghiệm. Các thuật toán này được thiết kế trên hệ máy tính song song không chia sẻ (shared-nothing architecture) có tính chất như sau:

Hệ có N bộ xử lý (BXL - processor), mỗi BXL P^i này có bộ nhớ trong (RAM) và bộ nhớ ngoài (thường là ổ đĩa) độc lập với các BXL còn lại trong hệ thống.

N BXL này có thể truyền thông với nhau nhờ một mạng tốc độ cao sử dụng cơ chế truyền thông điệp (message passing).

4.4.1. Thuật toán phân phối độ hỗ trợ

Thuật toán song song phân phối độ hỗ trợ dựa trên nền thuật toán Apriori [AS94]. Trong thuật toán này, N là số BXL, P^i là BXL thứ i , D^i là phần dữ liệu được gắn với BXL P^i (CSDL D ban đầu được chia ra làm N phần, mỗi phần gắn với một BXL). Thuật toán bao gồm các bước sau:

Bước 1: với $k = 1$, tất cả N BXL đều nhận được L_k là tập tất cả các tập thuộc tính phổ biến có lực lượng bằng 1.

Bước 2: với mọi $k > 1$, thuật toán thực hiện lặp đi lặp lại các bước sau:

Mỗi BXL P^i tạo ra tập các tập thuộc tính ứng cử viên C_k bằng cách kết nối các tập thuộc tính phổ biến trong L_{k-1} . Nhớ

rằng, tất cả các BXL đều có thông tin về L_{k-1} giống hệt nhau nên chúng sinh ra C_k cũng giống hệt nhau.

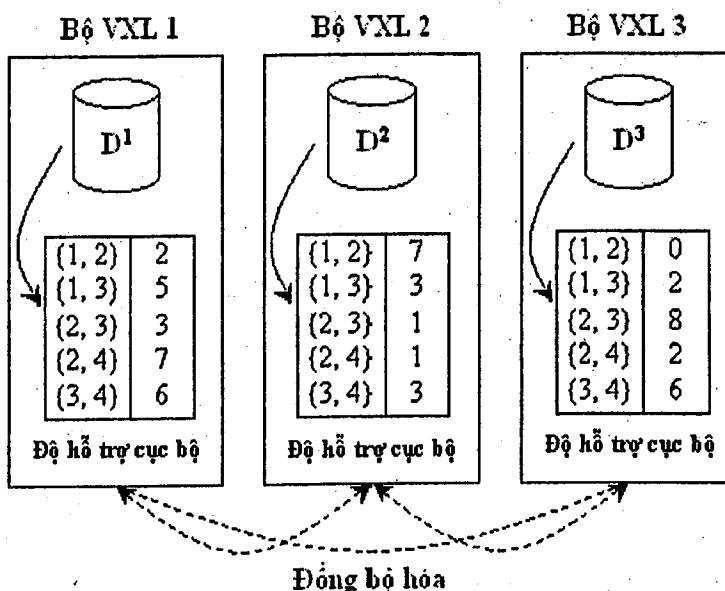
Mỗi BXL P^i duyệt qua CSDL D^i của riêng nó để cập nhật độ hỗ trợ cục bộ cho các tập thuộc tính ứng cử viên trong C_k . Đây chính là quá trình các BXL thực hiện song song với nhau.

Sau khi đã cập nhật xong độ hỗ trợ cục bộ cho các tập thuộc tính ứng cử viên trong C_k , các BXL tiến hành truyền thông tin cho nhau để thu được độ hỗ trợ toàn cục. Ở bước này, các BXL bắt buộc phải đồng bộ hóa với nhau.

Các BXL căn cứ vào độ hỗ trợ tối thiểu **minsup** để chọn ra tập những tập thuộc tính phổ biến L_k từ tập các ứng cử viên C_k .

Mỗi BXL có quyền kết thúc tại bước này hoặc tiếp tục thực hiện lặp lại bước 2.1.

Hình 4.8 minh họa nguyên lý làm việc của thuật toán này.



Hình 4.8. Thuật toán phân phối độ hỗ trợ trên 3 BXL

4.4.2. Thuật toán phân phối dữ liệu

Ưu điểm nổi bật của thuật toán phân phối độ hỗ trợ là không cần truyền dữ liệu giữa các BXL trong quá trình tính toán. Do đó, chúng có thể hoạt động độc lập và không đồng bộ với nhau trong

khi duyệt dữ liệu trên bộ nhớ hoặc ổ đĩa cục bộ. Tuy nhiên, nhược điểm của thuật toán này là không khai thác hết sức mạnh tổng hợp của N bộ nhớ ứng với N BXL của toàn hệ thống. Giả sử mỗi BXL có dung lượng bộ nhớ cục bộ là $|M|$ thì số tập thuộc tính ứng cử viên được cập nhật độ hỗ trợ trong mỗi pha bị giới hạn bởi hằng số m phụ thuộc $|M|$. Khi số BXL trong hệ thống tăng từ 1 đến N , hệ thống sẽ có một bộ nhớ tổng hợp với dung lượng $N \times |M|$, nhưng với thuật toán phân phối độ hỗ trợ ở trên, chúng ta cũng chỉ đếm được m tập thuộc tính ứng cử viên do tính chất của thuật toán là tất cả các BXL đều có tập C_k giống hệt nhau.

Thuật toán phân phối dữ liệu (data distribution) được thiết kế với mục đích tận dụng được sức mạnh tổng hợp của bộ nhớ hệ thống khi số BXL tăng lên. Trong thuật toán này, mỗi BXL tiến hành cập nhật độ hỗ trợ cho một số các tập thuộc tính ứng cử viên của riêng nó. Do đó, khi số BXL trong hệ thống tăng lên, thuật toán này có thể cập nhật độ hỗ trợ cho rất nhiều các tập thuộc tính ứng cử viên trong một pha. Nhược điểm của thuật toán này là mỗi BXL phải truyền và nhận dữ liệu ở mỗi pha nên nó chỉ khả thi khi hệ thống có một môi trường truyền thông nhanh và ổn định giữa các nút trong hệ thống. Thuật toán song song phân phối dữ liệu (data distribution) cũng dựa trên nền thuật toán Apriori [AS94]. Trong thuật toán này, N là số BXL, P^i là BXL thứ i , D^i là phần dữ liệu được gắn với BXL P^i (CSDL D ban đầu được chia ra làm N phần, mỗi phần gắn với một BXL). Thuật toán bao gồm các bước sau:

Bước 1: Tương tự như trong thuật toán phân phối độ hỗ trợ

Bước 2: Với $k > 1$:

Mỗi BXL P^i tạo tập các tập thuộc tính ứng cử viên C_k từ tập các tập thuộc tính phổ biến L_{k-1} . Nó không thao tác tất cả trên C_k mà chỉ giữ lại một phần của C_k được chia đều cho N BXL. Phần được giữ lại cho BXL P^i được xác định nhờ định danh tiến trình (process identification) mà không cần truyền thông giữa các tiến

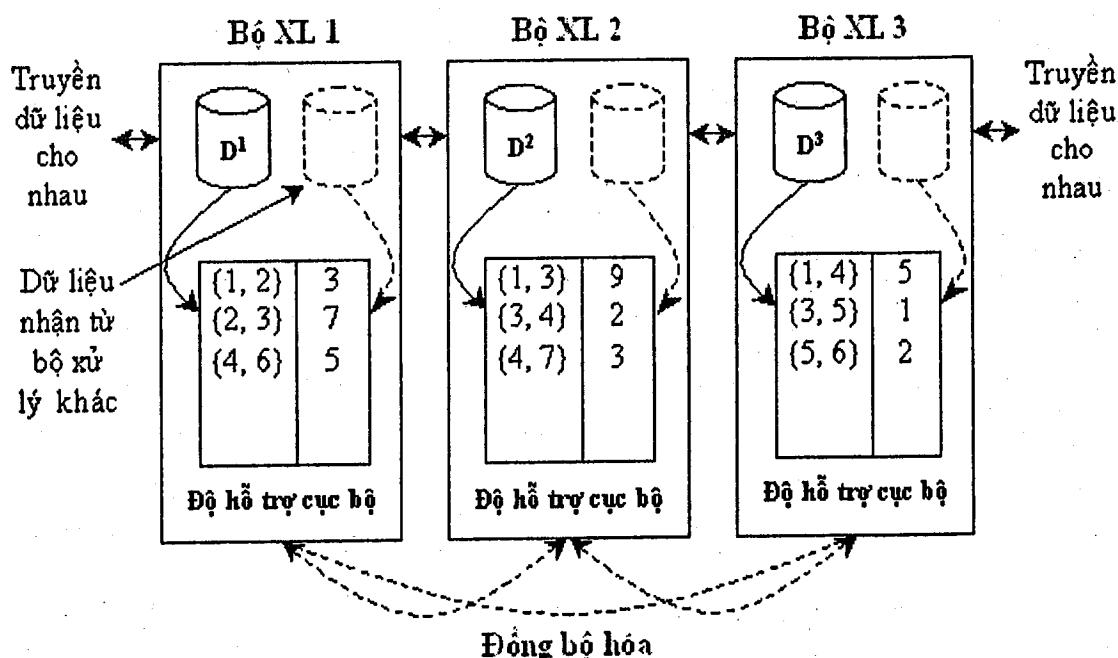
trình. Các C_k^i được chia thỏa mãn: $C_k^i \cap C_k^j = \emptyset$ (với mọi $i \neq j$) và

$$\bigcup_{i=1}^N C_k^i = C_k$$

BXL P^i chỉ đếm độ hỗ trợ cho các tập mục ứng cử viên trong C_k^i bằng cách sử dụng dữ liệu cục bộ D^i của nó và dữ liệu nhận được từ các BXL khác trong hệ thống.

Sau khi đếm xong độ hỗ trợ, mỗi BXL P^i chọn ra tập những tập thuộc tính phổ biến cục bộ L_k^i từ C_k^i tương ứng. Nhớ rằng

$$L_k^i \cap L_k^j = \emptyset \quad (\text{với mọi } i \neq j) \text{ và } \bigcup_{i=1}^N L_k^i = L_k$$



Hình 4.9. Thuật toán phân phối dữ liệu trên 3 BXL

Các BXL tiến hành trao đổi L_k^i cho nhau sao cho tất cả các BXL đều nhận được L_k để sinh C_k cho lần lặp tiếp theo. Bước này cần sự đồng bộ hóa giữa các BXL. Sau khi nhận được bước L_k , mỗi BXL có thể độc lập quyết định ngừng làm việc hoặc tiếp tục thực hiện bước lặp tiếp theo. Hình 4.9 minh họa nguyên lý làm việc của thuật toán này.

4.4.3. Thuật toán phân phối tập ứng cử viên

Hạn chế của hai thuật toán trên (count & data distribution) ở chỗ do mọi giao dịch hoặc bản ghi trong CSDL đều có thể hỗ trợ một tập thuộc tính ứng cử viên nào đó nên các giao dịch hay bản ghi phải được đổi sánh với tất cả các tập thuộc tính ứng cử viên. Điều này dẫn đến việc thuật toán phân phối độ hỗ trợ phải lưu giữ tập các tập ứng cử viên giống nhau trên mọi BXL và thuật toán phân phối dữ liệu phải gửi dữ liệu cho nhau trong quá trình cập nhật độ hỗ trợ. Hơn nữa, hai thuật toán này phải tiến hành đồng bộ hóa ở cuối mỗi pha thực hiện song song để trao đổi độ hỗ trợ cục bộ hoặc tập các tập phổ biến cho nhau. Yêu cầu đồng bộ hóa trong suốt thời gian thực hiện của thuật toán sẽ làm giảm hiệu suất thực hiện của hệ thống do các BXL hoàn thành công việc sớm phải “chờ đợi” các BXL hoàn thành công việc muộn hơn. Nguyên nhân của vấn đề này là do hai thuật toán trên mới chia công việc cho các BXL một cách “công bằng” chứ chưa chia một cách vừa “công bằng” vừa “khôn ngoan”.

Thuật toán phân phối tập ứng cử viên (candidate distribution) cố gắng chia tập ứng cử viên sao cho các BXL có thể độc lập làm việc và hạn chế tối đa công việc đồng bộ hóa. Bắt đầu một pha l nào đó (l được xác định dựa theo kinh nghiệm), thuật toán này chia tập thuộc tính phổ biến $L_{l,1}$ cho các BXL sao cho mỗi BXL P^i có thể tạo ra tập ứng cử viên C_m^i ($m \geq l$) độc lập với các BXL khác ($C_m^i \cap C_m^j = \emptyset, \forall i \neq j$). Đồng thời, dữ liệu cũng được chia lại sao cho mỗi BXL P^i có thể cập nhật độ hỗ trợ cho các tập ứng cử viên trong C_m^i độc lập với các BXL khác. Đúng thời gian đó, dữ liệu được phân chia lại sao cho mỗi BXL P^i có thể cập nhật độ hỗ trợ cho các tập thuộc tính ứng cử viên trong C_m^i một cách độc lập với các BXL khác. Nhớ rằng, sự phân chia dữ liệu phụ thuộc rất nhiều vào bước phân chia tập ứng cử viên trước đó. Nếu phân chia tập ứng cử viên không “khéo léo” thì chúng ta không thể có một phân hoạch dữ liệu cho các BXL mà chỉ có một phân chia tương đối – nghĩa là có thể có những phần dữ liệu trùng lặp trên các BXL.

Sau khi phân hoạch L_{k-1} , các BXL làm việc độc lập với nhau. Việc cập nhật độ hỗ trợ cho tập các ứng cử viên cục bộ không đòi hỏi các BXL phải truyền thông với nhau. Chỉ có một sự phụ thuộc duy nhất giữa các BXL là chúng phải gửi cho nhau những thông tin cần cho việc cắt tỉa các ứng cử viên không cần thiết. Tuy nhiên, những thông tin này có thể được truyền cho nhau theo chế độ dị bộ và các BXL không cần phải đợi để nhận đầy đủ thông tin này từ các BXL khác. Các BXL cố gắng cắt tỉa được càng nhiều càng tốt nhờ vào những thông tin đến từ các BXL khác. Những thông tin đến muộn sẽ được sử dụng cho lần cắt tỉa tiếp theo. Thuật toán phân phối tập ứng cử viên bao gồm những bước sau:

Bước 1 ($k < l$): Sử dụng một trong hai thuật toán phân phối độ hỗ trợ hoặc phân phối dữ liệu.

Bước 2 ($k = l$):

Phân chia L_{k-1} cho N BXL. Chúng ta sẽ xem xét cách phân chia ở phần sau. Quá trình phân chia này là giống hệt nhau và được thực hiện song song trên các BXL.

Mỗi BXL P^i sẽ sử dụng L_{k-1}^i để tạo ra C_k^i của nó.

P^i sẽ cập nhật độ hỗ trợ cho các tập ứng cử viên trong C_k^i và CSDL sẽ được phân chia lại ngay sau đó.

Sau đó, P^i thực hiện trên dữ liệu cục bộ và tất cả dữ liệu nhận được từ các BXL khác. Nó tạo ra $N-1$ bộ đệm nhận dị bộ để nhận các L_k^j từ các BXL khác. Những L_k^j này cần thiết cho bước cắt tỉa các tập ứng cử viên trong C_{k+1}^i .

P^i sinh ra L_k^i từ C_k^i và truyền thông lan truyền (broadcast) dị bộ tới $N-1$ bộ vi xử lý khác.

Bước 3 ($k > l$):

Mỗi BXL P^i thu thập tất cả những tập phổ biến từ các BXL khác. Thông tin về các tập phổ biến này sẽ được dùng để cắt tỉa. Các tập thuộc tính nhận được từ BXL j sẽ có độ dài $k-1$, nhỏ hơn $k-1$ (nếu là BXL chậm), hoặc lớn hơn $k-1$ (nếu là BXL nhanh).

P^i tạo ra C_k^i dựa vào L_{k-1}^i . Một trường hợp có thể xảy ra là P^i không nhận được L_{k-1}^j từ các BXL khác, do đó P^i cần phải “cẩn thận” trong khoảng thời gian cắt tỉa.

P^i thực hiện duyệt dữ liệu để cập nhật độ hỗ trợ cho các tập thuộc tính trong C_k^i . Sau đó nó tính toán L_k^i từ C_k^i và truyền dữ bộ thông tin về L_k^i tới $N-1$ BXL còn lại trong hệ thống.

Chiến lược phân chia dữ liệu: Chúng ta xem xét cách phân chia dữ liệu của thuật toán này thông qua một ví dụ đơn giản sau đây.

Cho $L_3 = \{ABC, ABD, ABE, ACD, ACE, BCD, BCE, BDE, CDE\}$.

$L_4 = \{ABCD, ABCE, ABDE, ACDE, BCDE\}$,

$L_5 = \{ABCDE\}$,

$L_6 = \emptyset$.

Chúng ta xét tập $\varepsilon = \{ABC, ABD, ABE\}$ với các thành viên của nó có chung phần đầu là AB . Nhớ rằng, các tập thuộc tính $ABCD, ABCE, ABDE$ và $ABCDE$ cũng có chung tiền tố AB .

Do đó, giả sử rằng các thuộc tính trong tập thuộc tính được sắp theo thứ tự từ vựng, chúng ta có thể phân chia các tập phổ biến trong L_k dựa vào tiền tố có độ dài $k-1$ đầu tiên của các tập, nhờ vậy các BXL có thể làm việc độc lập với nhau.

Cài đặt thuật toán này trong thực tế phức tạp hơn rất nhiều bởi hai lý do. Lý do thứ nhất là một BXL có thể phải nhận các tập thuộc tính phổ biến được tính toán bởi các BXL khác cho bước cắt tỉa tiếp theo. Trong ví dụ trên, BXL được gán tập ứng cử viên ε phải biết $BCDE$ có phải là tập phổ biến hay không mới quyết định được có cắt tỉa tập $ABCDE$ hay không, nhưng tiền tố của $BCDE$ là BC nên $BCDE$ lại thuộc về một BXL khác. Lý do thứ hai là chúng ta phải tính toán cân bằng tải cho các BXL trong hệ thống.

4.4.4. Thuật toán sinh luật song song

Cho một tập phổ biến h , chương trình con sinh luật kết hợp sẽ sinh ra luật dạng $a \Rightarrow (h - a)$, trong đó a là một tập con khác

rỗng của h . Độ hỗ trợ của luật chính là độ hỗ trợ của tập phổ biến h (tức là $s(h)$), còn độ tin cậy của luật là tỷ số $s(h)/s(a)$.

Để sinh luật hiệu quả, chúng ta tiến hành duyệt các tập con của h có kích thước lớn trước tiên và sẽ tiếp tục xét các tập con nhỏ hơn khi luật vừa sinh thỏa mãn độ tin cậy tối thiểu (minconf). Ví dụ, h là tập phổ biến $ABCD$, nếu luật $ABC \Rightarrow D$ không thỏa mãn độ tin cậy tối thiểu thì luật $AB \Rightarrow CD$ cũng không thỏa mãn do độ hỗ trợ của AB luôn lớn hơn hoặc bằng ABC . Như vậy chúng ta không cần xét các luật mà惟 trái là tập con của ABC vì chúng không thỏa mãn độ tin cậy tối thiểu.

Thuật toán sinh luật tuần tự [AS94] thể hiện ý tưởng trên như sau:

```

Forall frequent itemset  $h_k, k > 1$  do
    Call gen_rules( $h_k, h_k$ );
    // The gen_rules generates all valid rules  $\alpha \geq (l - \alpha)$ ,
    // for all  $\alpha \in a_m$ 
    Procedure gen_rules( $h_k$ : frequent k-itemset,  $a_m$ : frequent m-itemset)
        1.       $A = \{(m-1)-itemsets a_{m-1} | a_{m-1} \subset a_m\}$ 
        2.      Forall  $a_{m-1} \subset A$  do
            3.          conf =  $s(h_k)/s(a_{m-1})$ ;
            4.          if (conf  $\geq$  minconf) then
                5.              output the rule  $a_{m-1} \Rightarrow (h_k - a_{m-1})$ ;
                6.              if ( $m - 1 > 1$ ) then
                    7.                  Call gen_rules( $h_k, a_{m-1}$ );
                8.          end
        9.      end
    
```

Để sinh luật song song, chúng ta chia tập các tập thuộc tính phổ biến cho tất cả các BXL trong hệ thống. Mỗi BXL sinh luật trên các tập phổ biến được phân chia cho nó sử dụng thuật toán trên. Trong thuật toán sinh luật song song, để tính độ tin cậy của một luật, BXL có thể cần phải tham chiếu đến độ hỗ trợ của một tập phổ biến nằm trên một BXL khác. Vì lý do này, các BXL nên có thông tin về toàn bộ các tập phổ biến trước khi thực hiện thuật toán sinh luật song song.

4.4.5. Một số thuật toán khác

Ngoài ba thuật toán nêu trên, các nhà nghiên cứu trong lĩnh vực này đã đề xuất thêm khá nhiều thuật toán khai phá luật kết hợp song song khác.

Thuật toán phân phối dữ liệu thông minh (Intelligent Data Distribution Algorithm) [HKK97] được đề xuất dựa trên thuật toán phân phối dữ liệu với một bước cải tiến trong việc truyền dữ liệu giữa các BXL trong thời gian tính toán. Thay vì truyền dữ liệu giữa cặp BXL, các BXL trong thuật toán này được tổ chức thành một vòng logic và chúng tiến hành truyền dữ liệu theo vòng tròn này.

Thuật toán MLFPT (Multiple Local Frequent Pattern Tree) [ZHL98] là thuật toán dựa trên FP-growth. Thuật toán này giảm được số lần duyệt qua CSDL, không cần tạo ra tập ứng cử viên và cân bằng tải giữa các BXL trong hệ thống.

Thuật toán khai phá luật kết hợp song song do [ZPO01] đề xuất khác với các thuật toán khác ở chỗ nó làm việc trên hệ thống đa xử lý đối xứng (SMP, còn được gọi là shared-everything system) thay vì trên hệ song song phân tán không chia sẻ tài nguyên (shared-nothing system).

4.5. MỘT SỐ ỨNG DỤNG CỦA LUẬT KẾT HỢP

Ngoài việc áp dụng các kĩ thuật phân tích luật kết hợp để hỗ trợ kinh doanh, tìm hiểu thói quen mua sắm của khách hàng như trên. Luật kết hợp cũng được áp dụng để phát hiện thông tin trong một số lĩnh vực khác như:

Các khái niệm có liên quan: Coi các từ là các mục và tài liệu là một giao dịch (ví dụ trang web, blogs, tweets...). Một tài liệu sẽ chứa rất nhiều từ trong đó. Nếu ta bỏ qua tất cả những từ thông dụng như ‘và’, ‘nhưng’... chúng ta có thể tìm ra trong các cặp từ thường xuyên xuất hiện cùng nhau được một số cặp từ mà có

quan hệ kết hợp với nhau. Ví dụ các cặp như {Brad, Angelina}, {Mac, Angen}...

Vi phạm bản quyền: Ta coi mỗi mục là một tài liệu và mỗi giao dịch là một câu. Thứ tự này ngược so với thực tế suy nghĩ thông thường. Nhưng đối với bài toán tìm ra việc sao chép trái phép thì thứ tự này bị đảo ngược là có ý đồ. Bởi vì ta thấy nhiều tài liệu có thể cùng chứa một câu. Ta cần phải tìm các cặp mục (tài liệu) mà có cùng trong một giao dịch (câu). Có thể hiểu rằng giao dịch được gán nhãn là một câu và giao dịch này chứa các mục là các tài liệu. Trong thực tế, chỉ cần phát hiện các tài liệu có 1-2 câu giống nhau thì cũng là dấu hiệu thuận lợi để tìm vi phạm bản quyền tài liệu.

Dấu hiệu sinh học: Coi các mục là dữ liệu gồm 1 bộ 2 thuộc tính là gen (hoặc protein máu) và bệnh tật. Mỗi giao dịch là một tập dữ liệu về một bệnh nhận như bộ gen, phân tích sinh hóa máu và lịch sử bệnh. Một tập mục phổ biến bao gồm một bệnh và một hoặc nhiều gen, protein quy định. Nó có khả năng gợi ý, hỗ trợ chuẩn đoán bệnh tật của người bệnh.

Hệ hỗ trợ ra quyết định trong chứng khoán: mỗi giao dịch là một tập các mã cổ phiếu trong mỗi phiên và chỉ lấy các mã mà người dùng quan tâm. Mỗi mục là một một mã cổ phiếu. Trong một giao dịch, mỗi mục (mã cổ phiếu) chỉ xuất hiện nếu giá của nó tăng trong phiên đó. Hệ hỗ trợ sẽ gợi ý nhà đầu tư những mã cổ phiếu nào có khả năng cao sẽ tăng cùng nhau trong một phiên.

Một số kỹ thuật mới như luật kết hợp hiếm, luật kết hợp âm... đã được phát triển trong thời gian gần đây nhằm tăng khả năng của luật kết hợp và mở rộng phạm vi ứng dụng của luật kết hợp.

CÂU HỎI VÀ BÀI TẬP

4.1. Cho CSDL D với 4 thuộc tính z, y, z và t với mỗi thuộc tính có 3 giá trị khác nhau. Cho biết có thể tạo ra được bao nhiêu luật nếu chỉ có 1 thuộc tính ở bên phải luật?

4.2. Giả sử L_3 bao gồm danh sách sau

$\{\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{b, c, w\}, \{b, c, x\}, \{p, q, r\}, \{p, q, s\}, \{p, q, t\}, \{p, r, s\}, \{q, r, s\}\}$

Tập mục nào sẽ bị loại bỏ ở bước nối tiếp theo C_4 ? Và tập nào sẽ bị loại bỏ ở bước tách bớt L_4 ?

4.3. Cho biết CSDL giao dịch gồm 5000 giao dịch và ta thu được 1 luật tương ứng $L \rightarrow R$ với các độ hỗ trợ sau

$$\text{Count}(L) = 3400$$

$$\text{Count}(R) = 4000$$

$$\text{Count}(L \cup R) = 3000$$

Tìm độ hỗ trợ và độ tin cậy của luật trên?

4.4. Cho CSDL giao dịch gồm 8 giao dịch với thứ tự giao dịch như sau:

TID	Tập mục trong giao dịch
1	{a, b, c}
2	{a, b, c, d, e}
3	{b}
4	{c, d, e}
5	{b}
6	{b, c, d}
7	{c, d, e}
8	{c, e}

Tìm tất cả các luật có thể sinh ra được bởi $\text{minsup} = 25\%$ và $\text{minconf} = 80\%$

Chương 5.

PHÂN CỤM DỮ LIỆU

5.1. GIỚI THIỆU

5.1.1. Bài toán phân cụm

Một trong những bài toán quan trọng trong lĩnh vực khai phá dữ liệu là bài toán phân cụm. Trong nhiều trường hợp, ta có một tập dữ liệu lớn chưa có nhãn (đánh dấu xem một phần tử dữ liệu là thuộc lớp nào), lý do là việc gán nhãn cho các phần tử dữ liệu là rất tốn kém. Ví dụ trong cơ sở dữ liệu của ngân hàng chứa một số lượng lớn các giao dịch của khách hàng, việc gán nhãn cho các khách hàng nào thuộc khách hàng tiềm năng có thể mang lại lợi nhuận cho ngân hàng là điều rất khó thực hiện. Một trong những giải pháp để xử lý vấn đề này là tự động nhóm các phần tử dữ liệu có độ tương tự nhau (giống nhau) vào cùng một cụm. Các phần tử trong cùng một cụm sẽ có độ tương tự lớn, và độ tương tự giữa các phần tử trong cùng một cụm sẽ lớn hơn độ tương tự giữa nó với một phần tử dữ liệu trong cụm khác. Hay nói một cách khác, các phần tử giữa các cụm khác nhau sẽ có độ khác biệt hẳn với nhau. Độ đo khác biệt được đo dựa trên giá trị của các thuộc tính mô tả phần tử dữ liệu, thông thường người ta thường sử dụng độ đo khoảng cách để đo độ khác biệt.

Phân cụm cũng là một việc rất tự nhiên, giống như việc chúng ta phân loại động vật thành các loài, các họ khác nhau (hay các nhóm có cùng một số đặc điểm nào đó, và các đặc điểm này lại rất khác với các loại khác). Trong lớp học, người ta có thể phân ra các nhóm sinh viên học giỏi, học khá, học kém,... Phân cụm được

sử dụng rộng rãi trong rất nhiều lĩnh vực (bài toán) như nghiên cứu thị trường, nhận dạng mẫu, phân tích dữ liệu, xử lý ảnh, ... Trong kinh doanh, phân cụm có thể giúp ta phân khách hàng thành các nhóm khác nhau đồng thời cho ta biết các đặc trưng của các nhóm người dùng này, từ đó công ty sẽ có các chính sách khác nhau cho các nhóm khách hàng này.

Việc phân cụm dữ liệu là bài toán cần được một cách tự động, do đó nó thuộc vào lớp các bài toán học không giám sát (unsupervised learning).

5.1.2. Các phương pháp phân cụm

Việc phân loại các giải thuật phân cụm là bài toán không đơn giản, lý do là có nhiều tiêu chí phân loại, hơn nữa có nhiều giải thuật có cùng một số đặc trưng nên việc phân loại cũng không thể tách bạch được. Hay nói cách khác, giữa các phân loại có sự giao nhau. Dưới đây liệt kê một số cách phân loại các phương pháp phân cụm:

1. *Phân cụm phẳng và phân cụm phân cấp*: Phân cụm phẳng chỉ đơn giản chia tập dữ liệu thành một số tập con không giao nhau. Phân cụm phẳng còn được gọi là phương pháp phân cụm phân vùng (partitioning), lý do là khi biểu diễn trên mặt phẳng thì mỗi một cụm sẽ tương ứng với một vùng. Một trong các giải thuật thuộc lớp giải thuật phân cụm phẳng là *k-means*. Còn phân cụm phân cấp tạo ra một cây phân cấp của các cụm: trên mỗi nút trong cây sẽ tương ứng với một cụm, cụm ở nút cha sẽ là hợp của các cụm nút con. Việc phân hoạch có thể thực hiện theo hai cách (hay hai phương pháp): gộp (agglomerative) hay chia/tách (divisive).

- Phương pháp phân cụm gộp, ban đầu sẽ coi từng phần tử dữ liệu là các cụm đơn. Giải thuật sẽ lần lượt gộp (ghép) các cụm đơn có độ tương tự nhau cao vào thành một cụm lớn hơn. Quá trình gộp các cụm sẽ được lặp đi

lặp lại cho đến khi chúng ta thu được một cụm duy nhất (nút gốc) hoặc thỏa mãn 1 điều kiện dừng nào đó (xem chi tiết ở phần giải thuật). Phương pháp phân cụm gộp còn được gọi là phân cụm từ dưới lên (bottom-up), lý do là cây phân cấp được xây dựng từ lá đến gốc (từ dưới lên trên).

- Phương pháp phân cụm chia, ban đầu sẽ coi toàn bộ tập dữ liệu là một cụm (nút gốc), cụm này sẽ được chia nhỏ ra thành các cụm con. Từng cụm con sẽ được tiếp tục chia nhỏ ra thành các cụm nhỏ hơn. Quá trình chia tiếp tục cho đến khi mỗi cụm chỉ chứa một phần tử dữ liệu hoặc thỏa mãn điều kiện dừng nào đó. Giải thuật này cũng còn được gọi là phương pháp phân cụm từ trên xuống, lý do là việc xây dựng cây phân cấp được tiến hành từ gốc đến lá (từ trên xuống dưới).

Một trong những nhược điểm của phương pháp phân cụm phân cấp là khi một phần tử đã được phân vào một cụm thì nó sẽ không bao giờ được phân lại vào cụm khác. Do đó nếu việc phân đó là sai thì nó sẽ tạo ra lỗi và lỗi đó sẽ không được chỉnh sửa.

2. *Phân cụm dựa vào mật độ (density-based)*: Phân lớn các giải thuật phân cụm thường dựa vào độ đo khoảng cách để quyết định việc phân dữ liệu vào các cụm, dẫn đến các cụm được tạo ra thường có dạng hình cầu (có tâm là trọng tâm của cụm). Do đó phương pháp này chỉ phù hợp khi các cụm được phân bố theo hình cầu. Tuy nhiên điều này sẽ không đáp ứng được các thể loại dữ liệu khác nhau trong thực tế (chúng có thể tồn tại ở bất kỳ hình dạng nào). Phương pháp phân cụm này sẽ dựa vào *mật độ* phân bố của dữ liệu để quyết định gán một phần tử dữ liệu vào các cụm. Mật độ ở đây được định nghĩa là số lượng các phần tử lân cận (neighbour) trong một bán kính nào đó và tâm là phần tử đang xét. Một cụm sẽ được tiếp tục có thêm phần tử dữ liệu đang xét nếu mật độ của nó lớn hơn 1 ngưỡng nào đó. Một số các giải thuật phân cụm thuộc lớp này là DBSCAN, OPTICS và DENCLUE.

3. Phương pháp phân cụm dựa trên lưới (grid-based):

Phương pháp này chia không gian dữ liệu thành một lưới (grid) chứa một số lượng hữu hạn các ô (cell). Toàn bộ các thao tác phân cụm sẽ được thực hiện dựa trên các ô này. Ưu điểm của phương pháp này là thời gian xử lý, do nó chỉ phụ thuộc vào số lượng các ô chứ không phụ thuộc vào số lượng các phần tử dữ liệu. Giải thuật phân cụm thuộc lớp này là STING.

4. Phương pháp phân cụm dựa trên mô hình (model):

Phương pháp này giả thiết là có một mô hình tương ứng biểu diễn một cụm, giải thuật sẽ tìm các phần tử dữ liệu để phân vào các cụm sao cho phù hợp với mô hình nhất. Giải thuật phân cụm dựa trên mô hình thường tạo ra các cụm bằng cách xây dựng các hàm mật độ phản ánh sự phân bố của dữ liệu trong không gian. Giải thuật này cũng có thể được sử dụng để tìm ra số lượng cụm tối ưu một cách tự động dựa vào thống kê. Giải thuật cực đại kỳ vọng Expectation Maximization (EM) là thuộc lớp phân cụm dựa trên mô hình.

5. Phân cụm đơn định (deterministic) và phân cụm xác suất (probability):

Trong phân cụm đơn định, mỗi một phần tử dữ liệu chỉ phụ thuộc vào một cụm (hay xác suất của phần tử đó thuộc vào trong cụm nó được phân là 100%, còn xác suất của nó thuộc vào các cụm khác là 0%). Việc chỉ cho phép một phần tử dữ liệu thuộc vào một lớp trong một số trường hợp là không chính xác. Ví dụ nếu chúng ta phân những người có tuổi nhỏ hơn hoặc bằng 30 là thuộc lớp trẻ, còn lớn hơn 30 là thuộc lớp già là không tự nhiên. Vì 2 người (một người 30 người kia 31) chỉ hơn nhau 1 tuổi đã thuộc 2 lớp khác nhau. Phương pháp phân cụm xác suất sẽ gán xác suất mà một phần tử dữ liệu thuộc vào một lớp, xác suất này có giá trị nằm trong khoảng [0,1]. Trong trường hợp này người 30 tuổi sẽ có một xác suất (>0) thuộc lớp già và ngược lại người 31 tuổi

cũng sẽ có một xác suất (>0) thuộc vào lớp trẻ. Ví dụ về giải thuật phân cụm thuộc loại phương pháp này là phân phương pháp phân cụm mờ (fuzzy).

6. **Phân cụm dữ liệu có số chiều lớn (high-dimensional data):** trong một số miền ứng dụng, số lượng chiều của dữ liệu là rất lớn, chẳng hạn như xử lý văn bản (text) hay xử lý dữ liệu chuỗi DNA. Đặc điểm của loại dữ liệu này là số chiều lớn, có nhiều chiều không liên quan, dữ liệu thừa (vì số chiều lớn), mật độ dữ liệu nhỏ. Do đó ta cần một lớp giải thuật để giải quyết loại dữ liệu này. Một số giải thuật thuộc lớp này là CLIQUE và PROCLUS. Ý tưởng của các giải thuật này là tìm ra một tập con các thuộc tính (chiều) có liên quan và thao tác trên tập các thuộc tính đó.

7. **Phân cụm dựa trên ràng buộc (constraint-based):** Giải thuật thuộc lớp này sẽ được bổ sung thêm một số ràng buộc khi thực thi. Mỗi một ràng buộc sẽ thể hiện một yêu cầu (kỳ vọng) của người dùng hay nó mô tả thuộc tính (property) của cụm kết quả. Phương pháp này cho phép sự tương tác giữa người dùng và giải thuật. Ví dụ trong dữ liệu giao dịch của một siêu thị người ta chỉ muốn phân cụm các khách hàng mỗi lần mua có số tiền lớn 3 triệu.

8. **Phân cụm theo lô (batch) và phân cụm gia tăng (incremental):** Phương pháp phân loại này dựa vào cách thức xử lý dữ liệu của giải thuật. Trong phân theo lô, toàn bộ tập dữ liệu được sử dụng để tạo ra các cụm. Nếu chúng ta có thêm 1 phần tử dữ liệu mới, thì nó sẽ tạo ra một tập dữ liệu mới và giải thuật phân cụm lại phải thực thi trên tập dữ liệu mới này để phân cụm lại. Do đó giải thuật phân cụm theo lô chỉ phù hợp khi tập dữ liệu là ít biến đổi (nếu không thì độ phức tạp thuật toán sẽ cao). Trong phân cụm gia tăng, giải thuật phân cụm lấy từng phần tử dữ liệu và cập nhật các cụm để phân vào cụm thích hợp. Khi có thêm phần tử dữ liệu mới thì nó chỉ làm nhiệm vụ phân phần tử đó vào cụm thích hợp chứ không cần phải phân cụm lại

những phần tử dữ liệu đã được phân trước đó. Giải thuật này rất thích hợp khi tập dữ liệu luôn luôn biến đổi.

Vì số lượng các giải thuật phân cụm rất lớn nên trong chương này ta chỉ tập trung giới thiệu một số giải thuật điển hình. Một số ký hiệu được sử dụng chung cho các giải thuật phân cụm trong chương là: D - tập dữ liệu cần phân cụm, nó gồm n phần tử dữ liệu; Một phần tử dữ liệu p (viết tắt từ point) được biểu diễn bằng d thuộc tính (chiều).

5.2. MỘT SỐ ĐỘ ĐO CƠ BẢN DÙNG TRONG PHÂN CỤM

5.2.1. Độ đo tương đồng

Giả sử trong một miền dữ liệu D , một phần tử dữ liệu p được biểu diễn bằng một vector có số chiều là n (p_1, p_2, \dots, p_n), trong đó mỗi chiều biểu diễn một thuộc tính mô tả phần tử dữ liệu p . Tùy vào kiểu giá trị biểu diễn mà độ tương tự giữa hai phần tử dữ liệu p_1 và p_2 có thể được tính toán bằng các cách khác nhau.

- Trường hợp các giá trị thuộc tính được biểu diễn bằng các giá trị nhị phân $p_i \in \{0,1\}$, ta lập bảng mô tả số lượng các thuộc tính có cùng giá trị và các thuộc tính không cùng giá trị như bảng 5.1. Khi đó độ đo Jaccard xác định độ tương tự giữa 2 phần tử dữ liệu p_1 và p_2 được định nghĩa như sau:

$$Jaccard(p_1, p_2) = \frac{a}{a+b+c} \quad (5.1)$$

Bảng 5.1. Ma trận kề

Phần tử dữ liệu p_2	Phần tử dữ liệu p_1		Tổng
	1	0	
1	a	b	$a+b$
0	c	d	$c+d$
Tổng	$a+c$	$b+d$	$a+b+c+d$

- Trường hợp thuộc tính A_i có giá trị p_i được biểu diễn bằng các giá trị rời rạc $p_i \in \{0, 1, \dots, m\}$ thì ta biến thuộc tính A_i thành m thuộc tính nhị phân sau đó áp dụng công thức Jaccard ở trên để đo độ tương tự. Ví dụ thuộc tính màu (color) có các giá trị rời rạc là {xanh, đỏ, vàng}, khi đó ta biến thuộc tính màu này thành 3 thuộc tính xanh, đỏ, vàng với giá trị của các thuộc tính này là các giá trị nhị phân {0, 1};
- Trường hợp giá trị biểu diễn p_i các thuộc tính là liên tục (hay là các số thực), thì một trong các công thức hay dùng để đo độ tương tự là độ đo cosin():

$$\cos(p_1, p_2) = \frac{p_1 \cdot p_2}{\|p_1\| \|p_2\|} = \frac{\sum_{i=1}^n p_{1i} p_{2i}}{\sqrt{\sum_{i=1}^n p_{1i}^2} \sqrt{\sum_{i=1}^n p_{2i}^2}} \quad (5.2)$$

5.2.2. Độ đo khác biệt

Trong nhiều trường hợp ta có thể sử dụng *độ đo khác biệt* (dissimilarity) thay cho độ tương tự: một trong những độ đo khác biệt là *độ đo khoảng cách* (distance). Tương tự như độ tương tự, tùy thuộc vào giá trị biểu diễn các thuộc tính mà các độ đo khoảng cách sẽ được tính toán bằng những công thức khác nhau.

- Trường hợp các giá trị thuộc tính được biểu diễn bằng các giá trị nhị phân $p_i \in \{0, 1\}$. Trường hợp này ta còn phân nhỏ ra là *thuộc tính đối xứng* (symmetric) và *thuộc tính bất đối xứng* (asymmetric). Thuộc tính đối xứng là thuộc tính mà giá trị của nó dù là 0 hay 1 thì ý nghĩa của nó cũng không tạo sự khác biệt nhau. Ví dụ trong một cơ sở giao dịch mua hàng thì thuộc tính giới tính (nam được biểu diễn bằng giá trị 1, nữ là giá trị 0), thì chúng ta không thấy sự khác biệt về người mua cho dù đó là nam hay nữ. Tuy nhiên nếu ta xét thuộc tính biểu diễn một giao dịch có mua mặt hàng máy tính hay không, thì thuộc tính này bằng 1 (có mua) và

bằng 0 (không mua) có ý nghĩa khác hẳn nhau. Hay một ví dụ về thuộc tính chứa kết quả xét nghiệm máu xem một bệnh nhân có bị viêm gan B hay không, nếu có bị nhiễm thì giá trị là 1 và không bị nhiễm giá trị bằng 0 sẽ có ý nghĩa khác hẳn nhau. Do đó, các công thức tính độ đo khoảng cách cũng sẽ khác nhau dựa vào bảng ma trận kề như bảng 5.1.

- Nếu là thuộc tính đối xứng thì khoảng cách d được tính bằng công thức:

$$d(p_1, p_2) = \frac{b + c}{a + b + c + d} \quad (5.3)$$

- Nếu là thuộc tính bất đối xứng thì khoảng cách d được tính bằng công thức:

$$d(p_1, p_2) = \frac{b + c}{a + b + c} \quad (5.4)$$

- Trường hợp thuộc tính A_i có giá trị p_i được biểu diễn bằng các giá trị rời rạc $p_i \in \{0, 1, \dots, m\}$ thì ta biến thuộc tính A_i thành m thuộc tính nhị phân sau đó áp dụng công thức khoảng cách ở trên để đo độ khác biệt. Một phương pháp khác đơn giản hơn là tìm số lượng các thuộc tính mà p_1 và p_2 có cùng giá trị. Giả sử chúng có p thuộc tính có giá trị giống nhau, thì độ đo khoảng cách được tính bằng:

$$d(p_1, p_2) = \frac{n - q}{n} \quad (5.5)$$

trong đó, n là số lượng các thuộc tính.

- Trường hợp giá trị biểu diễn p_i các thuộc tính là liên tục (hay là các số thực), thì ta có một số các công thức đo khoảng cách như sau:

- Độ đo khoảng cách Manhattan:

$$d(p_1, p_2) = \sum_{i=1}^n |p_{1i} - p_{2i}| \quad (5.6)$$

- Độ đo khoảng cách Euclidean:

$$d(p_1, p_2) = \sqrt{\sum_{i=1}^n |p_{1i} - p_{2i}|^2} \quad (5.7)$$

- Độ đo khoảng cách Minkowski:

$$d(p_1, p_2) = \sqrt[q]{\sum_{i=1}^n |p_{1i} - p_{2i}|^q} \quad (5.8)$$

Nếu để ý thì ta sẽ thấy độ đo khoảng cách Manhattan là trường hợp đặc biệt của độ đo Minkowski với $q = 1$, còn trường hợp $q = 2$ thì nó chính là độ đo Euclidean.

Các độ đo khoảng cách trên đều có đặc điểm sau:

- Tính xác định dương (positive definiteness): $d(p_i, p_j) > 0$ nếu $i \neq j$ và $d(p_i, p_i) = 0$
- Tính đối xứng (symmetric): $d(p_i, p_j) = d(p_j, p_i)$
- Tính bất đẳng thức tam giác (triangle inequality): $d(p_i, p_j) \leq d(p_i, p_k) + d(p_k, p_j)$

Ví dụ về cách tính một số độ đo được minh họa như sau: Giả sử ta có một cơ sở dữ liệu trong bệnh viện chứa kết quả các xét nghiệm của các bệnh nhân như bảng 5.2.

Bảng 5.2. Bảng kết quả xét nghiệm

STT	Tên	Giới tính	Chóng mặt	Ho	XN1	XN2	XN3	XN4
1	Nam	M	Y	N	P	N	N	N
2	Vân	F	Y	N	P	N	P	N
3	Thắng	M	Y	P	N	N	N	N

trong đó giới tính M là nam (male), F là nữ (female); Thuộc tính *chóng mặt* được biểu diễn bằng giá trị Y (có)/ N (không); các xét nghiệm XN có giá trị P (dương tính- positive) và N (âm tính – negative). Lập bảng ma trận kề ta có thể tính được độ khác biệt của các phần tử dữ liệu như sau:

$$d(\text{Nam}, \text{Vân}) = (0+1)/(2+0+1) = 0,33$$

$$d(\text{Nam}, \text{Thắng}) = (1+1)/(1+1+1) = 0,67$$

$$d(\text{Thắng}, \text{Vân}) = (1+2)/(1+1+2) = 0,75$$

Trường hợp giá trị của các thuộc tính được biểu diễn bằng các số thực, trong nhiều trường hợp có thể ta sẽ cần phải chuẩn hóa trước khi tính toán nhằm làm tăng độ chính xác. Độc giả có thể tham khảo tại chương 3 của tài liệu [Han06].

5.3. THUẬT TOÁN PHÂN CỤM PHẢNG

5.3.1. Thuật toán k-means

Giải thuật k-means thuộc lớp phân cụm phẳng, đầu vào cho thuật toán k-means là tập dữ liệu D gồm n phần tử dữ liệu, số lượng các cụm đầu ra k . Đầu ra của giải thuật là k cụm dữ liệu. Giải thuật k-means được trình bày như sau:

Đầu vào: Tập dữ liệu D , số lượng các cụm k

Đầu ra: Tập dữ liệu đã được phân thành k cụm

Thuật toán k-means

1. Chọn ngẫu nhiên k phần tử trong D làm trọng tâm ban đầu cho các cụm.
 2. Phân các phần tử dữ liệu trong D vào các cụm dựa vào độ tương đồng của nó với trọng tâm của các cụm. Phần tử dữ liệu sẽ được phân vào cụm có độ tương đồng lớn nhất.
 3. Tính lại trọng tâm của các cụm.
 4. Nhảy đến bước 2 cho đến khi quá trình hội tụ (không có sự gán lại các phần tử dữ liệu giữa các cụm, hay trọng tâm của các cụm là không đổi).
-

Điểm mấu chốt của giải thuật là ở bước 2, các phần tử dữ liệu được di chuyển giữa các cụm để làm cực đại hóa độ tương tự giữa các phần tử dữ liệu bên trong 1 cụm (hay cực đại hóa độ tương tự

trong nội tại một cụm, hay cực tiểu hóa khoảng cách giữa các phần tử dữ liệu trong nội tại một cụm). Độ đo tương tự trong nội tại một cụm được tính bằng công thức:

$$J = \sum_{i=1}^k \sum_{p \in C_i} sim(p, m_i) \quad (5.9)$$

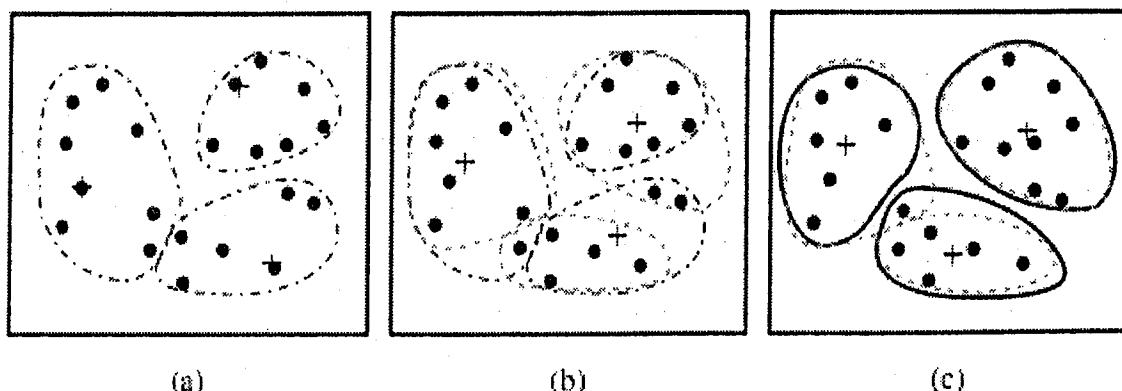
trong đó, C_i và m_i lần lượt là ký hiệu cụm thứ i và trọng tâm của nó. Và $sim(p, m_i)$ là độ tương tự giữa p và m_i . Trọng tâm m_i của C_i được tính theo công thức sau:

$$m_c = \sum_{p \in C} \frac{p}{|C|} \quad (5.10)$$

Nói một cách khác, giải thuật k-means hoạt động sao cho hàm điều kiện (criterion function) của nó là hội tụ. Thông thường hàm hội tụ được chọn là hàm *tổng bình phương lỗi* (squared-error) được định nghĩa như sau:

$$E = \sum_C \sum_{p \in C} |p - m_c|^2 \quad (5.11)$$

Giải thuật k-means trả về số lượng biến thể các cụm là tối thiểu, nhưng nó không đảm bảo tìm được giá trị cực đại toàn cục của hàm J nhưng ta có thể chạy thuật toán một số lần để thu được giá trị cực đại cục bộ. Giải thuật k-means phân các phần tử dữ liệu vào các cụm dựa vào trọng tâm của các cụm, do đó nó có tên là k-means (mean là giá trị trung bình).



Hình 5.1. Minh họa hoạt động của giải thuật k-means

Một ví dụ mô phỏng hoạt động của giải thuật k-means được minh họa trên hình 5.1. Ban đầu ta có tập dữ liệu như hình a), và giả sử số lượng các cụm $k = 3$. Thuật toán lựa chọn 3 phần tử dữ liệu ngẫu nhiên làm trọng tâm của các cụm (được đánh dấu bằng dấu + bên cạnh). Các phần tử dữ liệu sẽ được gán vào 3 cụm dựa vào độ tương tự của nó với 3 trọng tâm này. Chúng ta thu được 3 cụm được khoanh bằng đường đứt nét. Ở hình b) mô tả quá trình các trọng tâm được tính lại dựa vào các phần tử trong 1 cụm, sau đó các phần tử dữ liệu lại được gán lại dựa vào 3 trọng tâm mới (được đánh dấu bằng dấu +). Hình c) diễn tả quá trình tương tự và cuối cùng ta thu được 3 cụm đầu ra (được khoanh bằng đường liền nét).

Kết quả cuối cùng của k-means phụ thuộc rất nhiều vào cách lựa chọn k phần tử dữ liệu ban đầu làm trọng tâm của k cụm. Bởi vì sự lựa chọn k cụm ban đầu là hoàn toàn ngẫu nhiên, nên kết quả thu được sau khi chạy k-means các lần khác nhau là có thể khác nhau. Như vậy ta có thể chạy thuật toán k-means một số lần và lấy kết quả của lần chạy có giá trị của hàm J là lớn nhất. Ngoài ra cũng có một số các đề xuất để cải tiến thuật toán k-means bằng cách cải tiến việc xây dựng các trọng tâm ban đầu [Cui].

Trong thực tế khi ta gặp trường hợp dữ liệu quá lớn, hoặc giải thuật không hội tụ (trọng tâm của các cụm cứ liên tục thay đổi) dẫn đến thời gian chạy chương trình có thể rất lớn. Trong trường hợp này, người ta có thể sử dụng một số điều kiện dừng sau đây:

- Khi số lượng vòng lặp vượt qua một ngưỡng nào đó. Điều kiện này có thể làm cho chất lượng của giải thuật phân cụm không được tốt vì nó chưa chạy đủ số vòng lặp cần thiết.
- Khi giá trị của J nhỏ hơn 1 ngưỡng nào đó (đảm bảo chất lượng của các cụm đủ tốt, hay nó đã chạy được đủ số vòng lặp cần thiết). Trong thực tế điều kiện này thường được dùng kết hợp với điều kiện số vòng lặp ở trên.
- Khi hiệu của giá trị của J trong hai vòng lặp liên tiếp ($J_i - J_{i+1}$) nhỏ hơn 1 ngưỡng nào đó. Người ta cũng hay kết hợp điều kiện này với điều kiện vòng lặp để tránh chương trình bị chạy lặp.

Giả sử số lần lặp của giải thuật là t thì độ phức tạp của thuật toán là $O(nkt)$, trong đó n là số lượng các phần tử dữ liệu, k là số lượng các cụm. Thông thường trong thực tế thì $t \ll n$ và $k \ll n$. Với độ phức tạp này thì thuật toán thực hiện khá nhanh trên tập dữ liệu lớn. Như đã đề cập ở trên, giải thuật k-means dựa trên độ đo tương tự, nên nó phù hợp với miền dữ liệu mà các cụm phân bố theo hình cầu và nó hoạt động không tốt trong miền dữ liệu mà các cụm được phân bố theo hình dạng bất kỳ.

5.3.2. Thuật toán k-medoids

Một nhược điểm nữa của giải thuật k-means là nó nhạy cảm với các dữ liệu ngoại lệ (outlier). Giả sử trong tập dữ liệu có một số phần tử có giá trị lớn (nhưng bản thân chúng chỉ là các trường hợp ngoại lệ chứ không phải là phổ biến), khi đó các phần tử này sẽ có ảnh hưởng lớn đến trọng tâm của các cụm mà nó thuộc vào. Hệ quả là các cụm sẽ không được tối ưu và tổng bình phương lỗi sẽ cao. Giải thuật k-medoids được đề xuất để tránh nhược điểm trên. Trong giải thuật này, thay vì tính toán trọng tâm của cụm, nó lựa chọn 1 phần tử cụ thể trong cụm làm trọng tâm của cụm. Tiếp đến thay vì sử dụng hàm điều kiện là tổng bình phương lỗi E như công thức (5.11), hàm tổng số lỗi tuyệt đối (absolute-error) được dùng làm hàm điều kiện, nó được tính là tổng số lỗi tuyệt đối trên toàn bộ tập dữ liệu như công thức (5.12):

$$E = \sum_{i=1}^k \sum_{p \in C_i} |p - o_i|^2 \quad (5.12)$$

trong đó, p là phần tử dữ liệu trong cụm C_i và o_i là phần tử được chọn làm trọng tâm của C_i . Giải thuật sẽ lặp đi lặp lại cho đến khi o_i sẽ trùng với trọng tâm của cụm hoặc rất gần trọng tâm của cụm (trong trường hợp tổng quát). Chi tiết hơn về giải thuật k-medoids như sau: ban đầu các phần tử đại diện cho các cụm o_i được chọn ngẫu nhiên. Sau đó gán các phần tử còn lại vào các cụm dựa vào độ tương đồng giữa chúng với o_i . Chọn một phần tử trong cụm o_{random} làm phần tử đại diện của cụm, sau đó kiểm tra từng phần tử p còn lại trong tập dữ liệu và thực hiện các hành động tương ứng với 4 trường hợp sau:

1. Phần tử dữ liệu p hiện tại đang thuộc về cụm j được đại diện bởi o_j . Nếu o_j bị thay thế bởi o_{random} và p lại có độ tương đồng lớn nhất với một phần tử đại diện o_i ($i \neq j$), thì p được gán vào cụm o_i .
2. Phần tử dữ liệu p hiện tại đang thuộc về cụm j được đại diện bởi o_j . Nếu o_j bị thay thế bởi o_{random} và p lại có độ tương đồng lớn nhất với một phần tử đại diện o_{random} , thì p được gán vào cụm o_{random} .
3. Phần tử dữ liệu p hiện tại đang thuộc về cụm i được đại diện bởi o_i . Nếu một phần tử đại diện o_j của cụm j ($i \neq j$) bị thay thế bởi o_{random} và p vẫn có độ tương đồng lớn nhất với o_i , thì p vẫn được gán vào cụm o_i .
4. Phần tử dữ liệu p hiện tại đang thuộc về cụm i được đại diện bởi o_i . Nếu một phần tử đại diện o_j của cụm j ($i \neq j$) bị thay thế bởi o_{random} và p lại có độ tương đồng lớn nhất với một phần tử đại diện o_{random} , thì p được gán vào cụm o_{random} .

Các trường hợp trên được minh họa trên hình 5.2. Giả sử E_t là tổng số lỗi tuyệt đối trước khi chọn o_{random} và E_{t+1} là tổng số lỗi tuyệt đối sau khi chọn o_{random} , giá trị $\Delta = E_{t+1} - E_t$ được gọi là hàm chi phí (cost function). Nếu $\Delta < 0$ tức là tỉ lệ lỗi giảm do đó ta chọn o_{random} để thay thế cho phần tử đại diện trước đó của cụm, ngược lại ($\Delta > 0$) thì phần tử đại diện trước đó của cụm vẫn được giữ nguyên. Quá trình này lặp lại với các phần tử khác được chọn làm o_{random} .

Giải thuật đầu tiên thuộc lớp giải thuật k-mediods là giải thuật *phân vùng quanh trọng tâm* Partition Around Mediods (PAM). Giải thuật PAM được trình bày như sau:

Đầu vào: Tập dữ liệu D , số lượng các cụm k

Đầu ra: Tập dữ liệu đã được phân thành k cụm

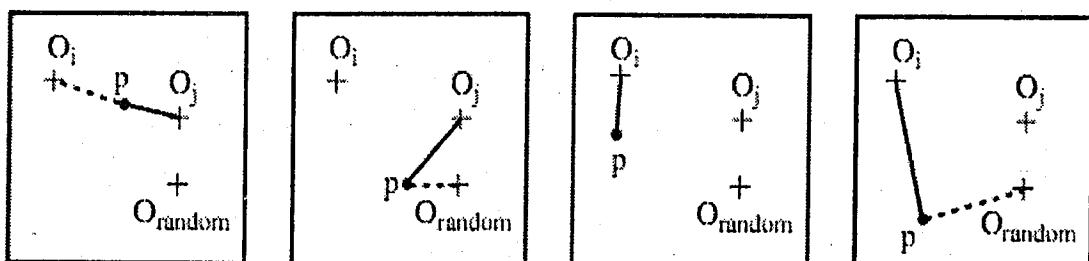
Thuật toán PAM

1. Chọn ngẫu nhiên k phần tử trong D làm phần tử đại diện o_i cho các cụm.

-
2. Phân các phần tử dữ liệu trong D vào các cụm dựa vào độ tương đồng của nó với các o_i . Phần tử dữ liệu sẽ được phân vào cụm có độ tương đồng lớn nhất.
 3. Chọn ngẫu nhiên một phần tử dữ liệu không phải là phần tử đại diện o_{random}
 4. Tính toán hàm chi phí Δ nếu thay thế phần tử đại diện o_i bằng o_{random}
 5. Nếu $\Delta < 0$ thì thay thế o_i bằng o_{random}
 6. Nhảy đến bước 2 cho đến khi quá trình hội tụ (không có sự thay đổi phần tử đại diện).
-

Độ phức tạp của mỗi vòng lặp trong giải thuật PAM là $O(k(n \cdot k)^2)$, do đó có thể dễ dàng nhận ra rằng khi n lớn thì độ phức tạp của giải thuật là rất lớn.

Một số biến thể của giải thuật k-means khác là giải thuật k-modes, hay k-median chúng ta có thể tham khảo thêm ở các tài liệu [ZZ].



Hình 5.2. Các trường hợp xảy ra khi thay thế một phần tử đại diện trong giải thuật PAM

5.3.3. Tìm số lượng cụm thích hợp

Các giải thuật phân cụm phẳng trình bày ở trên cần xác định số lượng các cụm cố định từ trước, tuy nhiên trong nhiều trường hợp ta không thể biết trước được số lượng cụm như thế nào là cho chất lượng tốt nhất. Do vậy rất hữu ích nếu giải thuật cung cấp

cho chúng ta số lượng các cụm như thế nào là tối ưu nhất. Một phương pháp để xác định số lượng cụm tối ưu là dựa vào hàm mục tiêu (objective function) nào đó. Một ví dụ về hàm mục tiêu là hàm giá trị J (công thức (5.9)). Để tìm ra số cụm tối ưu chấp nhận được, ta tìm giá trị cực đại (có thể là cục bộ) của giá trị J . Cho giải thuật k-means thực hiện với các tham số k (số lượng các cụm) khác nhau, giá trị k nào cho giá trị của J cao nhất thì đó là số cụm tối ưu. Tuy nhiên chúng ta cũng phải cân đối với thời gian thực hiện của giải thuật.

Nếu ta chọn hàm mục tiêu chính là hàm tổng số bình phương lỗi E (công thức 5.11), gọi $E(k)$ là giá trị tổng số bình phương lỗi khi phân dữ liệu thành k cụm, giá trị cụm tối ưu sẽ là $\arg \min_k E(k)$. Nhưng ta có thể nhận ra rằng $E(k)$ sẽ có giá trị là 0

khi $k = n$ (n là tổng số phần tử dữ liệu), tức là mỗi cụm sẽ gồm 1 phần tử dữ liệu. Tuy nhiên khi số cụm bằng n thì lại không phải là cái ta cần tìm.

Phương pháp khác để tìm số cụm tối ưu là thêm giá trị phạt (penalty) cho số lượng cụm, khi đó hàm mục tiêu sẽ được tính như sau:

$$k = \arg \min_k [E(k) + \lambda k] \quad (5.13)$$

trong đó, λ là một trọng số, ta có thể thấy giá trị đủ lớn của λ ($\lambda > 0$) ở đây có tác dụng tránh được trường hợp số cụm tối ưu sẽ là n như trường hợp ở trên. Ở đây ta mô hình hóa bài toán phân cụm, trong đó độ phức tạp (complexity) của phân cụm có phụ thuộc vào số lượng cụm (hay một hàm của số lượng cụm). Tuy nhiên ở đây ta lại gặp phải vấn đề là làm sao xác định được giá trị phù hợp cho λ . Một trong những phương pháp xác định λ là dựa vào thực nghiệm và giá trị đó sẽ được dùng cho cùng 1 miền dữ liệu khi tập dữ liệu thay đổi. Ví dụ ta phân cụm dữ liệu thu được từ một tập hợp các phần tử dữ liệu, khi ta xác định được giá trị λ thì giá trị này sẽ được sử dụng trong những lần phân cụm dữ liệu mới (khi nội dung các phần tử dữ liệu từ các website nguồn thay đổi). Chú ý trong trường hợp này, chúng ta chỉ thừa kế giá trị λ chứ không thừa kế số lượng cụm k .

5.4. THUẬT TOÁN PHÂN CỤM PHÂN CẤP

Khác với các giải thuật phân cụm phẳng, các thuật toán phân cụm phân cấp sẽ tạo ra một cây phân cấp các cụm dữ liệu. Các giải thuật phân cụm phân cấp thường được chia làm 2 loại: phân cụm từ dưới lên (lá đến gốc) và từ trên xuống (gốc xuống lá).

5.4.1. Phân cụm phân cấp gộp

Giải thuật đầu tiên chúng ta tìm hiểu là giải thuật phân cụm từ dưới lên có tên là phân cụm gộp (Hierarchical Agglomerative clustering – HAC). Mặc dù có nhiều dạng thức liên quan tới phương pháp phân cụm từ dưới lên, song một tư duy rất tự nhiên để tìm ra các cụm là:

1. Bắt đầu từ mỗi phần tử dữ liệu được coi như một cụm (tại thời điểm này thì số lượng cụm bằng chính số lượng các phần tử dữ liệu);
2. Sau đó từng bước gộp các cụm đã có thành các cụm lớn hơn với yêu cầu phải đảm bảo độ tương tự giữa các phần tử dữ liệu nội bộ trong mỗi cụm cao (số lượng cụm giảm dần);
3. Thuật toán ngừng lại khi hoặc đã đạt được số lượng cụm mong muốn hoặc chỉ còn một cụm duy nhất chứa toàn bộ dữ liệu hay thỏa mãn một điều kiện dừng nào đó.

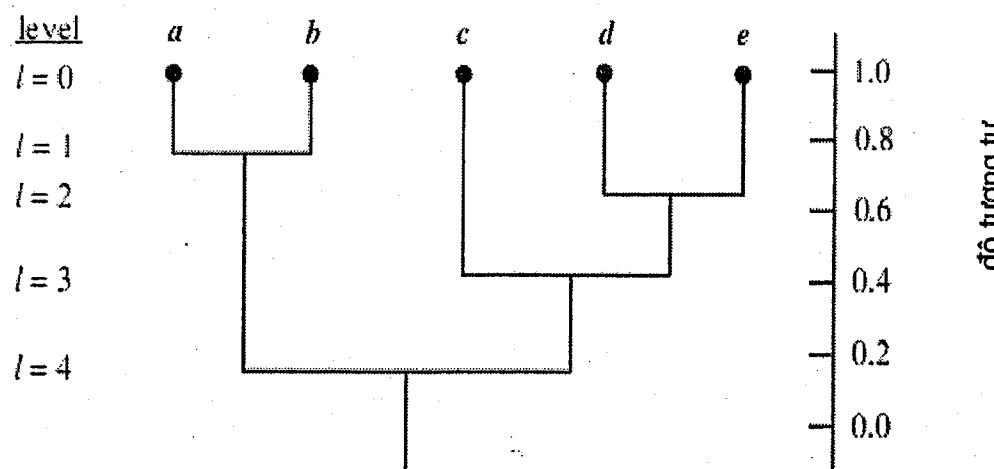
Thuật toán *phân cụm phân cấp gộp* (Hierarchical agglomerative clustering- HAC) là thuật toán phân cụm phân cấp từ dưới lên được sử dụng rất rộng rãi và được tích hợp vào các ứng dụng truy tìm thông tin (Information Retrieval) [Markov07]. HAC chỉ yêu cầu định nghĩa hàm *khoảng cách* giữa các cụm. Ta cũng có thể dùng *độ tương tự* để thay thế độ đo khoảng cách. Chú ý là giá trị của 2 độ đo này là tỉ lệ nghịch với nhau. Nếu dùng độ đo khoảng cách, giả sử C_i và C_j là 2 cụm, có một số phương pháp tính khoảng cách giữa hai cụm C_i và C_j là $d(C_i, C_j)$ như sau:

- *Khoảng cách giữa 2 cụm* được tính là *khoảng cách giữa 2 trọng tâm* của C_i và C_j : $d_{mean}(C_i, C_j) = |m_i - m_j|$, trong đó m_i và m_j lần lượt là trọng tâm của hai cụm C_i và C_j .

- Khoảng cách giữa 2 cụm được tính là khoảng cách cực đại giữa 2 phần tử dữ liệu thuộc vào 2 cụm: $d_{\max}(C_i, C_j) = \max_{p \in C_i, p' \in C_j} |p - p'|$. Giải thuật sử dụng độ đo này còn được gọi là phân cụm người láng giềng gần nhất, và độ đo này còn được gọi là *single-link*.
- Khoảng cách giữa 2 cụm được tính là khoảng cách cực tiểu giữa 2 phần tử dữ liệu thuộc vào 2 cụm: $d_{\min}(C_i, C_j) = \min_{p \in C_i, p' \in C_j} |p - p'|$. Giải thuật sử dụng độ đo này còn được gọi là phân cụm người láng giềng xa nhất, và độ đo này còn được gọi là *complete-link*.
- Khoảng cách giữa 2 cụm được tính là khoảng cách trung bình giữa các tài liệu trong 2 cụm:
- $d_{avg}(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, p' \in C_j} |p - p'|$. Độ đo này còn được gọi là *group-average*.

Tương tự như giải thuật phân cụm khác, mục đích của HAC là làm cực đại độ tương tự giữa các phần tử dữ liệu trong nội tại một cụm. Trong quá trình HAC hoạt động, các cụm được ghép lại với nhau tạo thành một cụm ở cấp cao hơn, độ tương tự nội tại của các cụm mới này sẽ giảm so với các cụm ở cấp thấp hơn trong cây phân cấp (xem minh họa trong hình 5.3).

Như vậy, để đạt được chất lượng phân cụm tổng thể tốt, chúng ta có thể dừng quá trình ghép cụm ở một mức nào đó chứ không bắt buộc phải tạo ra một cụm duy nhất ở gốc của cây phân cấp. Để cài đặt ý tưởng này ta có thể sử dụng các *tham số điều khiển*. Tham số thứ nhất k để dừng thuật toán là khi số lượng cụm mong muốn đã được tạo ra, tham số thứ hai q là dừng thuật toán khi khoảng cách giữa hai cụm được chọn để ghép lớn hơn một ngưỡng nào đó. Gọi G là tập các cụm, D là tập hợp các phần tử dữ liệu cần phân cụm, thuật toán HAC được thể hiện như sau:



Hình 5.3. Một cây phân cấp của thuật toán phân cụm HAC

Đầu vào: + tập dữ liệu không có nhãn D

- + ngưỡng q là giá trị độ tương đồng nhỏ nhất (điều kiện thứ nhất để dừng thuật toán)
- + giá trị k là số lượng cụm mong muốn (điều kiện thứ 2 để dừng thuật toán)

Đầu ra: cây phân cụm phân cấp G

1. $G \leftarrow \{ \{p\} \mid p \in D \}$ (khởi tạo G là tập các cụm chỉ gồm một phần tử dữ liệu trong tập D).
2. Nếu $|G| < k$ thì dừng thuật toán (đã đạt được số lượng cụm mong muốn).
3. Tìm hai cụm $C_i, C_j \in G$ sao cho $(i, j) = \arg \min_{(i,j)} d(C_i, C_j)$ (tìm hai cụm có khoảng cách nhỏ nhất hay độ tương tự lớn nhất).
4. Nếu $d(C_i, C_j) > q$ thì dừng thuật toán (khoảng cách giữa 2 cụm lớn hơn ngưỡng cho phép).
5. Loại bỏ C_i, C_j khỏi G .
6. $G = G \cup \{ C_i, C_j \}$ (ghép hai cụm C_i, C_j và đưa vào trong tập G).
7. Nhảy đến bước 2.

Thuật toán phân cụm phân cấp HAC

Giải thuật có thể dừng tại bước 2 khi số lượng cụm k mong muốn đã thỏa mãn, hay ở bước 4 khi khoảng cách nhỏ nhất giữa 2 cụm là lớn hơn ngưỡng q cho phép. Khi $k = 1$ và $q = 0$ thì G là cây phân cụm hoàn chỉnh có gốc là cụm duy nhất. Khi $k > 1$ thì có k cụm ở mức cao nhất. Một ví dụ về giải thuật phân cụm HAC là cây phân cấp ở hình 5.3. Một điều đáng chú ý đối với thuật toán HAC là nó luôn tạo ra một cây nhị phân chứ không phải là một cây phân cấp tổng quát, vì khi ghép cụm nó chỉ ghép 2 cụm có độ tương tự nhau là lớn nhất.

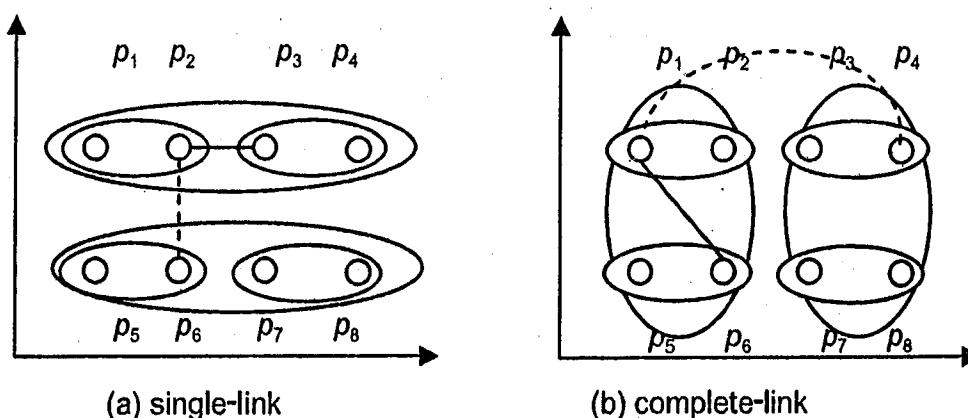
Nhận xét về một số độ đo

Với phân cụm dựa trên độ đo single-link, khoảng cách giữa 2 cụm được tính chính là khoảng cách lớn nhất giữa hai phần tử dữ liệu nằm trong 2 cụm (hình 5.4 a). Do đó khi dùng độ đo này để quyết định ghép 2 cụm lại với nhau thì nó mang tính cục bộ. Vì khi ghép cụm chúng ta chỉ quan tâm đến những vùng dữ liệu mà ở đó có phần tử của 2 cụm gần nhau nhất, mà không cần quan tâm đến các phần tử khác trong cụm cũng như cấu trúc tổng thể của các cụm. Điều này sẽ làm cho chất lượng phân cụm của giải thuật có thể sẽ kém nếu có trường hợp chỉ có duy nhất 2 phần tử dữ liệu ở trong 2 cụm là gần nhau, còn các phần tử dữ liệu còn lại trong 2 cụm ở rất xa nhau.

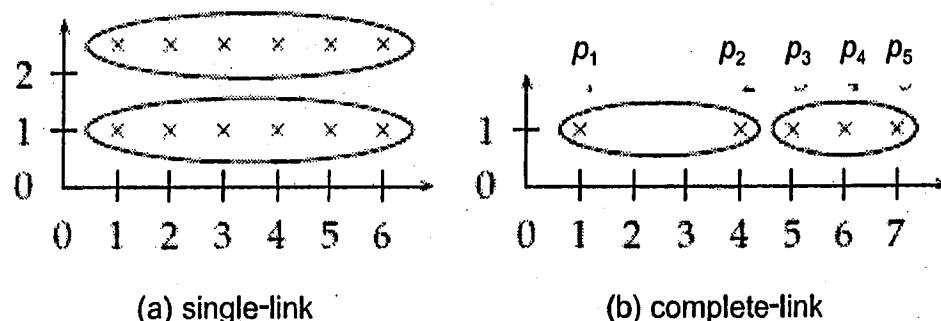
Với phân cụm dựa trên độ đo complete-link, khoảng cách của 2 cụm lại được lấy là khoảng cách của 2 phần tử dữ liệu nằm trong 2 cụm có giá trị nhỏ nhất (hình 5.4 b). Việc này tương đương với việc lựa chọn 2 cụm để ghép lại sẽ tạo ra cụm mới có đường kính nhỏ nhất. Điều kiện lựa chọn dùng để ghép 2 cụm này không mang tính cục bộ, vì cấu trúc toàn cục của các cụm được xem xét trong quá trình quyết định ghép cụm. Điều kiện này có ưu điểm là luôn tạo ra các cụm “cô đọng” vì các cụm mới được tạo ra có bán kính nhỏ nhất. Cũng như phân cụm với single-link, giải thuật phân cụm với complete-link cũng có thể cho chất lượng kém khi có 2 phần tử dữ liệu trong 2 cụm ở rất xa nhau trong khi trọng tâm của 2 cụm này lại rất gần nhau, khi đó 2 cụm này có thể không được lựa chọn để ghép lại với nhau.

Hình 5.4 minh họa phân cụm phân cấp HAC với độ đo single-link (a) và độ đo complete-link (b) trên 8 phần tử dữ liệu $\{p_1, p_2, \dots, p_8\}$. Từ hình minh họa cho thấy bốn bước đầu tiên của cả 2 giải thuật đều tạo ra các cụm giống nhau. Bước thứ 5, giải thuật HAC với single-link sẽ ghép 2 cụm ở phía trên lại với nhau, và bước thứ 7 là ghép 2 cụm ở dưới lại. Trong khi đó giải thuật HAC với complete-link lại ghép 2 cụm ở phía bên trái ở bước thứ 6 và ghép hai cụm phía bên phải lại ở bước thứ 5.

Cả hai độ đo single-link và complete-link đều đánh giá khoảng cách của 2 cụm dựa trên một cặp phần tử dữ liệu duy nhất, do đó giải thuật phân cụm sử dụng các độ đo này đều có khả năng tạo ra các cụm không mong muốn (có chất lượng không tốt). Hình 5.5 a đưa ra ví dụ một trường hợp mà thuật toán HAC với độ đo single-link cho kết quả không mong muốn. Vì điều kiện ghép cụm của độ đo này mang tính cục bộ mà không quan tâm đến hình dáng của cụm được tạo ra.



Hình 5.4. Phân cụm với độ đo single-link và complete-link



Hình 5.5. Trường hợp ghép cụm không tốt của độ đo single-link vào complete-link

Do đó nó đã tạo ra một cụm có hình như một chuỗi (chain). Nếu ta để ý thì có thể nhận ra tình huống tạo chuỗi với độ đo single-link cũng xuất hiện ngay trong hình 5.4a. Nhưng giải thuật phân cụm HAC với độ đo complete-link với cùng tập dữ liệu này lại không tạo chuỗi (hình 5.4 b), do đó kết quả các cụm tạo ra trong trường hợp này là tốt hơn.

Còn giải thuật HAC với độ đo complete-link lại có nhược điểm khác, đó là khi ghép cụm lại với nhau nó lại quan tâm nhiều đến trường hợp ngoại lệ của 2 phần tử dữ liệu trong 2 cụm có khoảng cách nhau là thấp nhất mà không quan tâm đến các phần tử dữ liệu còn lại trong cụm, hay cấu trúc toàn cục của các cụm. Do đó nó có thể tạo ra các cụm không mong muốn như minh họa trong hình 5.5 b. Một cách trực quan, nếu ta quan tâm đến cấu trúc của dữ liệu thì kết quả phân cụm ở mức gần gốc nên là 2 cụm $\{p_1\}$ và $\{p_2, p_3, p_4, p_5\}$, thì tốt hơn nhiều so với 2 cụm $\{p_1, p_2\}$ và $\{p_3, p_4, p_5\}$.

Độ đo group-average tính toán khoảng cách của 2 cụm dựa trên khoảng cách của toàn bộ các cặp phần tử dữ liệu trong 2 cụm chứ không chỉ dựa trên một cặp phần tử dữ liệu duy nhất. Do đó nó tránh được các trường hợp không mong muốn như 2 độ đo vừa thảo luận ở trên.

Độ đo dựa vào trọng tâm cũng có đặc điểm là không dựa trên một cặp phần tử dữ liệu để quyết định khoảng cách của 2 cụm. Ở đây giá trị của khoảng cách giữa 2 cụm chính là khoảng cách của trọng tâm của 2 cụm. Độ đo này tránh được một số nhược điểm của độ đo single-link và complete-link, tuy nhiên nó cũng có nhược điểm là khoảng cách từ dưới lên trên cây phân cấp có thể là không giảm dần (do trọng tâm của các cụm ở mức cao có thể ở gần nhau hơn so với các cụm ở mức dưới). Điều này trái ngược với giả thiết cơ bản là các cụm nhỏ thường có *độ kết dính* (coherent) cao hơn các cụm có kích thước lớn hơn.

5.4.2. Các thuật phân cụm phân cấp BIRCH

Giải thuật phân cụm phân cấp tiếp theo là BIRCH được viết tắt từ cụm từ Balanced Iterative Reducing Clustering Using Hierarchies.

BIRCH được thiết kế để giải quyết các bài toán có số lượng dữ liệu lớn bằng cách kết hợp phân cụm phân cấp trong bước phân cụm vi mô (micorclustering stage), với các phương pháp phân cụm khác (chẳng hạn phân cụm phẳng trong bước phân cụm vĩ mô (macroclustering stage). Nó giải quyết được nhược điểm của các phương pháp phân cụm phân cấp là: (1) tính khả cỡ (scalability) – khả năng làm việc với một tập dữ liệu rất lớn; và (2) khả năng không thay đổi được khi đã gán một phần tử dữ liệu vào một cụm.

Có 2 khái niệm (hay 2 cấu trúc dữ liệu) được đề cập trong giải thuật BIRCH là *đặc trưng phân cụm* (clustering feature) ký hiệu là CF; và *cây đặc trưng phân cụm* (clustering feature tree) ký hiệu là CF tree. Việc đề xuất ra hai cấu trúc dữ liệu này đã làm cho giải thuật BIRCH có tốc độ khá nhanh và có thể xử lý được một lượng dữ liệu lớn, đặc biệt là nó tạo khả năng phân cụm các dữ liệu phát sinh (thêm mới) một cách đơn giản mà không cần phải phân cụm lại toàn bộ tập dữ liệu. Nói một cách khác nó có khả năng xử lý dữ liệu một cách gia tăng (incremental). Cho một tập n phần tử dữ liệu trong một cụm, khi đó trọng tâm của cụm x_0 , bán kính R và đường kính D được định nghĩa như sau:

$$x_0 = \frac{\sum_{i=1}^n x_i}{n}, R = \sqrt{\frac{\sum_{i=1}^n (x_i - x_0)^2}{n}} \text{ và } D = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^n (x_i - x_j)^2}{n(n-1)}} \quad (5.14)$$

trong đó, x_i là một phần tử dữ liệu trong cụm; R là khoảng cách trung bình từ các phần tử dữ liệu đến trọng tâm của cụm; D là khoảng cách trung bình của tất cả các cặp phần tử dữ liệu trong cùng một cụm. Hai giá trị R và D thể hiện độ co cụm (tightness) của các phần tử dữ liệu quanh trọng tâm của nó. Đặc trưng phân cụm CF là một bộ ba chứa thông tin tóm tắt về một cụm. Cho một tập n phần tử dữ liệu $\{x_i\}$ trong một cụm, khi đó giá trị CF của cụm được định nghĩa như sau:

$$CF = \langle n, LS, SS \rangle, LS = \sum_{i=1}^n x_i \text{ và } SS = \sum_{i=1}^n x_i^2 \quad (5.15)$$

trong đó, n là số lượng các phần tử trong cụm; LS là tổng tuyến tính của n phần tử dữ liệu và SS là tổng bình phương các phần tử trong cụm.

Ví dụ, cụm dữ liệu C_1 có 3 phần tử dữ liệu $\{(2,5), (3,2), (4,3)\}$ thì đặc trưng phân cụm của nó là

$$\begin{aligned} CF_1 &= \langle 3, (2+3+4, 5+2+3), (2^2+3^2+4^2, 5^2+2^2+3^2) \rangle \\ &= \langle 3, (9,10), (29,38) \rangle \end{aligned}$$

Bản chất của CF là chứa thông tin thống kê của một cụm. Một đặc điểm quan trọng của CF là nó có tính cộng dồn (additive). Giả sử chúng ta có 2 cụm không giao nhau C_1 và C_2 có giá trị đặc trưng phân cụm tương ứng là CF_1 và CF_2 , nếu ta gộp 2 cụm này thành một cụm lớn hơn thì đặc trưng phân cụm của cụm được tạo ra sẽ chính bằng $CF_1 + CF_2$ (chứ ta không phải tính lại giá trị CF cho cụm mới tạo thành dựa trên các phần tử dữ liệu của nó). Đây là đặc điểm cực kỳ quan trọng nó cho phép BIRCH không cần lưu các phần tử dữ liệu của từng cụm mà vẫn tính toán ra được các độ đo cần thiết.

Ví dụ, giả sử đặc trưng phân cụm của cụm C_2 là

$$CF_2 = \langle 3, (35, 36), (417, 440) \rangle,$$

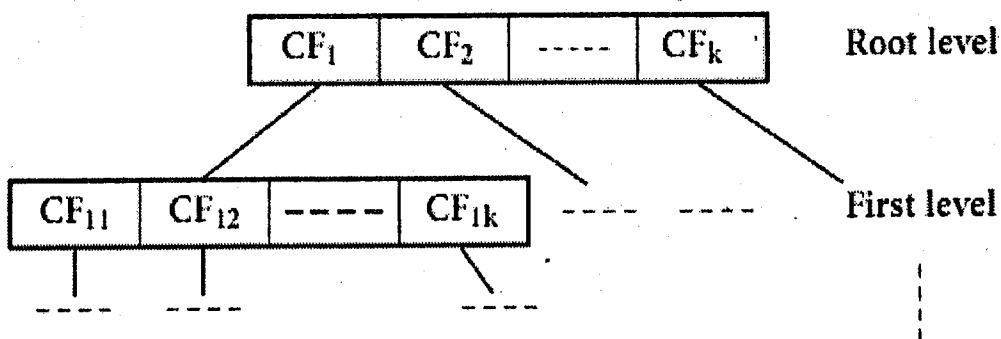
nếu ta ghép cụm C_1 và C_2 thành cụm C_3 , khi đó đặc trưng phân cụm của C_3 sẽ là:

$$\begin{aligned} CF_3 &= \langle 3+3, (9+35, 10+36), (29+417, 38+440) \rangle \\ &= \langle 6, (44, 46), (446, 478) \rangle \end{aligned}$$

Giải thuật phân cụm BIRCH chỉ cần dùng đặc trưng phân cụm để tính toán các độ đo cần thiết cho việc quyết định phân cụm dữ liệu. Nói một cách khác, BIRCH không cần lưu chi tiết từng phần tử dữ liệu đầu vào, do đó nó sử dụng rất ít bộ nhớ so với các giải thuật phân cụm ở trên.

Cấu trúc dữ liệu tiếp theo là cây đặc trưng phân cụm, nó là một cây cân bằng (height-balanced tree) chứa các đặc trưng phân cụm. Mỗi nút (không phải là nút lá) của cây sẽ có các nút con, và giá trị CF của nó sẽ được tính bằng tổng các giá trị đặc trưng phân cụm của các nút con của nó. Chúng ta có thể thấy cây này cũng

chứa luôn các cụm hay nó chính là cây phân cụm phân cấp. Ngoài ra mỗi một cây đặc trưng phân cụm còn có thêm 2 tham số: (1) hệ số phân nhánh (branching factor) B và (2) một ngưỡng T . Hệ số rẽ nhánh xác định số lượng con tối đa mà một nút (không phải là nút lá) có thể có. Ngưỡng T xác định đường kính tối đa của các cụm tại các nút lá. Hai tham số này sẽ ảnh hưởng lớn đến kích thước của cây phân cụm phân cấp đầu ra. Nếu ngưỡng T lớn thì số lượng cụm sẽ giảm và kích thước cây sẽ nhỏ và ngược lại.



Hình 5.6. Minh họa một cây đặc trưng phân cụm

BIRCH sẽ cố gắng tạo ra một tập các cụm tốt nhất dựa vào tài nguyên hữu hạn bộ nhớ và làm giảm thiểu các thao tác vào ra. BIRCH sử dụng kỹ thuật phân cụm nhiều pha (multiphase), cụ thể nó có 2 pha chính sau:

- **Pha 1:** BIRCH duyệt toàn bộ dữ liệu để xây dựng cây đặc trưng phân cụm CF tree ban đầu. Quá trình này có thể được coi là bước nén dữ liệu ở nhiều mức nhưng vẫn giữ tính chất phân bố thành cụm của dữ liệu. Pha này được gọi là pha phân cụm vi mô (microclustering), nó tạo ra các vi cụm (microcluster) là các nút lá.
 - **Pha 2:** BIRCH áp dụng giải thuật phân cụm tại các nút lá của cây CF tree, các cụm có cấu trúc thừa sẽ bị coi là ngoại lệ và bị bỏ đi, các cụm có mật độ dày sẽ được ghép với nhau tạo thành cụm lớn hơn. Pha này còn được gọi là pha phân cụm vĩ mô (macroclustering), nó xử lý trên toàn cây CF tree.

Tại pha 1, cây CF tree được xây dựng động, các phần tử dữ liệu sẽ được lần lượt chèn vào nút lá gần nó nhất. Quá trình này ta

thấy giải thuật hoạt động theo cơ chế gia tăng (incremental). Nếu nút lá sau khi chèn thêm dữ liệu có đường kính lớn hơn ngưỡng T , thì nút lá đó sẽ bị chia thành các cụm nhỏ hơn, hay một nút lá mới sẽ được tạo ra. Sau khi thực hiện xong thao tác chèn một phần tử dữ liệu, thông tin về nó sẽ được cập nhật ngược lên cho đến tận nút gốc. Quá trình này lại cho ta thấy khi có dữ liệu mới thì cây CF tree sẽ được cập nhật chứ không phải xây dựng lại từ đầu. Chú ý rằng tham số ngưỡng T có ảnh hưởng đến kích thước của cây CF tree, do đó trong trường hợp kích thước của cây CF tree lớn hơn kích thước bộ nhớ trong thì ta có thể điều chỉnh lại giá trị ngưỡng T này (chọn giá trị lớn hơn giá trị hiện tại của T), sau đó xây dựng lại cây. Quá trình xây dựng lại cây được thực hiện từ các nút lá của cây cũ, do đó ta không cần phải đọc lại dữ liệu. Lý do của việc ta có thể xây dựng lại cây mới từ cây cũ là: ở cây mới có ngưỡng T lớn hơn, nên kích thước của một cụm sẽ to hơn, dẫn đến việc tạo cây mới chỉ cần thao tác gộp các cụm lại với nhau. Quá trình này cũng có thể so sánh với quá trình thêm một nút và phân chia một nút trong giải thuật xây dựng cây B+. Như vậy chúng ta chỉ cần một lần đọc toàn bộ dữ liệu để xây dựng cây CF tree. Đây là giải pháp cho phép giải thuật hoạt động được trong điều kiện bộ nhớ trong là hữu hạn đồng thời vẫn hạn chế được số lượng các thao tác vào ra. Một số phương pháp được giới thiệu để loại bỏ các dữ liệu ngoại lệ, khi đó ta có thể cần phải duyệt dữ liệu một lần nữa. Chúng ta chú ý rằng có 2 trường hợp dữ liệu có thể được gán lại: chia nhỏ một nút lá thành các nút con hoặc xây dựng lại cây. Đây là một trong những ưu điểm của BIRCH, nó khắc phục được đặc điểm là sau khi phân dữ liệu vào một cụm thì ta không thể phân lại được của giải thuật HAC được trình bày ở trên.

Sau khi ta đã xây dựng được cây, thì ta có thể sử dụng bất kỳ giải thuật phân cụm phân cấp nào để xử lý dữ liệu trên cây CF tree. Một trong những giải thuật ta có thể sử dụng được là giải thuật HAC.

Độ phức tạp của thuật toán khi xây dựng cây là $O(n)$, thực nghiệm đã cho thấy BIRCH cho kết quả phân cụm có chất lượng khá tốt và thời gian xử lý nhanh. Tuy nhiên BIRCH cũng gặp phải

nhiều điểm giống các giải thuật phân cụm dựa vào độ đo khoảng cách. Đó là nếu các cụm không được phân bố theo hình cầu thì kết quả phân cụm của BIRCH là không tốt. Hơn nữa vì số lượng các cụm trong cây CF tree là hữu hạn (do hạn chế về bộ nhớ) nên có thể kết quả phân cụm của nó sẽ không phản ánh đúng phân bố tự nhiên của các cụm.

5.4.3. Thuật toán phân cụm phân cấp từ trên xuống DIANA

Theo các nghiên cứu được công bố, kỹ thuật phân cụm từ dưới lên (bottom-up) được sử dụng trực tiếp tồn thời gian với độ phức tạp là $O(n^2)$ và không thích hợp cho các tập dữ liệu lớn. Nếu coi như đặt trước số cụm là k , kỹ thuật phân hoạch *từ trên xuống* (top-down) thường được sử dụng vì hiệu quả hơn. Một kỹ thuật đi theo hướng này là sử dụng thuật toán k-means. Thuật toán bắt đầu từ đỉnh của cây với chỉ có một cụm là toàn bộ các phần tử dữ liệu. Cụm này sẽ được chia thành các cụm nhỏ hơn sử dụng thuật toán phân cụm phẳng (chẳng hạn như k-means). Với các cụm nhỏ ta lại áp dụng đệ quy thuật toán phân cụm phẳng. Về lý thuyết thì thuật toán phân cụm phân cấp từ trên xuống phức tạp hơn so với phương pháp phân cụm từ dưới lên vì chúng ta gọi giải thuật phân cụm phẳng (như là một thủ tục) nhiều lần. Tuy nhiên nó có ưu điểm trong trường hợp chúng ta không cần thiết phải sinh ra một cây phân cấp hoàn chỉnh (cây có các cụm ở nút lá chỉ chứa đúng một phần tử dữ liệu). Khi giới hạn số lượng mức (level) của cây phân cấp, và kết hợp sử dụng giải thuật phân cụm phẳng k-means, thuật toán phân cụm phân cấp từ trên xuống có độ phức tạp gần như là tuyến tính với số lượng các phần tử dữ liệu và số lượng các cụm. Do đó thuật toán phân cụm từ trên xuống sẽ chạy nhanh hơn so với thuật toán phân cụm từ dưới lên HAC.

Giải thuật phân cụm từ trên xuống còn được chứng minh là có độ chính xác cao hơn so với các giải thuật phân cụm từ dưới lên như HAC trong một số trường hợp. Lý do là giải thuật phân cụm từ dưới lên đưa ra quyết định ghép các cụm lại với nhau chỉ sử dụng các thông tin cục bộ (ở các cụm) mà không thể dựa trên

thông tin toàn cục (tất cả dữ liệu). Và các cụm sau khi ghép rồi thì không thể tách ra để ghép với các cụm khác. Ngược lại các giải thuật phân cụm từ trên xuống ngay từ đầu đã khai thác được thông tin toàn cục (phân bố toàn cục của tập dữ liệu) khi quyết định phân dữ liệu đang xét thành các cụm nhỏ hơn.

Để minh họa rõ hơn cách làm việc của giải thuật phân cụm từ trên xuống, mục này sẽ trình bày chi tiết giải thuật DIANA (viết tắt từ cụm từ DIvisive ANAlysis). Giải thuật này có cách hoạt động rất giống với giải thuật HAC, tuy nhiên điểm khác biệt là nó hoạt động từ trên xuống. Chi tiết về thuật toán được mô tả như sau:

Đầu vào: tập D gồm n phần tử dữ liệu $\{x_1, x_2, \dots, x_n\}$

Đầu ra: cây phân cụm phân cấp

Thuật toán DIANA

Bước khởi tạo: Tạo cụm ban đầu gồm toàn bộ tập dữ liệu $D \{x_1, x_2, \dots, x_n\}$

Ở các vòng lặp sau, cụm lớn nhất sẽ được chọn để chia thành 2 cụm nhỏ hơn. Quá trình này lặp lại cho đến khi mỗi cụm chỉ chứa 1 phần tử dữ liệu (quá trình này sẽ được thực hiện trong $n-1$ bước), hoặc thỏa mãn 1 điều kiện dừng nào đó. Chú ý rằng, giả sử một cụm có n phần tử thì chúng ta có tổ hợp $2^{n-1} - 1$ cách để chia cụm này thành 2 cụm con. Đây là một tổ hợp rất lớn, do đó để giảm độ phức tạp (tránh phải xét toàn bộ tổ hợp), giải thuật DIANA sử dụng phương pháp chia cụm như sau:

Bước chia cụm:

1. Với cụm đang được chọn để chia, tìm phần tử dữ liệu khác biệt hẳn với các phần tử còn lại trong cụm. Tạo một cụm mới chứa phần tử khác biệt này, gọi là *cụm khác biệt* (splinter group) S .
2. Với từng phần tử dữ liệu x_i không thuộc vào tập S ($x_i \notin S$), tính giá trị d_i là trung bình khoảng cách giữa x_i với các phần tử không thuộc S trừ đi trung bình khoảng cách giữa x_i với các phần tử thuộc S :

$$2.1. d_i = \text{average}(\sum_{x_j \in S} |x_i - x_j|) - \text{average}(\sum_{x_j \notin S} |x_i - x_j|) \quad (5.16)$$

2.2. Tìm phần tử dữ liệu x_h sao cho d_h có giá trị lớn nhất. Nếu $d_h > 0$ thì thêm x_h vào tập S . Điều này có nghĩa là tìm phần tử gần với cụm S hơn so với phần còn lại để thêm vào trong S .

3. Lặp lại bước 2 cho đến khi không còn phần tử nào có $d_i > 0$. Tại thời điểm này thì cụm đã được chia thành 2 cụm con.

4. Chọn cụm có đường kính d lớn nhất $d = \sqrt{\frac{\sum_{i=1}^m \sum_{j=1}^m (x_i - x_j)^2}{m(m-1)}}$,

trong đó m là số lượng các phần tử trong cụm. Lặp lại các bước 1 đến 3 để chia cụm này thành 2 cụm nhỏ hơn

5. Lặp lại bước 4 cho đến khi mỗi cụm chỉ chứa một phần tử dữ liệu hay một điều kiện dừng nào đó xảy ra. Một ví dụ về điều kiện dừng là tổng số lượng cụm đã tạo ra vượt một ngưỡng k nào đó.

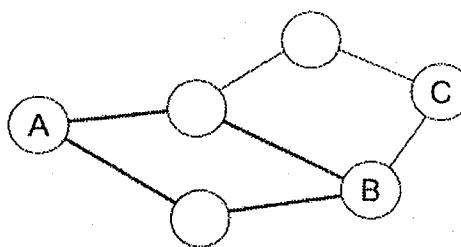
So sánh với giải thuật HAC thì giải thuật DIANA có điểm giống là tại mỗi bước nó chỉ tách một cụm ra làm 2 cụm nhỏ hơn (giải thuật HAC thì mỗi bước chỉ gộp 2 cụm thành 1 cụm lớn hơn). Do đó cây kết quả đầu ra của 2 giải thuật đều có dạng cây nhị phân.

5.4.4. Thuật toán phân cụm phân cấp ROCK

Việc phân cụm dựa vào khoảng cách (hay độ tương tự) là có một số nhược điểm như đã trình bày trong mục 5.3.1 khi nhận xét về một số độ đo. Đặc biệt là khi chúng ta phân cụm dữ liệu được biểu diễn bằng các giá trị rời rạc (hay bằng mô hình boolean) thì thực nghiệm đã chứng minh được rằng các độ đo khoảng cách cho các cụm có chất lượng không cao. Để minh họa cho trường hợp này ta xem một ví dụ sau: giả sử trong một siêu thị có 5 mặt hàng $\langle a, b, c, d, e, f \rangle$ và cơ sở dữ liệu biểu diễn các giao dịch (giỗ mua hàng)

được biểu diễn dưới dạng rời rạc 1(hay *true*) hoặc 0 (hay *false*) thể hiện các mặt hàng có được mua hay không. Xét 3 giao dịch $A = (1, 0, 0, 0, 0)$, $B = (0, 0, 0, 0, 1)$ và $C = (1, 1, 1, 1, 0)$. Nếu dùng độ đo khoảng cách $d = \|p - p'\| = \sqrt{\sum_{i=1}^n (p_i - p'_i)^2}$ (trong đó n là số chiều của vector biểu diễn dữ liệu) để phân cụm thì $\|A - B\| = \sqrt{2}$, $\|A - C\| = \sqrt{3}$ và $\|B - C\| = \sqrt{5}$. Dựa vào các giá trị này thì hai giao dịch A và B (có khoảng cách nhỏ nhất) sẽ được gộp vào thành 1 cụm, đây là trường hợp gộp sai vì A và B không hề có chung một mặt hàng nào, chỉ có A và C mới có chung mặt hàng a .

Giải thuật có tên là ROCK (viết tắt từ cụm từ RObust Clustering using linKs) đã được đề xuất để xử lý dữ liệu rời rạc. ROCK là một giải thuật phân cụm phân cấp, nó khai thác khái niệm liên kết (link) để thực hiện quá trình phân cụm. Ở đây, một liên kết là một phần tử láng giềng chung (common neighbor) giữa một cặp 2 phần tử dữ liệu. Nếu một cặp phần tử dữ liệu tương tự nhau và chúng lại có chung một số lượng lớn các phần tử láng giềng thì nó có khả năng cùng thuộc về một cụm, do đó ta có thể gộp chúng lại với nhau vào cùng 1 cụm. Đây là điểm khác biệt lớn giữa giải thuật phân cụm dựa trên khoảng cách (hay độ tương đồng) với ROCK. Khi dựa trên khoảng cách (hay độ tương đồng) để phục vụ cho quyết định phân cụm, ta chỉ sử dụng thông tin từ chính phần tử dữ liệu đó (thông tin cục bộ). Còn ROCK có sử dụng thông tin mang tính toàn cục hơn vì nó có quan tâm đến các phần tử láng giềng. Tuy nhiên không phải phần tử nào cũng có thể là phần tử láng giềng của một phần tử dữ liệu cụ thể nào đó. Phần tử dữ liệu p_i được gọi là láng giềng của p_j nếu $sim(p_i, p_j) > \theta$, trong đó $sim(p_i, p_j)$ là hàm đo độ tương tự giữa 2 phần tử và θ là một ngưỡng cho trước. Hàm $sim(p_i, p_j)$ có thể chọn là hàm dựa trên khoảng cách hay có thể là một hàm cung cấp bởi chuyên gia trong lĩnh vực cụ thể miễn là đảm bảo thuộc tính: hàm $sim(p_i, p_j)$ này có giá trị lớn thì cặp (p_i, p_j) càng tương tự nhau, và giá trị của hàm $sim(p_i, p_j)$ phải được chuẩn hóa nằm trong khoảng $[0, 1]$. Khi $sim(p_i, p_j) = 1$ thì p_i trùng với p_j và khi $sim(p_i, p_j) = 0$ thì p_i hoàn toàn khác p_j .



Hình 5.7. Minh họa khái niệm liên kết trong ROCK

Nếu biểu diễn mỗi phần tử dữ là một đỉnh, các phần tử là láng giềng của nhau được nối với nhau bằng 1 cạnh, khi đó ta có thể biểu diễn tập dữ liệu đầu vào dưới dạng một đồ thị như minh họa trên hình 5.7. Khi đó một liên kết giữa 2 phần tử dữ liệu là một đường đi có chiều dài là 2 (trong đồ thị) từ đỉnh mô tả phần tử này sang đỉnh mô tả phần tử kia. Ví dụ trong hình 5.7 số lượng liên kết giữa 2 đỉnh A và B là 2.

Một ví dụ về các *giỏ mua hàng* (market basket) trong siêu thị để minh họa hiệu năng của giải thuật ROCK so với giải thuật phân cụm dựa trên độ tương tự trên dữ liệu rời rạc ta tìm hiểu một bài toán cụ thể sau. Giả sử một siêu thị có các mặt hàng $\langle a, b, c, \dots, g \rangle$. Các giao dịch đã được phân thành 2 cụm có chất lượng cao là C_1 và C_2 , trong đó cụm C_1 chứa các giao dịch $\{a, b, c\}$, $\{a, b, d\}$, $\{a, b, e\}$, $\{a, c, d\}$, $\{a, c, e\}$, $\{a, d, e\}$, $\{b, c, d\}$, $\{b, c, e\}$, $\{b, d, e\}$, và $\{c, d, e\}$. Như vậy cụm C_1 có chứa các mặt hàng $\langle a, b, c, d, e \rangle$. Cụm C_2 chứa các giao dịch $\{a, b, f\}$, $\{a, b, g\}$, $\{a, f, g\}$, và $\{b, f, g\}$. Như vậy cụm C_2 chứa các mặt hàng $\langle a, b, f, g \rangle$. Để đo độ tương tự giữa các phần tử dữ liệu được biểu diễn bằng các giá trị rời rạc ta có thể sử dụng hệ số Jaccard (Jaccard efficient) được tính bằng công thức sau:

$$\text{sim}(T_i, T_j) = \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \quad (5.17)$$

Ban đầu ta giả sử chỉ sử dụng độ đo tương tự để phân cụm, khi đó hệ số Jaccard giữa hai giao dịch $\{a, b, c\}$ và $\{b, d, e\}$ nằm trong cụm C_1 là:

$$\frac{|\{a, b, c\} \cap \{b, d, e\}|}{|\{a, b, c\} \cup \{b, d, e\}|} = \frac{|\{b\}|}{|\{a, b, c, d, e\}|} = \frac{1}{5}$$

Nếu ta tính cho từng cặp giao dịch trong C_1 thì sẽ thấy hệ số Jaccard của chúng sẽ nằm trong khoảng từ $1/5$ đến $1/2$ (ví dụ trường hợp hệ số Jaccard = $1/2$ là cặp giao dịch $\{a, b, c\}$ và $\{a, b, d\}$). Đặc biệt là nếu so sánh các cặp giao dịch giữa cụm C_1 và C_2 thì cũng có trường hợp ta thu được hệ số Jaccard là $1/2$, chẳng hạn như giao dịch $\{a, b, c\}$ của cụm C_1 với giao dịch $\{a, b, f\}$ hay $\{a, b, g\}$ của cụm C_2 . Điều này chứng tỏ rằng nếu sử dụng độ đo tương tự thì không thể tạo ra được 2 cụm C_1 và C_2 như trên.

Bây giờ nếu ta sử dụng khái niệm liên kết và khái niệm phần tử láng giềng trong giải thuật ROCK. Hai phần tử dữ liệu p_i và p_j là láng giềng của nhau nếu thỏa mãn điều kiện $sim(p_i, p_j) > \theta$. Trong trường hợp này giả sử ngưỡng θ là $0,5$, xét hai giao dịch $\{a, b, f\}$ và $\{a, b, g\}$ trong cụm C_2 , ta dễ dàng nhận ra được giao dịch $\{a, b, f\}$ có tập các láng giềng là:

$$\{\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, b, g\}, \{a, f, g\}, \{b, f, g\}\}$$

Giao dịch $\{a, b, g\}$ có các láng giềng là:

$$\{\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, b, f\}, \{a, f, g\}, \{b, f, g\}\}$$

Do đó, cặp hai giao dịch $\{a, b, f\}$ và $\{a, b, g\}$ có chung các láng giềng:

$$\{\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, f, g\}, \{b, f, g\}\}$$

Hay số lượng liên kết giữa 2 giao dịch này là 5 , như vậy ta có thể kết luận là 2 giao dịch này thuộc về cùng một cụm. Tương tự cho các giao dịch còn lại trong C_2 ta cũng dễ dàng nhận ra chúng đều có chung các láng giềng. Nếu so sánh các cặp giao dịch giữa 2 cụm C_1 và C_2 thì ta sẽ thấy số lượng các liên kết giữa chúng là nhỏ. Ví dụ xét cặp giao dịch $\{a, b, f\}$ trong cụm C_2 và $\{a, b, c\}$ trong cụm C_1 . Giao dịch $\{a, b, c\}$ có các láng giềng $\{\{a, b, d\}, \{a, b, e\}, \{a, b, f\}, \{a, b, g\}\}$. Cặp giao dịch này có các láng giềng chung là: $\{\{a, b, d\}, \{a, b, e\}, \{a, b, g\}\}$, hay số lượng liên kết giữa cặp giao dịch này là 3 nhỏ hơn số lượng liên kết giữa cặp giao dịch $\{a, b, f\}$ và $\{a, b, g\}$ do đó nó không được phân vào trong cùng một cụm với $\{a, b, f\}$.

Tương tự giao dịch $\{a, f, g\}$ trong cụm C_2 đều có 2 liên kết với các phần tử trong C_2 nhưng nó lại không có liên kết nào với các

giao dịch trong C_1 . Hay việc phân giao dịch này vào cụm C_2 là hoàn toàn hợp lý.

Dựa trên khái niệm liên kết, với một ngưỡng θ (dùng để xác định các phần tử láng giềng), ta định nghĩa hàm $link(p, p')$ là số lượng liên kết giữa 2 phần tử dữ liệu p và p' . Tương tự giống các giải thuật phân cụm đã được giới thiệu ở trên, ta cần có một hàm để đánh giá chất lượng của các cụm kết quả. Mục tiêu của giải thuật ROCK là phân dữ liệu vào các cụm sao cho số lượng liên kết giữa các phần tử trong cùng một cụm là cao, và số lượng liên kết giữa các phần tử nằm trong các cụm khác nhau là nhỏ, do đó một trong những hàm điều kiện có thể dùng là:

$$E_i = \sum_{i=1}^k \sum_{p, p' \in C_i} link(p, p') \quad (5.18)$$

trong đó, k là số lượng cụm; C_i là cụm thứ i ; Tuy nhiên hàm điều kiện trên chỉ có đảm bảo các phần tử dữ liệu có số lượng liên kết lớn sẽ được ghép vào một cụm, chứ không có khả năng ngăn chặn việc phân toàn bộ các phần tử dữ liệu vào cùng một cụm. Do đó hàm điều kiện khác đã được đề xuất như sau:

$$E_i = \sum_{i=1}^k n_i * \sum_{p, p' \in C_i} \frac{link(p, p')}{n_i^{1+2f(\theta)}} \quad (5.19)$$

trong đó, n_i là kích thước của cụm C_i (số lượng phần tử dữ liệu trong C_i); và $f(\theta)$ là hàm phụ thuộc vào miền dữ liệu và kiểu cụm ta muốn quan tâm. Với công thức (5.19) ta có thể dễ dàng nhận ra khi kích thước của cụm C_i tăng lên thì mẫu số trong công thức trên sẽ tăng nhanh, do đó nó dẫn đến $E_i = \sum_{p, p' \in C_i} \frac{link(p, p')}{n_i^{1+2f(\theta)}}$ sẽ có

giá trị nhỏ. Hay nói cách khác công thức (5.19) sẽ ngăn được việc giải thuật có thể gán quá nhiều phần tử dữ liệu vào một cụm. Trong thực tế việc xác định hàm $f(\theta)$ là công việc khó khăn, trong miền dữ liệu giao dịch các giỏ mua hàng thì người ta tìm được $f(\theta) = \frac{1-\theta}{1+\theta}$.

Hoạt động của giải thuật ROCK được mô tả sơ lược như sau:

1. Xây dựng đồ thị biểu diễn các phần tử dữ liệu dựa trên khái niệm láng giềng (với một độ đo tương tự và ngưỡng θ cho trước).
2. Áp dụng giải thuật phân cụm phân cấp gộp HAC (agglomerative hierarchical clustering) trên đồ thị được xây dựng trong bước 1.

Trong giải thuật HAC thì nó cần xác định được 2 cụm có độ tương tự nhau lớn nhất để gộp lại với nhau và tất nhiên chúng ta không thể sử dụng độ tương tự giữa 2 cụm dựa trên khoảng cách được. Trong giải thuật ROCK hàm đo độ tương tự giữa 2 cụm cần phải làm cực đại hóa hàm điều kiện (công thức 5.19), do đó công thức tính độ tương tự cũng được xây dựng giống như hàm điều kiện như sau:

$$g(C_i, C_j) = \frac{\text{link}(C_i, C_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \quad (5.20)$$

trong đó, n_i và n_j là kích thước của cụm C_i và C_j ; hàm $\text{link}(C_i, C_j)$ đo số lượng liên kết giữa 2 cụm C_i và C_j , nó được định nghĩa như sau:

$$\text{link}(C_i, C_j) = \sum_{p \in C_i, p' \in C_j} \text{link}(p, p') \quad (5.21)$$

Một cách trực quan ta có thể thấy rằng nếu số lượng liên kết giữa 2 cụm là lớn thì chúng ta nên gộp chúng lại với nhau.

Thực nghiệm trên một số tập dữ liệu giao dịch giỏ hàng trong thực tế đã chứng minh giải thuật ROCK cho kết quả là các cụm có ý nghĩa hơn nhiều so với các giải thuật phân cụm truyền thống (dựa trên độ đo khoảng cách).

5.5. THUẬT TOÁN PHÂN CỤM DỰA TRÊN MẶT ĐỘ

Nhắc lại rằng các giải thuật phân cụm dựa trên độ đo khoảng cách hay độ tương tự chỉ phù hợp đối với các miền dữ liệu trong đó các cụm được phân bố theo hình cầu. Để xử lý trường hợp dữ liệu không phân bố theo hình cầu (mà có thể ở hình dạng bất kỳ),

lớp giải thuật phân cụm dựa trên mật độ đã được đề xuất. Một số giải thuật thuộc lớp giải thuật phân cụm dựa trên mật độ là DBSCAN, OPTICS và DENCLUE. Mục này sẽ trình bày giải thuật DBSCAN – một giải thuật đặc trưng thuộc lớp giải thuật phân cụm dựa trên mật độ.

Tên DBSCAN được viết tắt từ Density-Based Spatial Clustering of Application with Noise. Nó có thể phát hiện các cụm ở hình dạng bất kỳ và thậm chí cả trong trường hợp dữ liệu có chứa nhiều nhiễu. Giải thuật sẽ mở rộng các miền (cụm) nếu thấy mật độ của nó là cao. Nó định nghĩa một cụm là một tập các miền (phần tử dữ liệu) liên thông có mật độ cao nhất (density-connected). Có một số định nghĩa liên quan đến giải thuật này như sau.

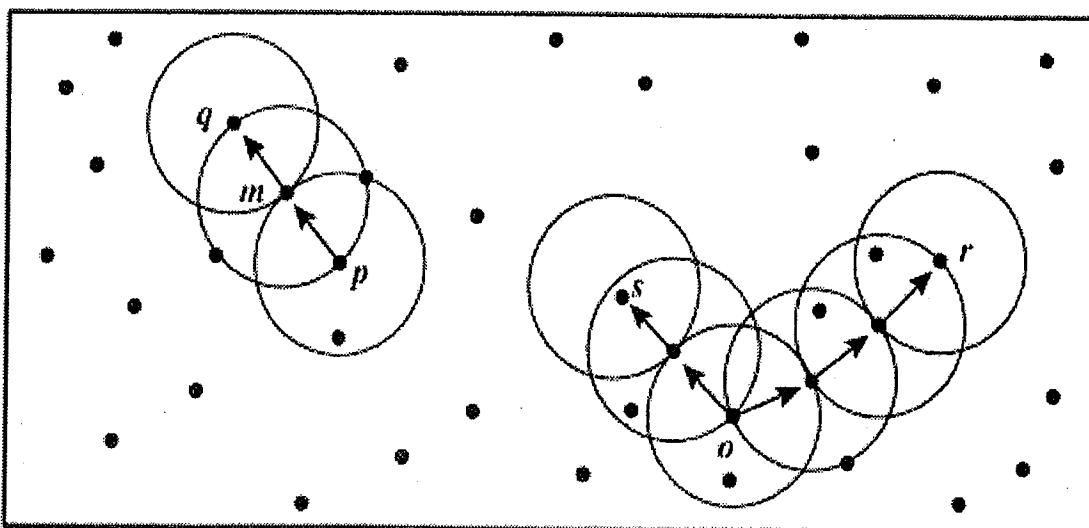
Cho tập các đối tượng D gồm n đối tượng (phần tử dữ liệu):

- Vùng lân cận trong vòng bán kính ϵ với tâm là đối tượng (phần tử dữ liệu) đang xét được ký hiệu là ϵ -neighborhood.
- Nếu ϵ -neighborhood của một đối tượng (phần tử dữ liệu) có số phần tử lớn hơn hoặc bằng một giá trị ngưỡng $MinPts$ thì nó được gọi là một *đối tượng lõi* (core object).
- Ta nói đối tượng p với *được trực tiếp theo mật độ* (directly density-reachable) từ đối tượng q nếu p ở trong miền ϵ -neighborhood của q và q là một đối tượng lõi.
- Đối tượng p được gọi là *với được theo mật độ* (density-reachable) từ đối tượng q (tương ứng với ϵ và $MinPts$) nếu tồn tại một dãy các đối tượng p_1, \dots, p_n trong đó $p_1 = q$, $p_n = p$ sao cho p_{i+1} là với được trực tiếp dựa trên mật độ từ p_i (với giá trị của i : $1 \leq i \leq n$).
- Đối tượng p được gọi là *liên thông mật độ* (density-connected) với đối tượng q (tương ứng với ϵ và $MinPts$) nếu tồn tại một đối tượng o sao cho cả p và q đều với được dựa trên mật độ từ o (tương ứng với ϵ và $MinPts$).
- Một *cụm dựa trên mật độ* (density-based cluster) là một tập hợp lớn nhất các đối tượng liên thông theo mật độ.

Chú ý là quan hệ *với được theo mật độ* là bất đối xứng (ngoại trừ các đối tượng lõi), tuy nhiên quan hệ *liên thông mật độ* lại là quan hệ đối xứng.

Minh họa cho các quan hệ được định nghĩa ở trên được thể hiện qua hình 5.7. Với giá trị ϵ được thể hiện là bán kính của các hình tròn, và $MinPts = 3$ thì:

- Các đối tượng có nhãn là m , o , p và r được gọi là các đối tượng lõi bởi vì chúng đều chứa ít nhất 3 đối tượng trong hình tròn bán kính ϵ của nó.
- Đối tượng q có thể với được trực tiếp theo mật độ từ m , m có thể với được trực tiếp theo mật độ từ p (và p có thể với được trực tiếp theo mật độ từ m , vì m và p đều là đối tượng lõi).
- Đối tượng q có thể với được (gián tiếp) theo mật độ từ p , vì tồn tại dây p, m, q thỏa mãn điều kiện với được theo mật độ. Tuy nhiên có thể dễ dàng nhận ra là p không với được theo mật độ từ q (tính bất đối xứng của quan hệ).
- o , r và s là liên thông mật độ vì tồn tại đối tượng o đã thỏa mãn điều kiện liên thông mật độ.



Hình 5.7. Minh họa các quan hệ trong DBSCAN

Giải thuật DBSCAN sẽ tìm các cụm bằng cách kiểm tra ϵ -neighborhood của từng đối tượng trong tập dữ liệu đầu vào D .

Nếu ε -neighborhood của đối tượng p chứa số đối tượng thỏa mãn ngưỡng $MinPts$ thì một cụm mới chứa đối tượng lõi p được tạo ra. DBSCAN tiếp tục mở rộng cụm bằng cách tìm các đối tượng với được trực tiếp theo mật độ từ các đối tượng lõi trong cụm. Quá trình mở rộng này có thể dẫn đến trường hợp ta ghép 2 hay nhiều cụm lại với nhau. Thuật toán dừng khi không có đối tượng nào được thêm vào các cụm.

Nếu có sử dụng cơ chế đánh chỉ mục thì độ phức tạp của DBSCAN là $O(n \log n)$, ngược lại thì độ phức tạp của nó là $O(n^2)$.

5.6. GIẢI THUẬT PHÂN CỤM DỰA TRÊN MÔ HÌNH

Phương pháp làm việc của các giải thuật thuộc lớp này là cố gắng làm tối ưu sự phù hợp giữa tập dữ liệu đầu vào với một mô hình toán học. Một số giải thuật điển hình thuộc lớp này là *cực đại kỳ vọng* (Expectation Maximization - EM), *phân cụm khái niệm* (Conceptual clustering) và phương pháp phân cụm dựa trên mô hình học máy mạng neural. Trong mục này chỉ xin trình bày giải thuật cực đại kỳ vọng.

Trong thực tế, mỗi cụm có thể được biểu diễn bằng một phân bố xác suất, nếu ta có k cụm thì sẽ có k phân bố xác suất được gọi là phân bố thành phần (component distribution), và toàn bộ tập dữ liệu sẽ là sự trộn hữu hạn (finite mixture) của các phân bố này (từ hữu hạn ở đây thể hiện số lượng các phân bố thành phần là hữu hạn). Do đó ta có thể phân cụm toàn bộ tập dữ liệu đầu vào bằng cách sử dụng mô hình mật độ trộn (mixture density model) của k phân bố xác suất, trong đó một phân bố biểu diễn một cụm. Như vậy, nhiệm vụ của giải thuật phân cụm là đi tìm (ước lượng) các tham số của các phân bố xác suất sao cho phù hợp với tập dữ liệu đầu vào nhất.

Cho một tập D gồm n phần tử dữ liệu $\{x_1, x_2, \dots, x_n\}$, và được chia thành k cụm. Để đơn giản, ta xét trường hợp mỗi phần tử dữ liệu được biểu diễn bằng 1 số thực. Gọi m_C , σ_C và $P(C)$ tương ứng là giá trị trung bình, độ lệch chuẩn và xác suất lấy mẫu của cụm C , các giá trị trên được tính như sau:

$$m_C = \frac{1}{|C|} \sum_{p \in C} p, \sigma_C = \sqrt{\frac{1}{|C|} \sum_{x \in C} (x - m_C)^2} \text{ và } P(C) = \frac{|C|}{|D|} \quad (5.22)$$

Khi đó bộ ba $\langle m_C, \sigma_C, P(C) \rangle$ được gọi là mô hình sinh của cụm C (theo phân bố chuẩn Gauss). Hình 5.8 minh họa trường hợp ta có 2 cụm, khi đó giá trị m_C và σ_C tương ứng sẽ là tâm và bán kính của đường tròn biểu diễn độ lệch chuẩn của cụm.

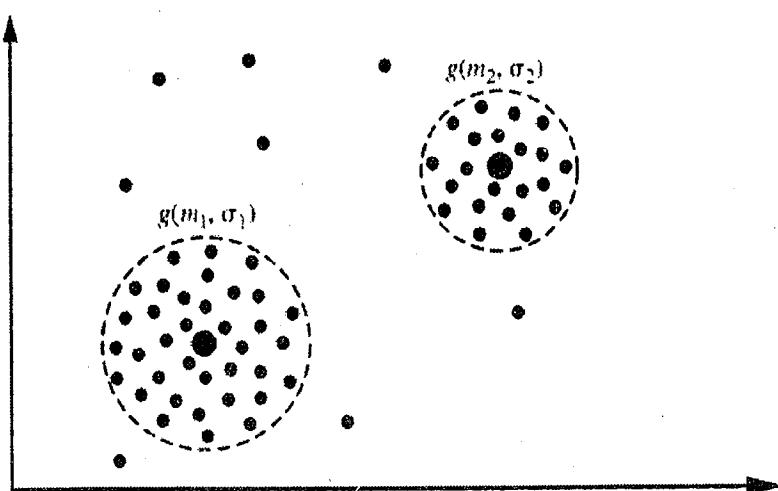
Giả sử chúng ta có tập dữ liệu được chia thành 2 cụm A và B . Cụm A gồm các phần tử dữ liệu $\{0, 0, 0, 0, 0, 0, 0.49, 0, 0, 0.387, 0.57\}$. Cụm B gồm các phần tử dữ liệu $\{0.961, 0.780, 0, 0.980, 0.135, 0.928, 0, 0.658, 0\}$. Khi đó bộ ba $\langle m_C, \sigma_C, P(C) \rangle$ được tính như sau:

$$\begin{aligned} m_A &= (0 + 0 + 0 + 0 + 0 + 0 + 0.49 + 0 + 0 + 0.387 + 0.57)/11 \\ &= 0.132 \end{aligned}$$

$$\sigma_A = 0.229 \text{ và } P(A) = 11/20 = 0.55$$

$$\begin{aligned} m_B &= (0.961 + 0.780 + 0 + 0.980 + 0.135 + 0.928 + 0 + 0.658 + 0)/9 \\ &= 0.494 \end{aligned}$$

$$\sigma_B = 0.449 \text{ và } P(B) = 9/20 = 0.45$$



Hình 5.8. Biểu diễn một cụm theo trọng tâm và độ lệch chuẩn

Sau khi đã có bộ ba $\langle m_C, \sigma_C, P(C) \rangle$ cho từng cụm, ta có thể xác định xác suất mà một phần tử dữ liệu thuộc vào cụm C là bao nhiêu. Trường hợp nếu phần tử dữ liệu được biểu diễn bằng các giá trị rời rạc (chỉ gồm các giá trị 0 và 1) thì xác suất của một phần tử dữ liệu x thuộc vào cụm C được tính bằng công thức Bayes:

$$P(C|x) = \frac{P(C)P(x|C)}{P(x)} \quad (5.23)$$

và $P(x|C)$ được tính bằng số lần xuất hiện của x trong cụm C chia cho tổng số phần tử dữ liệu trong cụm C . Trong trường hợp ta đang xét, dữ liệu được biểu diễn bằng số thực, khi đó xác suất $P(x|C)$ được tính bằng công thức

$$P(C|x) \approx \frac{f_c(x)P(C)}{P(x)} \quad (5.24)$$

$$\text{trong đó, } f_c(x) = \frac{1}{\sqrt{2\pi}\sigma_c} e^{-(x-m_c)/(2\sigma_c^2)} \quad (5.25)$$

Do $P(x)$ xuất hiện trong tất cả các công thức tính xác suất của x thuộc vào các cụm khác nhau nên ta có thể bỏ qua không cần tính. Nhưng khi đó các giá trị tính toán được $P(C|x)$ có thể không thỏa mãn điều kiện $\sum_c P(C|x) = 1$, do đó ta có thể cần phải chuẩn hóa lại.

Ví dụ trong trường hợp tập dữ liệu ở trên, nếu ta muốn xem xác suất của một phần tử dữ liệu có giá trị 0.78 thuộc vào từng cụm là bao nhiêu, ta có:

$$P(A|0.78) = f_A(0.78)P(A) = (0.032)(0.55) = 0.018$$

$$P(B|0.78) = f_B(0.78)P(B) = (0.725)(0.45) = 0.326$$

Thực hiện chuẩn hóa ta có:

$$P(A|0.78) = 0.018/(0.018 + 0.326) = 0.05;$$

$$P(B|0.78) = 0.326/(0.018 + 0.326) = 0.95$$

Khi sử dụng công thức trên thì ta cần phải chú ý là nếu có trường hợp có một thuộc tính nào đó có giá trị độ lệch chuẩn σ là 0 thì ta không thể tính được giá trị $f_c(x)$. Để xử lý trường hợp này ta có thể lấy một xác suất ngầm định nào đó, chẳng hạn là 0.05 để thay vào các xác suất $P(x|C)$.

Trong trường hợp tổng quát thì một phần tử dữ liệu có thể được biểu diễn bằng nhiều thuộc tính (nhiều chiều). Giả sử mỗi một phần tử dữ liệu x được biểu diễn bằng một vector d chiều

(x_1, x_2, \dots, x_d) , khi đó với giả thiết các thuộc tính là độc lập nhau thì ta có thể sử dụng công thức sau:

$$P(x | C) = P((x_1, x_2, \dots, x_d) | C) = \prod_{i=1}^d P(x_i | C) = \prod_{i=1}^d f_C^i(x_i) \quad (5.26)$$

Giải thuật cực đại kỳ vọng là một trong những giải thuật lặp để ước lượng các tham số cho mô hình. Nó cũng có thể coi là giải thuật mở rộng của k-means. Giải thuật k-means gán các phần tử dữ liệu vào các cụm có độ tương đồng với nó là lớn nhất, thì giải thuật cực đại kỳ vọng gán xác suất thuộc vào các cụm cho các phần tử dữ liệu. Nói một cách khác, giải thuật cực đại kỳ vọng thuộc loại thuật toán phân cụm xác suất. Với loại giải thuật phân cụm này thì không có ranh giới rõ ràng giữa các cụm. Cho một D chứa n phần tử dữ liệu $\{x_1, x_2, \dots, x_n\}$, tham số đầu vào k là số cụm cần tạo, thuật toán cực đại kỳ vọng hoạt động như sau:

Bước khởi tạo: Phân ngẫu nhiên các phần tử dữ liệu vào k cụm, mục đích của thao tác này là dùng để xây dựng bộ ba tham số $\langle m_C, \sigma_C, P(C) \rangle$ ban đầu cho k cụm. Các tham số này sẽ được làm mịn thông qua việc lặp 2 bước sau:

Bước kỳ vọng: Với từng phần tử x_i ($1 \leq i \leq n$) tính giá trị $w_i^C = P(C | x_i)$ là xác suất x_i thuộc vào cụm C . Chuẩn hóa giá trị w_i^C trên toàn bộ k cụm để đảm bảo $\sum_{C=1}^k w_i^C = 1$. Giá trị w_i^C thu được tại thời điểm này chính là giá trị kỳ vọng phần tử x_i thuộc vào cụm C .

Bước cực đại: Tính toán (ước lượng) lại giá trị của các tham số, cụ thể là giá trị trung bình m_C , độ lệch chuẩn σ_C và xác suất lấy mẫu $P(C)$ bằng công thức mới như sau (chứ không tính như công thức (5.22)):

$$m_C = \frac{\sum_{i=1}^n w_i^C x_i}{\sum_{i=1}^n w_i^C}, \quad \sigma_C^2 = \frac{\sum_{i=1}^n w_i^C (x_i - m_C)^2}{\sum_{i=1}^n w_i^C} \quad \text{và} \quad P(C) = \sum_{i=1}^n w_i^C \quad (5.27)$$

Lý do ta không thể tính được $P(C)$ dựa vào số phần tử thuộc vào lớp C (như công thức (5.8)) là vì giải thuật sẽ gán một xác suất phụ thuộc vào 1 cụm cho từng phần tử dữ liệu, nên không có ranh

giới rõ ràng giữa các cụm. Sau khi tính toán xong các xác suất lấy mẫu $P(C)$, ta cần chuẩn hóa lại để đảm bảo $\sum_C P(C) = 1$. Đây chính

là bước cực đại giá trị likelihood của phân bố xác suất trên tập dữ liệu đã cho. Giá trị likelihood L của phân bố xác suất được tính như sau:

$$L = \sum_{i=1}^n \log \sum_C P(x_i | C)P(C) \quad (5.28)$$

Chú ý là trong các công thức ở trên được thực hiện trên toàn bộ n phần tử trong tập dữ liệu đầu vào (chứ không phải là trên tập các phần tử thuộc vào cụm đang xem xét C).

Hai bước trên được lặp đi lặp lại cho đến khi giải thuật hội tụ hay nó đạt đến vị trí tối ưu toàn cục. Trong thực tế thì giải thuật hội tụ nhanh nhưng có thể không đạt đến vị trí tối ưu toàn cục. Giá trị likelihood L luôn tăng sau mỗi vòng lặp nên có một cách khác để dừng thuật toán là so sánh giá trị likelihood trong 2 vòng lặp gần nhau L_t và L_{t+1} , nếu $L_{t+1} - L_t < \epsilon$ (ϵ là một ngưỡng nào đó có giá trị rất nhỏ) thì ta có thể dừng thuật toán.

Cũng giống giải thuật k-means, một phần tử dữ liệu có thể gán đi gán lại vào các cụm khác nhau trong quá trình giải thuật hoạt động. Một đặc điểm nữa cũng giống giải thuật k-means là các cụm được tạo ngẫu nhiên giống với việc chọn ngẫu nhiên k phần tử làm trọng tâm của cụm (trong giải thuật k-means). Do vậy một trong những cách để tìm giá trị tối ưu toàn cục là chạy giải thuật cực đại kỳ vọng nhiều lần để tìm ra lần chạy có giá trị likelihood lớn nhất.

Nếu chúng ta để ý, thì trong giải thuật cực đại kỳ vọng ở trên được xây dựng trên cơ sở kết hợp với bộ phân lớp Naive Bayes (sẽ được trình bày chi tiết trong chương 6).

5.7. NHẬN XÉT SƠ BỘ CÁC THUẬT TOÁN PHÂN CỤM

Như đã được giới thiệu, thuật toán HAC thường chậm khi áp dụng cho các tập phân tử dữ liệu lớn. Các thuật toán khác theo

hướng này như *Single-link* và *Group-average* có thời gian thực hiện là $O(n^2)$, đồng thời thời gian kết nối hoàn toàn (*complete-link*) là $O(n^3)$ [Christopher08]. Các thuật toán theo hướng này là quá chậm so với yêu cầu của bài toán phân cụm Web. Một điểm đáng chú ý nữa đối với các thuật toán HAC là điều kiện dừng. Đã có rất nhiều đề xuất về điều kiện dừng được đưa ra nhưng chủ yếu là dựa trên việc điều kiện dừng đã được xác định trước (chẳng hạn, dừng khi chỉ còn 5 cụm). Điều kiện dừng đối với các thuật toán này (HAC) là cực kỳ quan trọng. Nếu như thuật toán ghép các cụm “tốt” với nhau có thể tạo ra kết quả không theo mong muốn của người dùng. Trên Web, với kết quả trả về theo truy vấn là vô cùng đa dạng (về số lượng, độ lớn, kiểu và sự phù hợp của các phần tử dữ liệu) thì điều kiện dừng không tốt sẽ làm cho kết quả trở nên nghèo nàn.

Thuật toán k-means thuộc vào lớp các thuật toán phân cụm thời gian tuyến tính và là những lựa chọn tốt nhất để đáp ứng yêu cầu về tốc độ của bài toán phân cụm on-line. Thời gian thực hiện của các thuật toán này là $O(nk)$ trong đó k là số các cụm mong muốn.Thêm một ưu điểm của thuật toán k-means so với HAC là việc đáp ứng các yêu cầu của bài toán phân cụm Web là nó có thể tạo ra các cụm có sự giao thoa. Điểm yếu chính của thuật toán này là nó chạy hiệu quả nhất chỉ khi các cụm mong muốn là các miền hình cầu đối với độ đo tương tự được dùng. Không có lý do gì để tin rằng các phần tử dữ liệu sẽ thuộc vào các miền cầu. Vì vậy, thuật toán có thể làm mất đi các thông tin có giá trị.

Buckshot là thuật toán kết hợp giữa HAC và k-means trong đó việc khởi tạo các trọng tâm cụm cho k-means được thực hiện bởi thuật toán HAC trên một mẫu của tập phần tử dữ liệu [Cutting93].

Các thuật toán như HAC, k-means hay Buckshot đều không phải là các thuật toán có tính gia tăng. Một số thuật toán gia tăng đã được phát triển như thuật toán phân cụm cây hậu tố (Suffix Tree Clustering - STC) [Branson02], với thời gian thực hiện $O(n)$ trong đó n là kích thước của tập phần tử dữ liệu.

5.8. ĐÁNH GIÁ CÁC GIẢI THUẬT PHÂN CỤM

5.8.1 Đánh giá dựa trên độ tương tự

Nhiệm vụ của các giải thuật phân cụm là nhóm các phần tử dữ liệu tương tự nhau thành một cụm, do đó chất lượng của giải thuật phân cụm sẽ được đánh giá mức độ giống nhau giữa các phần tử trong cùng một cụm. Một giải thuật phân cụm tốt sẽ cho kết quả là độ tương tự nội tại trong một cụm là cao và độ tương tự giữa các cụm là thấp. Vậy ta có thể dùng bất kỳ hàm đo độ tương tự của các phần tử dữ liệu trong cùng một cụm để đánh giá chất lượng của giải thuật phân cụm. Ta có thể sử dụng hàm J (trong công thức (5.9)) hàm đã được tích hợp vào trong một số giải thuật phân cụm để đánh giá chất lượng kết quả phân cụm của giải thuật. Một công thức khác ta có thể dùng để tính độ tương tự nội tại của một cụm là dựa vào độ tương tự của từng cặp dữ liệu trong cụm:

$$J = \frac{1}{2} \sum_C \frac{1}{|C|} \sum_{p_i, p_j \in C} sim(p_i, p_j) \quad (5.29)$$

Biến đổi tương đương công thức trên có thể được viết lại thành:

$$J = \frac{1}{2} \sum_C \frac{1}{|C|} \sum_{p_i, p_j \in C} sim(p_i, p_j) = \frac{1}{2} \sum_C |C| sim(C) \quad (5.30)$$

Trong đó $sim(C)$ là độ tương tự trung bình giữa các cặp phần tử dữ liệu trong cụm C . Với công thức này, giá trị của J càng lớn thì càng chứng tỏ giải thuật phân cụm cho chất lượng càng tốt.

Một công thức khác có thể dùng để đánh giá chất lượng phân cụm là hàm *tổng bình phương lỗi*: Ý tưởng của hàm đánh giá này là dựa trên quan điểm trọng tâm của mỗi cụm sẽ biểu diễn tốt nhất cụm đó, với mỗi phần tử dữ liệu p trong cụm đó càng cách xa trọng tâm của cụm thì “lỗi” của phần tử dữ liệu đó càng cao. Giá trị lỗi của phần tử dữ liệu p trong cụm được đo bằng chiều dài của vector $p - m_C$. Với mỗi cụm C , chúng ta xác định trọng tâm m_C của cụm đó. Hàm đánh giá chất lượng phân cụm này được tính bằng:

$$E = \sum_C \sum_{p \in C} |p - m_C|^2 \quad (5.31)$$

trong đó, m_C là trọng tâm của cụm được tính theo công thức (5.10). Với độ đo đánh giá này thì giá trị E của một giải thuật nào đó càng nhỏ thì chất lượng phân cụm của nó càng tốt. Bằng cách biến đổi số học, công thức (5.31) trên có thể được viết lại thành tổng khoảng cách từng cặp phần tử dữ liệu trong cụm:

$$E = \frac{1}{2} \sum_C \frac{1}{|C|} \sum_{p_i, p_j \in C} \|p_i - p_j\|^2 \quad (5.32)$$

5.8.2. Đánh giá dựa trên dữ liệu gán nhãn

Phương pháp đánh giá dựa vào độ chính xác và tỉ lệ lỗi

Phương pháp đánh giá ở mục 5.8.1 hoàn toàn dựa vào độ tương tự của các phần tử dữ liệu trong cùng một cụm. Tuy nhiên khi chúng ta phân thủ công các phần tử dữ liệu vào các cụm, chúng ta cần thêm một số tri thức khác nữa mà thông thường các tri thức này không có sẵn hay hiển thị rõ ràng trong nội dung của các phần tử dữ liệu. Khi ta đã biết trước nhãn của các phần tử dữ liệu thuộc vào các cụm thì việc đánh giá thuật toán phân cụm chỉ dựa vào hàm điều kiện J_s như trên là không chính xác. Phần này chúng ta sẽ tìm hiểu thêm một số phương pháp đánh giá các giải thuật phân cụm một cách chính xác hơn. Thông thường dữ liệu gán nhãn thường được dùng để áp dụng cho các giải thuật học có giám sát, tuy nhiên ngay cả giải thuật học không giám sát như các giải thuật phân cụm thì dữ liệu gán nhãn cũng hữu ích, cụ thể ta có thể dùng để đánh giá chất lượng của giải thuật phân cụm bằng cách so sánh dữ liệu gán nhãn (dữ liệu phân cụm bằng tay) với kết quả của giải thuật phân cụm. Chú ý rằng trong trường hợp này tuy rằng chúng ta đã có dữ liệu đã được gán nhãn (lớp/cụm) nhưng các nhãn của các phần tử dữ liệu không được dùng trong quá trình phân cụm mà chỉ dùng để đánh giá chất lượng của giải thuật phân lớp. Có một số độ đo đánh giá được dùng trong phương pháp này: *độ chính xác* (precision), *tỉ lệ lỗi* (error), *độ hồi tưởng* (recall) và

F-measure. Giả sử dữ liệu phân lớp bằng tay gồm có 2 lớp (để phân biệt với cụm) A và B , và giải thuật phân cụm cũng phân thành 2 cụm. Đối với mỗi lớp ví dụ lớp A , những phần tử dữ liệu thuộc vào lớp A được gọi là các *ví dụ dương* (positive), những phần tử dữ liệu không thuộc vào lớp A được gọi là các *ví dụ âm* (negative). Kết quả phân cụm của một giải thuật sẽ có một số khả năng sau:

- Đúng dương (true positive): phần tử dữ liệu là ví dụ dương và được giải thuật phân cụm dự đoán là ví dụ dương (phân cụm đúng), ký hiệu là TP.
- Sai dương (false positive): phần tử dữ liệu là ví dụ dương nhưng giải thuật phân cụm lại đoán là ví dụ âm (phân cụm sai), ký hiệu là FP.
- Đúng âm (true negative): phần tử dữ liệu là ví dụ âm và được giải thuật phân cụm đoán là ví dụ âm (phân cụm đúng), ký hiệu là TN.
- Sai âm (false negative): phần tử dữ liệu là ví dụ âm và được giải thuật phân cụm đoán là ví dụ dương (phân cụm sai), ký hiệu là FN.

Để tính toán ra được các độ đo ở trên ta dựa vào các khả năng liệt kê ở trên. Để dễ tính toán ta có thể lập ma trận biểu diễn các trường hợp trên, ma trận này được gọi là ma trận lẩn lộn (confusion matrix) như bảng 5.3:

Bảng 5.3. Ma trận lẩn lộn

		Lớp được dự đoán bởi giải thuật phân cụm	
		Dương	Âm
Lớp thực tế	Dương	TP	FN
	Âm	FP	TN

Với trường hợp chỉ có 2 lớp như trên, từ ma trận lẩn lộn này các công thức độ đo sẽ được tính toán cụ thể như sau:

- Tỉ lệ lỗi tổng thể:

$$\text{Error} = \frac{FP + FN}{TP + FP + TN + FN} \times 100\% \quad (5.33)$$

- Độ chính xác tổng thể:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (5.34)$$

Đối với từng lớp thì ta có thể sử dụng thêm 2 độ đo đánh giá sau:

- Độ chính xác:

$$P = \frac{TP}{TP + FP} \times 100\% \quad (5.35)$$

- Độ hồi tưởng:

$$R = \frac{TP}{TP + FN} \times 100\% \quad (5.36)$$

Ví dụ bảng 5.2 đưa ra kết quả phân cụm với thuật toán k-means với k là 2, so sánh với tập dữ liệu đã được gán nhãn. Với kết quả phân cụm với thuộc tính A_3 , ta có các giá trị của các độ đo như sau:

$$\text{Error} = \frac{FP + FN}{TP + FP + TN + FN} \times 100\% = \frac{0 + 3}{8 + 0 + 9 + 3} \times 100\% = 15\%$$

$$\begin{aligned} \text{Accuracy} &= \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \\ &= \frac{8 + 9}{8 + 0 + 9 + 3} \times 100\% = 75\% \end{aligned}$$

Với chỉ riêng lớp A ta có các giá trị của độ chính xác và độ hồi tưởng như sau:

$$P = \frac{TP}{TP + FP} \times 100\% = \frac{8}{8 + 0} \times 100\% = 100\%$$

$$R = \frac{TP}{TP + FN} \times 100\% = \frac{8}{8 + 3} \times 100\% = 73\%$$

Tương tự ta cũng có thể tính toán được độ chính xác (precision) của phân cụm với thuộc tính A_6 cho lớp A là 60% và độ hồi tưởng là 82%.

So sánh kết quả độ chính xác và độ hồi tưởng của phân cụm với 2 thuộc tính khác nhau A_3 và A_6 như trên rất khó để có thể kết

luận là kết quả nào tốt hơn vì cái có độ chính xác cao hơn thì lại có độ hồi tưởng thấp hơn và ngược lại. Do vậy một độ đo khác được đề xuất là F-measure (hay còn có tên khác là F-score) đã kết hợp 2 loại độ đo này lại để giúp đánh giá chính xác được kết quả nào tốt hơn. Công thức của độ đo này là:

Bảng 5.4. Kết quả phân cụm với k-means chỉ sử dụng 1 thuộc tính

	Thuộc tính A_3		Thuộc tính A_6	
	Lớp được dự đoán bởi giải thuật phân cụm		Lớp được dự đoán bởi giải thuật phân cụm	
	Lớp thực tế	A	B	A
A	8	3	9	2
B	0	9	6	3

$$F\text{- measure} = \frac{2 \times P \times R}{P + R} \quad (5.37)$$

Như vậy kết quả của giải thuật phân cụm với thuộc tính A_3 có F-measure = 86%, và phân cụm với thuộc tính A_6 có F-measure = 69%. Như vậy có thể kết luận là kết quả của phân cụm với thuộc tính A_3 tốt hơn phân cụm với thuộc tính A_6 .

Ta cũng có thể mở rộng trường hợp có 2 lớp sang trường hợp có nhiều hơn 2 lớp/cụm. Gọi số lớp là m , số cụm là k , chú ý là m có thể khác k . Ma trận lẫn lộn tổng quát (cho m lớp) sẽ có dạng như bảng 5.5. Và công thức dùng để tính toán các độ đo cho các ô (i, j) là:

$$\text{Độ chính xác } P(i, j) = \frac{n_{ij}}{\sum_{i=1}^m n_{ij}} \times 100\% \quad (5.38)$$

$$\text{Độ hồi tưởng } P(i, j) = \frac{n_{ij}}{\sum_{j=1}^k n_{ij}} \times 100\% \quad (5.39)$$

$$\text{Độ đo F-measure } F(i, j) = \frac{2 \times P(i, j) \times R(i, j)}{P(i, j) + R(i, j)} \quad (5.40)$$

Để thu được kết quả F-measure trên toàn bộ các cụm ta có thể dùng công thức:

$$F = \sum_{i=1}^m \frac{n_i}{n} \max_{j=1,\dots,k} F(i, j) \quad (5.41)$$

trong đó, n_i là tổng số phần tử dữ liệu thuộc vào lớp i (hay tổng số hàng thứ i trong ma trận lẩn lộn) $n_i = \sum_{j=1}^k n_{ij}$, và n là tổng số phần tử dữ liệu có trong tập dữ liệu $n = \sum_{i=1}^m \sum_{j=1}^k n_{ij}$. Tỉ lệ $\frac{n_i}{n}$ trong công thức trên cho biết được độ “quan trọng” của lớp thứ i trong toàn bộ tập dữ liệu.

Giả sử với kết quả phân cụm với thuộc tính A_6 ở bảng 5.4 ta có thể tính toán các độ đo đánh giá như sau:

$$P(1,1) = 9*100%/(9+6) = 60%; R(1,1) = 9*100%/(9+2) = 82%;$$

$$F(1,1) = 2*0.6*0.82/(0.6+0.82) = 69%;$$

$$P(1,2) = 2*100%/(2+3) = 40%; R(1,2) = 2*100%/(9+2) = 18%;$$

$$F(1,2) = 2*0.4*0.18/(0.4+0.18) = 25%;$$

$$P(2,1) = 6*100%/(6+3) = 67%; R(2,1) = 6*100%/(6+9) = 40%;$$

$$F(2,1) = 2*0.67*0.4/(0.67+0.4) = 50%;$$

$$P(2,2) = 3*100%/(3+2) = 60%; R(2,2) = 3*100%/(3+6) = 33%;$$

$$F(2,2) = 2*0.60*0.33/(0.60+0.33) = 43%;$$

Và giá trị F-measure toàn cục $F = \frac{15}{20} \times 0.69 + \frac{5}{20} \times 0.50 = 64\%$

Bảng 5.5. Ma trận lẩn lộn để đánh giá thuật toán phân cụm bằng dữ liệu gán nhãn trong trường hợp tổng quát

Lớp	Cụm					
	1	...	j	...	k	
1	n_{11}	...	n_{1j}	...	n_{1k}	
...
i	n_{i1}	...	n_{ij}	...	n_{ik}	
...
m	n_{m1}	...	n_{mj}	...	n_{mk}	

Phương pháp đánh giá dựa vào entropy

Một phương pháp đánh giá dựa vào lý thuyết xác suất bằng cách giả thiết nhãm lỏp của các phần tử dữ liệu trong tập dữ liệu là các sự kiện ngẫu nhiên. Giả thiết này cho phép chúng ta có thể đánh giá được phân bố xác suất trong mỗi cụm. Xác suất p_{ij} của lớp i ở trong cụm j có thể được ước lượng bằng tỉ lệ xuất hiện của các phần tử dữ liệu có nhãn i ở trong cụm j . Sử dụng ma trận lẩn lộn ta có thể tính được xác suất này là:

$$p_{ij} = \frac{n_{ij}}{\sum_{i=1}^m n_{ij}} \quad (5.42)$$

Nếu ta chú ý thì đây chính là độ chính xác $P(i,j)$ theo cách tính ở trên. Entropy là độ đo sự hỗn độn của thông tin, và entropy của cụm j được định nghĩa bằng:

$$H_j = -\sum_{i=1}^m p_{ij} \log p_{ij} \quad (5.43)$$

Và entropy của toàn bộ các cụm là:

$$H = \sum_{j=1}^k \frac{n_j}{n} H_j \quad (5.44)$$

trong đó, n_j là số lượng các phần tử dữ liệu nằm trong cụm j và n là tổng số các phần tử dữ liệu trong tập dữ liệu. Giải thuật phân cụm càng tốt thì entropy của nó có kết quả càng nhỏ. Ví dụ với kết quả phân cụm ở bảng 5.4 sử dụng thuộc tính A_6 , ta có thể tính giá trị entropy như sau:

$$\begin{aligned} H &= \frac{15}{20} \left(-\frac{9}{15} \log \frac{9}{15} - \frac{6}{15} \log \frac{6}{15} \right) + \frac{5}{20} \left(-\frac{2}{5} \log \frac{2}{5} - \frac{3}{5} \log \frac{3}{5} \right) \\ &= 0.292285253 \end{aligned}$$

5.9. MỘT SỐ ỨNG DỤNG CỦA PHÂN CỤM

Tuy bản chất của các giải thuật phân cụm chỉ là nhóm các phần tử dữ liệu lại với nhau thành cụm có các đặc điểm nào đó tương tự nhau, nhưng các ứng dụng của phân cụm lại rất đa dạng. Cho một tập dữ liệu gồm n phần tử, giải thuật phân cụm có thể

giúp ta hiểu cấu trúc phân bố tự nhiên của dữ liệu, hoặc đơn thuần giải thuật phân cụm có thể hiển thị cho ta thấy dữ liệu được phân bố như thế nào. Điểm mấu chốt của việc ứng dụng các giải thuật phân cụm là phụ thuộc vào tính sáng tạo của các nhà nghiên cứu. Một số ứng dụng của giải thuật phân cụm đã được đề xuất có thể liệt kê như sau:

- Trong sinh học: Phân cụm có thể giúp chúng ta tìm ra được các loại gen nào có các mẫu quan hệ với nhau.
- Trong kinh doanh: Phân cụm có thể giúp doanh nghiệp phân loại được khách hàng với các nhu cầu riêng, từ đó có các hướng tiếp thị khác nhau cho từng nhóm khách hàng.
- Trong khai phá dữ liệu văn bản, web, phân cụm có thể giúp chúng ta phân văn bản thành các nhóm thuộc các thể loại khác nhau. Một ví dụ khác: ta có thể phân cụm dữ liệu trả về từ một máy tìm kiếm (chẳng hạn như Google) để giúp người dùng có thể tìm tại liệu một cách nhanh chóng bằng cách chỉ cần tìm các tài liệu nằm trong cụm mà mình quan tâm.
- Trong xử lý ảnh: Phân cụm có thể giúp chúng ta phân loại được các đối tượng khác nhau trong một ảnh đầu vào. Hay ta có thể khoanh vùng được những nơi có cách thức sử dụng đất giống nhau dựa vào ảnh vệ tinh. Hay ta có thể phân loại các ảnh thành các thể loại giống nhau phục vụ cho quá trình tìm kiếm. Ví dụ nếu ta phân cụm được các ảnh về con hổ thì cụm ảnh này sẽ được dùng để làm kết quả cho câu truy vấn ảnh về hổ.
- Trong chứng khoán: Phân cụm có thể giúp ta phân loại được các mã chứng khoán tiềm năng hay ít tiềm năng.
- Trong bài toán lọc cộng tác, ta có thể phân cụm người dùng có thói quen mua hàng giống nhau, khi có một người dùng mới, ta sẽ tìm cụm tương ứng với người dùng này, từ đó có thể tư vấn các mặt hàng mà người dùng mới này có thể muốn mua. Đây là một phương pháp xử lý bài toán tư vấn (recommender system).

- Ngoài khả năng ứng dụng trực tiếp, các thuật toán phân cụm còn được sử dụng như bước tiền xử lý trong một số bài toán khai phá dữ liệu khác. Chẳng hạn trong bài toán tìm ảnh đại diện (thumbnail) cho một clip nào đó, ta phân các frame của clip đó thành các cụm tương ứng với một cảnh scene, sau đó giải thuật tiếp theo sẽ lựa chọn ảnh “tốt nhất” trong mỗi cụm làm ảnh đại diện.

CÂU HỎI VÀ BÀI TẬP

5.1. Mô tả phương pháp tính độ tương tự cũng như độ khác biệt của 2 phần tử dữ liệu có các kiểu dữ liệu biểu diễn các thuộc tính

- Giá trị rời rạc nhị phân
- Giá trị rời rạc tổng quát
- Giá trị liên tục

5.2. Cho 2 phần tử dữ liệu trong không gian 4 chiều được biểu diễn bằng các vector tương ứng là $(22, 1, 42, 10)$ và $(20, 0, 36, 8)$.

- Tính khoảng cách Manhattan giữa 2 phần tử trên
- Tính khoảng cách Euclidean giữa 2 phần tử trên
- Tính khoảng cách Minkowski giữa 2 phần tử trên với $q = 3$

5.3. Giả sử ta có tập dữ liệu $A1(2, 10), A2(2, 5), A3(8, 4), B1(5, 8), B2(7, 5), B3(6, 4), C1(1, 2), C2(4, 9)$. Sử dụng thuật toán k-means với độ đo khoảng cách Euclidean và 3 trọng tâm ban đầu là $A1, B1$ và $C1$ để phân dữ liệu ra làm 3 cụm. Tìm:

- Trọng tâm của 3 cụm sau vòng lặp đầu tiên (của thuật toán k-means)
- Các cụm kết quả của thuật toán

5.4. Lấy bộ công cụ xử lý khai phá dữ liệu weka tại địa chỉ <http://www.cs.waikato.ac.nz/ml/weka/>, tìm cách sử dụng thuật toán k-means được cài đặt sẵn để phân cụm một tập dữ liệu đi kèm theo gói phần mềm này, các file dữ liệu được đặt trong thư mục./data tại thư mục cài đặt của weka. Chú ý là trong bộ phần mềm này giải thuật k-means có tên là SimpleKMeans.

- 5.5. Thực nghiệm phân cụm với giải thuật EM trong weka với một tập dữ liệu đi kèm với phần mềm weka.
- 5.6. Thực nghiệm phân cụm với giải thuật HAC có tên là FarthestFirst trong weka với một tập dữ liệu đi kèm với phần mềm weka.
- 5.7. Thực nghiệm phân cụm với giải thuật HierarchicalCluster trong weka với một tập dữ liệu đi kèm với phần mềm weka.
- 5.8. Thực nghiệm phân cụm với giải thuật DBScan trong weka với một tập dữ liệu đi kèm với phần mềm weka.
- 5.9. Thực nghiệm phân cụm với giải thuật MakeDensityBasedClusterer trong weka với một tập dữ liệu đi kèm với phần mềm weka.
- 5.10. Cài đặt thuật toán phân cụm k-means với gán cứng, sau đó áp dụng trên một tập dữ liệu đi kèm với phần mềm weka. Chú ý cần phải tìm hiểu định dạng file arff của weka để biết cách xử lý dữ liệu.
- 5.11. Cài đặt thuật toán phân cụm phân cấp gộp từ dưới lên với độ đo *người láng giềng gần nhất*, sau đó áp dụng trên một tập dữ liệu đi kèm với phần mềm weka.
- 5.12. Cài đặt thuật toán phân cụm phân cấp gộp từ dưới lên HAC với độ đo *người láng giềng xa nhất*, sau đó áp dụng trên một tập dữ liệu đi kèm với phần mềm weka.
- 5.13. Cài đặt thuật toán phân cụm phân cấp gộp từ dưới lên HAC với độ đo *tương tự trung bình*, sau đó áp dụng trên một tập dữ liệu đi kèm với phần mềm weka.
- 5.14. Dùng phương pháp đánh giá dựa vào độ tương tự để đánh giá các thuật toán phân cụm từ bài 9 đến bài 12.
- 5.15. Dùng độ đo F-score để đánh giá các thuật toán phân cụm từ bài 10 đến bài 12.
- 5.16. Dùng độ đo Entropy để đánh giá các thuật toán phân cụm từ bài 10 đến bài 12.

Chương 6.

PHÂN LỚP DỮ LIỆU

6.1. GIỚI THIỆU

Phân lớp là một trong những mối quan tâm nhiều nhất của con người trong quá trình làm việc với một tập hợp đối tượng. Điều này giúp con người có thể tiến hành việc sắp xếp, tìm kiếm các đối tượng một cách thuận lợi. Khi biểu diễn đối tượng vào các cơ sở dữ liệu, tính chất lớp vốn có của đối tượng trong thực tế thường được biểu diễn tương ứng bằng một thuộc tính "lớp" riêng biệt. Chẳng hạn, trong hệ thống thông tin quản lý tư liệu của thư viện, thuộc tính về loại tư liệu có miền giá trị là tập tên chuyên ngành của tư liệu, gồm các giá trị như "Tin học", "Vật lý", ... Trước đây các công việc gán các giá trị của thuộc tính lớp thường được làm một cách thủ công. Nhưng hiện nay, với sự bùng nổ của thông tin và các loại dữ liệu, việc đánh giá, thuộc tính lớp một cách thủ công là rất khó khăn, có thể nói là không thể. Do vậy các phương pháp phân lớp tự động là rất cần thiết và là một trong những chủ đề chính trong khai phá dữ liệu.

Các cơ sở dữ liệu thường chứa rất nhiều các thông tin ẩn – các thông tin có thể sử dụng phục vụ quá trình phân lớp. Các giải thuật phân lớp thường phân tích dữ liệu nhằm tìm ra các mô hình mô tả các lớp dữ liệu, từ đó có thể quyết định được một phần tử dữ liệu mới là thuộc vào lớp nào.

Việc tìm ra lớp của một phần tử dữ liệu mới trong nhiều trường hợp có ý nghĩa rất quan trọng, nó hỗ trợ quá trình ra quyết định thông minh thậm chí là những quyết định mang tính sống

còn. Ví dụ, trong ngân hàng, một nhân viên cho vay vốn rất muốn có một hệ thống có khả năng tự học từ các dữ liệu lịch sử để có thể quyết định được một đơn vay vốn mới của khách hàng thuộc lớp “an toàn” hay “mạo hiểm”, trên cơ sở đó sẽ có các quyết định phù hợp. Một nhân viên tiếp thị trong một công ty buôn bán hàng điện tử thì rất muốn biết một khách hàng có khả năng mua máy tính hay không. Hay một bác sĩ sẽ rất muốn có một hệ thống phân tích dữ liệu điều trị lịch sử để dự đoán xem một bệnh nhân mới với những triệu chứng thu được sẽ thuộc bệnh nào, trên cơ sở đó sẽ có các phác đồ điều trị tương ứng.

Bản chất của bài toán phân lớp là dự đoán các nhãn (hay lớp) của các phần tử dữ liệu đầu vào và các nhãn (hay lớp) này là các giá trị rời rạc. Thông thường, các giải thuật phân lớp thường hoạt động thông qua 2 bước. Bước đầu tiên nó sẽ phân tích tập dữ liệu đã gán nhãn để tìm ra mô hình phù hợp mô tả tập dữ liệu đó. Bước này được gọi là *bước học* (learning step) hay *pha học* (learning phase) và tập dữ liệu gán nhãn phục vụ quá trình học này được gọi là *dữ liệu huấn luyện* (training data). Dữ liệu huấn luyện là một tập các *phần tử dữ liệu* (data point) có gán nhãn, hay còn được gọi là *bản ghi* (tuple) mô tả dữ liệu và nhãn (hay lớp) tương ứng của bản ghi đó. Trong cuốn giáo trình này khái niệm bản ghi và phần tử dữ liệu có cùng ý nghĩa với nhau, tương tự khái niệm nhãn và lớp cũng có cùng ý nghĩa. Ngoài ra còn có rất nhiều thuật ngữ khác cũng được sử dụng rộng rãi có cùng ý nghĩa với khái niệm phần tử dữ liệu như: mẫu (sample), ví dụ (example), thể hiện (instance) hay đối tượng (object). Một phần tử dữ liệu X thường được biểu diễn bằng một vector n chiều $X = (x_1, x_2, \dots, x_n)$, trong đó mỗi phần tử trong vector x_i chứa một giá trị biểu diễn thuộc tính (attribute) A_i của phần tử dữ liệu đó. Một thuật ngữ khác cùng ý nghĩa với khái niệm thuộc tính là khái niệm *đặc trưng* (feature). Vì nhãn của các phần tử dữ liệu được đi kèm với dữ liệu trong tập dữ liệu huấn luyện nên bước này còn được gọi là *học có giám sát* (supervised learning). Hay nói một cách khác, các giải thuật phân

lớp là thuộc lớp giải thuật học có giám sát. Về bản chất trong bước 1 này, các giải thuật phân lớp học ra hàm $y=f(X)$ để từ đó khi có một phần tử X mới nó sẽ dự đoán ra nhãn y tương ứng với nó. Theo khía cạnh này thì ta có thể thấy bước 1 là quá trình học ra một hàm hay một ánh xạ (mapping) nó có khả năng phân loại được các lớp dữ liệu. Tùy vào các giải thuật khác nhau mà hàm $f(X)$ này có thể có các dạng khác nhau như ở dạng luật (rule), cây quyết định (decision tree) hay các công thức toán học, ...

Sau khi học được hàm phân lớp, các giải thuật có thể dùng để dự đoán các dữ liệu mới. Tuy nhiên trước khi đem giải thuật vào ứng dụng trong thực tế, các giải thuật phải trải qua bước thứ 2 là bước kiểm tra hiệu năng của chúng. Để tránh hiện tượng quá phù hợp (overfit), một tập dữ liệu khác gọi là *tập dữ liệu kiểm thử* (testing set) sẽ được sử dụng để đo độ chính xác của giải thuật. Thông thường tập dữ liệu kiểm thử sẽ không chứa bất kỳ phần tử dữ liệu nào nằm trong tập dữ liệu huấn luyện. Cũng giống tập dữ liệu huấn luyện, trong tập dữ liệu kiểm thử, từng phần tử dữ liệu cũng có nhãn đi kèm. Các nhãn này được dùng để so sánh với nhãn được các giải thuật phân lớp dự đoán. Tỷ lệ đoán đúng nhãn của các giải thuật phân lớp được gọi là *độ chính xác* (accuracy) của giải thuật. Khi chất lượng phân lớp của các giải thuật là chấp nhận được trong một miền dữ liệu cụ thể nào đó, ta có thể dùng chúng để dự đoán lớp của các phần tử dữ liệu mới hoàn toàn chưa biết trước (thuật ngữ tiếng Anh là “unkown data” hay “previously unseen data”).

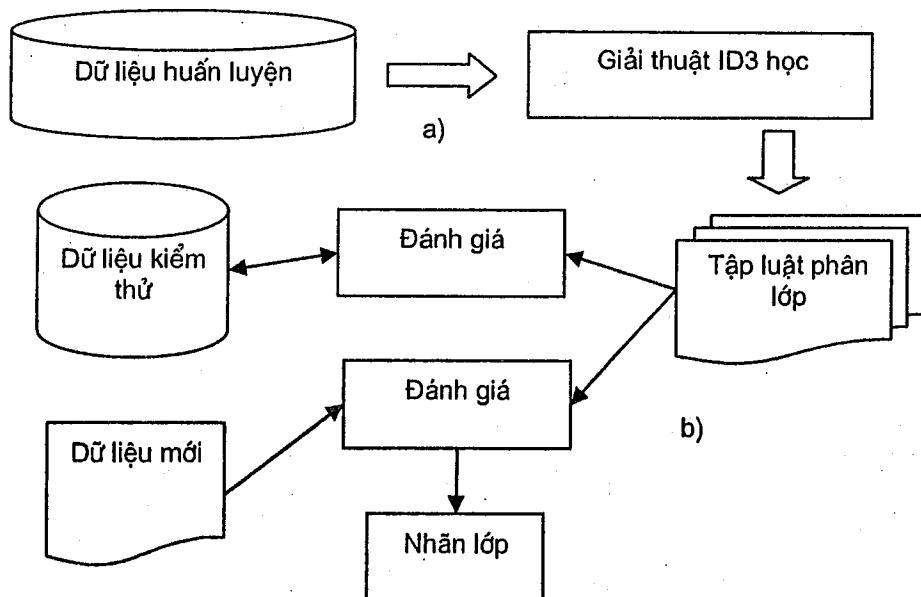
Minh họa của quá trình phân lớp được thể hiện trên hình 6.1 mô tả quá trình phân lớp của bài toán cho vay vốn trong ngân hàng. Trong đó hình 6.1 a) mô tả quá trình học của giải thuật. Kết quả của quá trình học là hàm phân lớp được thể hiện dưới dạng các luật. Hình 6.1 b) mô tả quá trình kiểm thử cũng như quá trình dự đoán dữ liệu mới. Hình 6.9 cũng minh họa mô hình chung của các giải thuật phân lớp: từ tập dữ liệu huấn luyện, các giải thuật

sẽ học và tìm ra mô hình mô tả dữ liệu đầu vào, kết quả của việc học là một mô hình. Mô hình này có thể đem ra dự đoán các phần tử dữ liệu mới. Tuy nhiên ta cũng sẽ cần bước thêm bước kiểm thử trong quá trình xây dựng một bộ phân lớp để đảm bảo chất lượng của nó phù hợp với miền ứng dụng.

Vì số lượng các giải thuật phân lớp là rất lớn, nên phần tiếp theo của chương này chúng ta sẽ chỉ tìm hiểu một số thuật toán phân lớp thông dụng.

6.2. PHÂN LỚP BẰNG CÂY QUYẾT ĐỊNH

J. Ross Quinlan là người phát triển giải thuật cây quyết định có tên là ID3 (viết tắt từ cụm từ “Iterative Dichotomiser”), sau đó cũng chính tác giả này đề xuất giải thuật phân lớp C4.5 (một hậu duệ của thuật toán ID3). Giải thuật C4.5 này đã được dùng làm chuẩn (benchmark) để các thuật toán mới so sánh. Cũng trong khoảng thời gian này thì một nhóm các nhà thống kê gồm L. Breiman, J. Friedman, R. Olshen và C. Stone đã xuất bản cuốn sách “Classification and Regression Trees (CART)” mô tả phương pháp tạo cây quyết định nhị phân. Giải thuật ID3 và CART đã trở thành các hòn đá tảng và nó mở đầu cho hàng loạt các giải thuật dựa trên *học quy nạp cây quyết định* (decision tree induction). Giải thuật học dựa trên cây quyết định hoạt động trên tập dữ liệu được biểu diễn bằng cách giá trị rời rạc, trong trường hợp dữ liệu được biểu diễn bằng các thuộc tính có giá trị liên tục thì cần thực hiện bước rời rạc hóa. Các giải thuật ID3, CART và C4.5 đều áp dụng cách tiếp cận *ăn tham* (greedy) (một thuật toán *không quay lui* (non-backtracking)) để xây dựng cây theo hướng từ trên xuống. Tập dữ liệu huấn luyện sẽ được chia thành các tập nhỏ hơn trong quá trình xây dựng cây theo cơ chế *chia để trị* (devide-and - conquer). Dưới đây mô tả thuật toán xây dựng cây cơ bản chung của các giải thuật này.



Hình 6.1. Phân lớp cho bài toán cho vay vốn của ngân hàng

Thuật toán xây dựng cây quyết định

Đầu vào: Tập D chứa dữ liệu huấn luyện

attribute_list chứa danh sách các thuộc tính ứng cử

Đầu ra: Cây quyết định

Generate_decision_tree ($D, attribute_list$)

1. Tạo một nút gốc N cho cây quyết định
2. If toàn bộ dữ liệu trong D đều thuộc lớp C , return nút N là nút lá có nhãn C
3. If *attribute_list* là rỗng, return nút N với nhãn là lớp xuất hiện nhiều nhất trong D
4. $splitting_attribute = attribute_selection_method(D, attribute_list)$ tìm thuộc tính phân chia tốt nhất
5. Gán cho nút N nhãn là *splitting_attribute*
6. $attribute_list \leftarrow attribute_list \setminus \{splitting_attribute\}$
(loại bỏ thuộc tính *splitting_attribute* khỏi *attribute_list*)

7. For each giá trị j của thuộc tính *splitting_attribute*

7.1. Gọi D_j là tập chứa các phần tử dữ liệu mà thuộc tính *splitting_attribute* có giá trị j

7.2. If D_j là rỗng thì thêm một nút lá N_j cho nút N có nhãn là nhãn phổ biến nhất xuất hiện trong D

7.3. Else gắn cây trả về bởi **Generate_decision_tree** (D_j , *attribute_list*) vào nút N

8. return N

Trong đó, *attribute_list* là tập các thuộc tính mô tả tập dữ liệu huấn luyện D ; *attribute_selection_method* là hàm lựa chọn thuộc tính tốt nhất để phân chia dữ liệu, bản chất nó là giải thuật dựa trên kinh nghiệm (heuristic) để tìm ra thuộc tính nào có khả năng phân biệt được các phần tử dữ liệu trong tập D vào các lớp nhất. Nó dựa trên một độ đo nào đó chẳng hạn *độ lợi thông tin* (information gain), hay *độ đo chỉ số gini* (Gini index) để tìm ra thuộc tính tốt nhất.

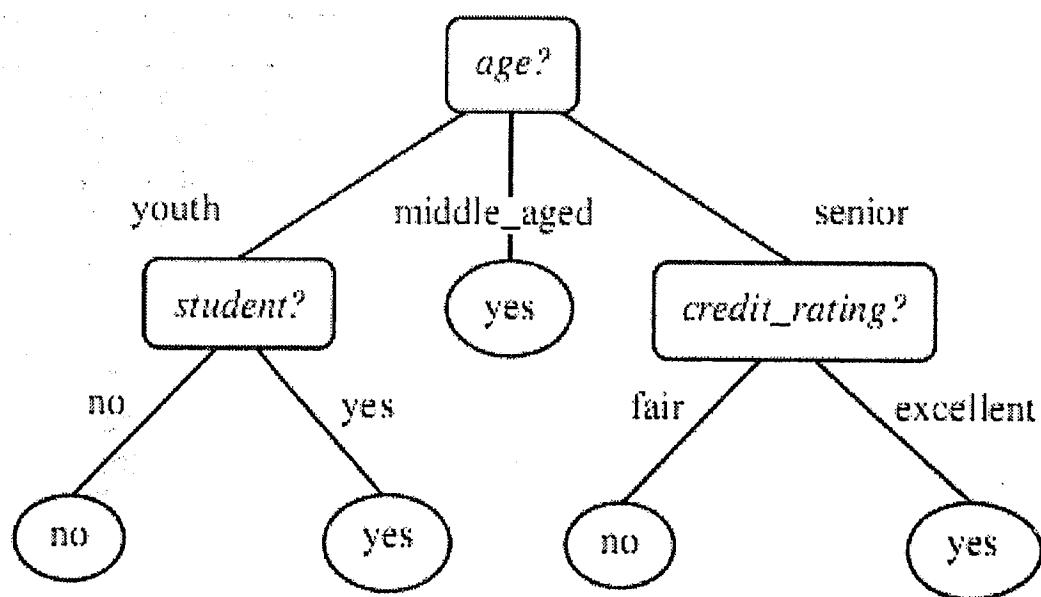
Giải thuật bắt đầu bằng thao tác tạo ra một nút N mô tả tập dữ liệu D (bước 1). Nếu toàn bộ dữ liệu trong D cùng có chung một nhãn lớp thì N sẽ là một nút lá có nhãn là nhãn chung của các phần tử dữ liệu, và thuật toán dừng. Nếu không thì nó sẽ gọi hàm *attribute_selection_method()* để tìm ra thuộc tính tốt nhất dùng để phân chia tập dữ liệu D thành các phần D_j , và nút N sẽ được gán nhãn là thuộc tính tìm được. Giải thuật đệ quy với các tập con dữ liệu D_j . Hình 6.2 minh họa cây quyết định được tạo ra bởi giải thuật trên tập dữ liệu bán hàng (trong bảng 6.1) để tìm ra những loại khách hàng nào có khả năng mua máy tính (*buys_computer*) (*yes* là có mua và *no* là không mua). Độ phức tạp của thuật toán là $O(n \times |D| \times \log(|D|))$, trong đó n là số thuộc tính mô tả tập dữ liệu D , $|D|$ là số lượng các phần tử trong D .

Bảng 6.1. Bảng dữ liệu khách hàng

ID	Tuổi	Thu nhập	Sinh viên	Đánh giá tín dụng	Mua máy tính
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middleaged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middleaged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middleaged	medium	no	excellent	yes
13	middleaged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Trong trường hợp giá trị của một thuộc tính nào đó không phải là giá trị rời rạc (chẳng hạn như thuộc tính tuổi), khi đó một phương pháp rời rạc hóa đã được áp dụng (xem bảng 6.1). Cụ thể nó đã được chia thành 3 loại tuổi rời rạc: trẻ (*youth*), trung niên (*middle_age*) và già (*senior*).

Điểm mấu chốt trong giải thuật xây dựng cây quyết định ở trên là hàm lựa chọn thuộc tính tốt nhất để phân chia dữ liệu. Phần tiếp theo sẽ trình bày một số độ đo dùng để đánh giá “chất lượng” của các thuộc tính.



Hình 6.2. Minh họa cây quyết định

6.2.1. Độ lợi thông tin

Độ lợi thông tin (information gain) là độ đo được sử dụng trong giải thuật ID3. Đầu tiên là công thức đo lượng thông tin kỳ vọng để phân lớp một phần tử trong tập dữ liệu D được đo bằng công thức sau:

$$Info(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (6.1)$$

trong đó, p_i là xác suất một phần tử dữ liệu trong D thuộc vào lớp C_i và nó được ước lượng bằng công thức $p_i = \frac{|D_i|}{|D|}$, với D_i là tập các phần tử dữ liệu trong D thuộc vào lớp C_i ; m là số lượng các lớp trong D . Hàm logarit cơ số 2 được sử dụng là do công thức trên đo lượng thông tin theo đơn vị bit (theo lý thuyết thông tin của C. Shannon). Hàm $Info(D)$ còn được gọi là entropy của D .

Bây giờ giả sử ta phân chia dữ liệu trong D theo thuộc tính A nào đó, và giả sử thuộc tính này có v giá trị (rời rạc) khác nhau là $\{a_1, a_2, \dots, a_v\}$. Thuộc tính này chia tập dữ liệu D thành v tập con $\{D_1, D_2, \dots, D_v\}$ trong đó D_j là tập các phần tử dữ liệu có giá trị của

thuộc tính A là a_i . Tập con này sẽ tương ứng với một nhánh cây được phát triển từ nút N trong giải thuật tạo cây quyết định. Trường hợp lý tưởng thì ta muốn tập con này sẽ có khả năng phân lớp chính xác các phần tử trong nó, hay nói một cách khác ta muốn tập con này càng đồng nhất (pure) càng tốt, đồng nhất ở đây có thể hiểu là các phần tử trong tập con này đều cùng thuộc về một lớp. Tuy nhiên trong thực tế thì các tập này thường không đồng nhất (impure) vì nó chứa các phần tử dữ liệu thuộc về các lớp khác nhau, do đó chúng ta cần thêm thông tin để phân lớp chính xác tập con này. Lượng thông tin này được đo bởi:

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (6.2)$$

trong đó, $\frac{|D_j|}{|D|}$ được dùng làm trọng số của tập con D_j . Giá trị của

$Info_A(D)$ là lượng thông tin kỳ vọng để phân lớp một phần tử dữ liệu trong D dựa trên việc chia dữ liệu bằng thuộc tính A . Giá trị này càng nhỏ thì độ đồng nhất của các tập con càng cao. Cuối cùng hàm đo độ lợi thông tin được tính bằng công thức:

$$Gain(A) = Info(D) - Info_A(D) \quad (6.3)$$

Giá trị $Gain(A)$ cho chúng ta biết ta được lợi bao nhiêu nếu chia dữ liệu theo thuộc tính A . Giá trị này càng lớn thì càng tốt, do đó thuộc tính nào có giá trị $Gain()$ lớn nhất sẽ được chọn để phân nhánh trong quá trình xây dựng cây quyết định.

Để minh họa cho độ đo này ta tính toán một thuộc tính trên tập dữ liệu ở bảng 6.1. Trong bảng này trường cuối cùng là nhãn của dữ liệu (Mua máy tính), nó có 2 giá trị, do đó số lớp ở đây là 2. Có 9 phần tử dữ liệu có nhãn là *yes* và 5 phần tử dữ liệu có nhãn là *no*, do đó theo công thức (6.1) ta có:

$$Info(D) = -\frac{9}{14} \log_2 \left(\frac{9}{14} \right) - \frac{5}{14} \log_2 \left(\frac{5}{14} \right) = 0.94 \text{ bits}$$

Tiếp đến theo công thức (6.2) ta tính giá trị của hàm cho thuộc tính tuổi (age):

$$\begin{aligned}
 Info_{age}(D) &= \frac{5}{14} \times \left(-\frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) \\
 &\quad + \frac{4}{14} \times \left(-\frac{4}{4} \log_2 \frac{4}{4} - \frac{0}{4} \log_2 \frac{0}{4} \right) \\
 &\quad + \frac{5}{14} \times \left(-\frac{3}{5} \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5} \right) \\
 &= 0.694 \text{ bits}
 \end{aligned}$$

Tiếp đến theo công thức 6.3 ta có độ lợi thông tin theo thuộc tính tuổi sẽ là:

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246 \text{ bits}$$

Tương tự ta có thể tính được giá trị độ lợi thông tin cho các thuộc tính thu nhập (income), sinh viên (student) và đánh giá tín dụng (credit_rating) $Gain(income) = 0.029$ bits, $Gain(student) = 0.151$ bits và $Gain(credit_rating) = 0.048$ bits. Từ kết quả này, chúng ta thấy thuộc tính tuổi sẽ được chọn để phân chia dữ liệu. Lặp lại quá trình xây dựng cây tương ứng với các tập con dữ liệu (đã bỏ đi thuộc tính tuổi) ta sẽ thu được cây quyết định như hình 6.2.

6.2.2. Tỉ số độ lợi

Độ đo độ lợi thông tin hoạt động không tốt trong trường hợp một thuộc tính có nhiều giá trị. Ví dụ, thuộc tính mã sản phẩm (product_ID), hay mã giao dịch sẽ có rất nhiều giá trị. Đặc biệt nữa, khi chia dữ liệu theo thuộc tính này thì mỗi một tập con dữ liệu sẽ chỉ có tương ứng một bản ghi, do đó các tập con này là hoàn toàn đồng nhất. Hay nói một cách khác, lượng thông tin cần để phân lớp tập dữ liệu D dựa trên cách phân chia dữ liệu trên thuộc tính này $Info_{Product_ID}(D) = 0$. Và giá trị độ lợi thông tin sẽ đạt giá trị tối đa $Gain(Product_ID) = Info(D) - Info_{Product_ID}(D) = Info(D)$.

Nhưng rõ ràng việc phân lớp dựa trên thuộc tính này là vô nghĩa.

Do đó, trong giải thuật C4.5 (hậu duệ của giải thuật ID3) tác giả đã đề xuất sử dụng một độ đo mới gọi là tỉ số độ lợi (gain ratio) để cố tránh nhược điểm trên. Hàm này sử dụng một phương pháp chuẩn hóa độ lợi thông tin bằng cách sử dụng giá trị *phân chia*

thông tin (split information) được định nghĩa tương tự như hàm $Info(D)$ như sau:

$$SplitInfo_A(D) = -\sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right) \quad (6.4)$$

Giá trị này biểu diễn thông tin tiềm năng được sinh ra thông qua việc chia tập dữ liệu huấn luyện D thành v tập con tương ứng với các giá trị của thuộc tính A . Chú ý rằng với mỗi giá trị của thuộc tính j , nó tính toán số lượng các phần tử có giá trị thuộc tính A là j trên tổng số lượng phần tử của D . Đây là điểm khác so với độ lợi thông tin, do đó công thức tính tỉ số độ lợi sẽ là:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \quad (6.5)$$

trong đó, hàm $SplitInfo_A(D)$ được viết ngắn gọn thành $SplitInfo(A)$. Dựa trên độ đo này, các thuộc tính có giá trị tỉ số độ lợi cao sẽ được chọn làm thuộc tính phân chia dữ liệu. Có một chú ý rằng, nếu hàm $SplitInfo(A) = 0$ thì công thức trên không dùng được, do đó có thêm ràng buộc để tránh trường hợp này. Cụ thể giá trị độ lợi thông tin của thuộc tính được chọn phải đủ lớn, ít nhất là lớn hơn giá trị trung bình độ lợi thông tin của tất cả các thuộc tính.

Trở lại bảng dữ liệu 6.1, ta tính tỉ số độ lợi cho thuộc tính thu nhập (income). Đầu tiên ta sử dụng công thức (6.4) để tính $SplitInfo_{income}(D)$

$$\begin{aligned} SplitInfo_{income}(D) &= -\frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) - \frac{6}{14} \times \log_2 \left(\frac{6}{14} \right) - \frac{4}{14} \times \log_2 \left(\frac{4}{14} \right) \\ &= 0.962 \end{aligned}$$

$$\text{Do đó } GainRatio(income) = \frac{Gain(income)}{SplitInfo(income)} = \frac{0.029}{0.962} = 0.031$$

6.2.3. Chỉ số Gini

Đây là độ đo được sử dụng trong giải thuật CART, chỉ số Gini đo độ không đồng nhất của một tập dữ liệu D bằng công thức:

$$Gini(D) = 1 - \sum_{i=1}^m p_i^2 \quad (6.6)$$

trong đó, p_i có ý nghĩa giống như công thức 6.1; m là số lượng lớp trong D . Chỉ số Gini quan tâm đến trường hợp ta sử dụng một thuộc tính và chia dữ liệu thành 2 nửa. Để đơn giản, ta xét trường hợp thuộc tính A có v giá trị khác nhau $\{a_1, a_2, \dots, a_v\}$ xuất hiện trong D . Để xác định cách phân chia tốt nhất ta xét toàn bộ các tập con của D phân chia theo các giá trị của A . Do đó nếu A có v giá trị khác nhau thì ta sẽ có 2^v tập con của D . Ví dụ thuộc tính thu nhập (income) có 3 giá trị $\{low, medium, high\}$ thì các tập con có thể sẽ là $\{low, medium, high\}, \{low, medium\}, \{medium, high\}, \{low, high\}, \{low\}, \{medium\}, \{high\}$ và tập rỗng \emptyset . Chúng ta không xét 2 tập con $\{low, medium, high\}$ và \emptyset vì nó không chia dữ liệu ra 2 tập, do đó ta có tổng số $2^v - 2$ cách để chia tập dữ liệu D thành 2 tập con dựa trên thuộc tính A . Khi chia tập dữ liệu D thành 2 nửa D_1 và D_2 chúng ta xem xét độ không đồng nhất (impurity) của dữ liệu trong 2 nửa này:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \quad (6.7)$$

Trong trường hợp thuộc tính A có giá trị liên tục thì chúng ta phải xác định các điểm (giá trị) *split_point* để chia tập dữ liệu D thành 2 tập con. Các điểm *split_point* có thể lấy là giá trị trung bình giữa 2 giá trị gần nhau nhất của thuộc tính A . Khi xác định được điểm chia dữ liệu *split_point* ta có thể chia dữ liệu D thành 2 tập dữ liệu con là D_1 và D_2 sao cho: $D_1 = \{X \in D \mid x_A \leq split_point\}$ và $D_2 = \{X \in D \mid x_A > split_point\}$ trong đó x_A là giá trị của thuộc tính A . Khi đó ta định nghĩa độ giảm của độ bất đồng nhất của dữ liệu khi chia dữ liệu thành 2 tập con theo thuộc tính A :

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \quad (6.8)$$

Do đó cách phân chia nào mà tạo ra 2 tập con có giá trị $\Delta Gini(A)$ lớn nhất (hay $Gini_A(D)$ nhỏ nhất) sẽ được chọn. Tuy nhiên trong trường hợp này khác với các độ đo trước, ta cần kết hợp cách phân chia hay giá trị điểm phân chia (*split point*) với thuộc tính để dùng làm điều kiện phân nhánh cây quyết định.

Quay lại cơ sở dữ liệu khách hàng ở bảng 6.1, ta có 9 phần tử dữ liệu thuộc vào lớp C_{yes} và 5 phần tử dữ liệu thuộc vào lớp C_{no} do đó chỉ số $Gini(D)$ đo độ bất đồng nhất trong D là:

$$Gini(D) = 1 - \left(\frac{9}{14} \right)^2 - \left(\frac{5}{14} \right)^2 = 0.459$$

Tiếp theo ta xét thuộc tính thu nhập (income), bắt đầu bằng cách phân chia $\{low, medium\}$ và $\{high\}$. Với cách phân chia này thì ta có tập D_1 chứa 10 phần tử dữ liệu có thuộc tính income có giá trị nằm trong tập $\{low, medium\}$ và tập D_2 chứa 4 phần tử có giá trị $income=high$. Khi đó chỉ số Gini sẽ được tính toán là:

$$\begin{aligned} Gini_{income \in \{low, medium\}}(D) &= \left(\frac{10}{14} \right) Gini(D_1) + \left(\frac{4}{14} \right) Gini(D_2) \\ &= \frac{10}{14} \left(1 - \left(\frac{6}{10} \right)^2 - \left(\frac{4}{10} \right)^2 \right) \\ &\quad + \frac{4}{14} \left(1 - \left(\frac{1}{4} \right)^2 - \left(\frac{3}{4} \right)^2 \right) \\ &= 0.45 = Gini_{income \in \{high\}}(D) \end{aligned}$$

Tương tự, giá trị Gini cho cách chia $\{medium, high\}$ và $\{low\}$ là 0.3; giá trị Gini cho cách chia $\{low, high\}$ và $\{medium\}$ là 0.315. Do đó cách chia $\{medium, high\}$ và $\{low\}$ sẽ được chọn làm điều kiện để phân nhánh cây quyết định vì nó cho ta giá trị Gini nhỏ nhất. Với thuộc tính tuổi (age) thì cách phân chia $\{youth, senior\}$ và $\{middle_age\}$ cho giá trị tốt nhất là 0.375. Với thuộc tính *student* và *credit_rating* đều là giá trị nhị phân nên chúng ta chỉ có một cách chia duy nhất, giá trị Gini của 2 thuộc tính này lần lượt là 0.367 và 0.429. Qua kết quả này ta thấy thuộc tính *income* cho giá trị Gini nhỏ nhất do đó nó sẽ được chọn để làm điều kiện phân nhánh cây quyết định, khác với 2 độ đo ở trên chọn thuộc tính tuổi làm điều kiện phân nhánh đầu tiên. Một điều chú ý là với độ đo này thì ta không chỉ quan tâm đến thuộc tính dùng để phân chia tập dữ liệu mà còn quan tâm đến cách chia dữ liệu theo thuộc tính đó.

Ngoài các độ đo này còn có nhiều độ đo khác, tuy nhiên trong khuôn khổ cuốn giáo trình này ta sẽ không trình bày hết.

6.2.4. Tỉa cây quyết định

Sau khi cây được xây dựng, nó có thể chứa nhiều nhánh phản ánh sự bất thường trong dữ liệu huấn luyện: có thể là các trường hợp ngoại lệ, dữ liệu lỗi hay là dữ liệu nhiễu. Hiện tượng trên cũng gây ra hệ quả là xảy ra hiện tượng cây thu được quá phù hợp dữ liệu (overfitting). Để giải quyết vấn đề này phương pháp *tỉa cây* (tree pruning) được đề xuất. Phương pháp tỉa cây về bản chất là loại bỏ các nhánh ít tin cậy nhất, do đó ta không những thu được cây có khả năng phân lớp tốt hơn mà còn làm cho cây cô đọng hơn và tốc độ xử lý sẽ nhanh hơn. Phương pháp tỉa cây được chia thành 2 loại: tỉa trước (prepruning) cây và tỉa sau (postpruning). Trong phương pháp tỉa trước, cây sẽ được tỉa ngay trong giai đoạn xây dựng cây, nó sẽ tương ứng với các điều kiện để dừng phát triển một nhánh nào đó. Còn phương pháp tỉa sau lại xử lý cây sau khi nó đã được xây dựng hoàn chỉnh. Trong nội dung cuốn giáo trình này sẽ không đi sâu vào các phương pháp tỉa cây, độc giả có thể tham khảo ở tài liệu [Han06].

6.3. THUẬT TOÁN PHÂN LỚP NAIVE BAYES

Bộ phân lớp Bayes là một giải thuật thuộc lớp giải thuật phân lớp thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Phân lớp Bayes được dựa trên định lý Bayes (định lý được đặt theo tên tác giả của nó là Thomas Bayes).

6.3.1. Định lý Bayes

Gọi X là một chứng cứ (evidence) (trong bài toán phân lớp thì X sẽ là một phần tử dữ liệu), H là một giả thiết nào đó cho X thuộc về một lớp một lớp C nào đó. Trong bài toán phân lớp chúng ta muốn xác định giá trị $P(H | X)$ là xác suất để giả thiết H là đúng

với chứng cứ X thuộc vào lớp C với điều kiện ta biết các thông tin mô tả X . $P(H | X)$ là một xác suất hậu nghiệm (posterior probability hay posteriori probability) của H với điều kiện X .

Giả sử tập dữ liệu khách hàng của chúng ta được mô tả bởi các thuộc tính tuổi và thu nhập, và một khách hàng X có tuổi là 35 và thu nhập là \$40000. Giả sử H là giả thiết khách hàng đó sẽ mua máy tính, thì $P(H | X)$ phản ánh xác suất người dùng X sẽ mua máy tính với điều kiện ta biết tuổi và thu nhập của người đó.

Ngược lại $P(H)$ là xác suất tiền nghiệm (prior probability hay priori probability) của H . Trong ví dụ trên, nó là xác suất một khách hàng sẽ mua máy tính mà không cần biết các thông tin về tuổi hay thu nhập của họ. Hay nói cách khác, xác suất này không phụ thuộc vào X . Tương tự, $P(X | H)$ là xác suất của X với điều kiện H , nó là một xác suất hậu nghiệm. Ví dụ, nó là xác suất người dùng X (có tuổi là 35 và thu nhập là \$40000) sẽ mua máy tính với điều kiện ta đã biết là người dùng đó sẽ mua máy tính. Cuối cùng $P(X)$ là xác suất tiền nghiệm của X . Trong ví dụ trên, nó sẽ là xác suất một người trong tập dữ liệu sẽ có tuổi 34 và thu nhập \$40000. Các xác suất này sẽ được tính dựa vào định lý Bayes như sau:

$$P(H | X) = \frac{P(X | H)P(H)}{P(X)} \quad (6.9)$$

6.3.2. Phân lớp Naive Bayes

Bộ phân lớp Naive Bayes hay bộ phân lớp Bayes đơn giản (simple Bayes classifier) hoạt động như sau:

- 1) Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu X được biểu diễn bằng một vector chứa n giá trị thuộc tính A_1, A_2, \dots, A_n , $X = \{x_1, x_2, \dots, x_n\}$.
- 2) Giả sử có m lớp C_1, C_2, \dots, C_m ; Cho một phần tử dữ liệu X , bộ phân lớp sẽ gán nhãn cho X là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán X thuộc vào lớp C_i nếu và chỉ nếu:

$$P(C_i | X) > P(C_j | X) \quad (1 \leq i \leq m, i \neq j) \quad (6.10)$$

Giá trị này sẽ được tính dựa vào định lý Bayes:

$$P(C_i | X) = \frac{P(X | C_i)P(C_i)}{P(X)} \quad (6.11)$$

- 3) Để tìm giá trị xác suất lớn nhất, ta nhận thấy trong công thức (6.10) thì giá trị $P(X)$ là giống nhau với mọi lớp nên ta không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của $P(X | C_i) \times P(C_i)$. Chú ý rằng $P(C_i)$ được ước lượng bằng công thức $P(C_i) = \frac{|D_i|}{|D|}$, trong đó D_i là tập các phần tử dữ

liệu thuộc vào lớp C_i . Nếu xác suất tiên nghiệm $P(C_i)$ cũng không xác định được thì ta coi chúng bằng nhau $P(C_1) = P(C_2) = \dots = P(C_m)$, khi đó ta chỉ cần tìm giá trị $P(X | C_i)$ lớn nhất.

- 4) Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toán $P(X | C_i)$ là rất lớn, do đó để làm giảm độ phức tạp, giải thuật Naive Bayes giả thiết các thuộc tính là độc lập nhau hay không có sự phụ thuộc nào giữa các thuộc tính. Khi đó ta có thể tính:

$$P(X | C_i) = \prod_{k=1}^n P(x_k | C_i) = P(x_1 | C_i) \times \dots \times P(x_n | C_i) \quad (6.12)$$

Chúng ta có thể ước lượng $P(x_1 | C_i), P(x_2 | C_i), \dots, P(x_n | C_i)$ từ tập dữ liệu huấn luyện với x_k là giá trị của thuộc tính A_k của phần tử dữ liệu X . Để tính toán xác suất $P(X | C_i)$, thì tùy vào giá trị của các thuộc tính mà ta sẽ sử dụng các phương pháp tính toán khác nhau:

- a) Nếu các A_k được biểu diễn bằng các giá trị rời rạc thì $P(x_k | C_i) = \frac{|D_i^k|}{|D_i|}$, trong đó D_i^k là tập các phần tử trong D_i có giá trị của thuộc tính A_k bằng x_k .
- b) Nếu các A_k được biểu diễn bằng các giá trị liên tục, khi đó ta giả thiết nó tuân theo phân bố chuẩn Gauss với giá trị trung bình m và độ lệch chuẩn σ và hàm mật độ g được định nghĩa như sau:

$$g(x, m, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (6.13)$$

và xác suất $P(x_k | C_i)$ được tính bằng công thức:

$$P(x_k | C_i) = g(x_k, \mu_{C_i}^k, \sigma_{C_i}^k) \quad (6.14)$$

trong đó, $\mu_{C_i}^k, \sigma_{C_i}^k$ là giá trị trung bình (mean) và độ lệch chuẩn (standard deviation) của thuộc tính A_k với điều kiện là thuộc lớp C_i . Gọi D_i là tập hợp các phần tử dữ liệu thuộc vào lớp C_i ($D_i \subseteq D$), khi đó giá trị trung bình và độ lệch chuẩn của các thuộc tính A_k của lớp C_i được tính như sau:

$$\mu_{C_i}^k = \frac{\sum_{x_j \in D_i} x_k^j}{|D_i|} \quad (6.15)$$

$$\sigma_{C_i}^k = \sqrt{\frac{1}{|D_i|} \sum_{x_j \in D_i} (x_k^j - \mu_{C_i}^k)^2} \quad (6.16)$$

Quay lại cơ sở dữ liệu khách hàng ở bảng 6.1, giả sử ta có một khách hàng mới X có các giá trị thuộc tính là:

$X = (\text{age} = \text{youth}, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$

Bây giờ cần xác định xem khách hàng X có thuộc lớp C_{yes} (mua máy tính) hay không, ta tính toán như sau:

$$P(C_{yes}) = 9/14 = 0.643; P(C_{no}) = 5/14 = 0.357;$$

Trước khi tính xác suất $P(X | C_i)$, ta tính các xác suất thành phần:

$$P(\text{age} = \text{youth} | C_{yes}) = 2/9 = 0.222$$

$$P(\text{age} = \text{youth} | C_{no}) = 3/5 = 0.600$$

$$P(\text{income} = \text{medium} | C_{yes}) = 4/9 = 0.444$$

$$P(\text{income} = \text{medium} | C_{no}) = 2/5 = 0.400$$

$$P(\text{student} = \text{yes} | C_{yes}) = 6/9 = 0.667$$

$$P(\text{student} = \text{yes} | C_{no}) = 1/5 = 0.200$$

$$P(\text{credit_rating} = \text{fair} \mid C_{\text{yes}}) = 6/9 = 0.667$$

$$P(\text{credit_rating} = \text{fair} \mid C_{\text{no}}) = 2/5 = 0.400$$

Cuối cùng ta có xác suất $P(X \mid C_i)$:

$$P(X \mid C_{\text{yes}}) = P(\text{age} = \text{youth} \mid C_{\text{yes}}) * P(\text{income} = \text{medium} \mid C_{\text{yes}}) *$$

$$P(\text{student} = \text{yes} \mid C_{\text{yes}}) * P(\text{credit_rating} = \text{fair} \mid C_{\text{yes}})$$

$$= 0.222 * 0.444 * 0.667 * 0.667 = 0.044$$

$$P(X \mid C_{\text{no}}) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019.$$

$$P(X \mid C_{\text{yes}}) * P(C_{\text{yes}}) = 0.044 * 0.643 = 0.028$$

$$P(X \mid C_{\text{no}}) * P(C_{\text{no}}) = 0.019 * 0.357 = 0.007$$

Từ kết quả này ta thấy $P(X \mid C_{\text{yes}}) * P(C_{\text{yes}})$ có giá trị lớn nhất, do đó thuật toán Bayes sẽ kết luận là khách hàng X sẽ mua máy tính.

Trong quá trình tính toán công thức (6.12), ta có thể gặp trường hợp $P(x_k \mid C_i) = 0$. Ví dụ trong trường hợp thuộc tính A_k là giá trị rác rưởi thì giá trị $P(x_k \mid C_i)$ được tính theo công thức $P(x_k \mid C_i) = \frac{|D_i^k|}{|D_i|}$, khi $|D_i^k| = 0$ thì $P(x_k \mid C_i) = 0$. Điều này có nghĩa là $P(X \mid C_i)$ theo công thức (6.12) sẽ có giá trị là 0. Để tránh trường hợp này xảy ra, ta có thể sử dụng công thức ước lượng Laplace (Laplace estimator), công thức Laplace có rất nhiều dạng tùy thuộc vào các bài toán khác nhau, trong trường hợp cụ thể này ta có thể sử dụng công thức:

$$P(x_k \mid C_i) = \frac{1 + |D_i^k|}{|D_i| + m} \quad (6.17)$$

trong đó, m là số lượng lớp, ta có thể nhận thấy ở tử số đã được cộng thêm giá trị 1 nên nó sẽ tránh được trường hợp $P(x_k \mid C_i) = 0$. Một ví dụ cụ thể, giả sử lớp C_{yes} có 1000 phần tử dữ liệu, trong đó không có phần tử nào có giá trị thuộc tính thu nhập $\text{income} = \text{low}$, có 990 phần tử dữ liệu có $\text{income} = \text{medium}$, và 10 phần tử dữ liệu

có $income = high$. Nếu không sử dụng ước lượng Laplace thì xác suất của $P(x_k | C_{yes})$ tương ứng sẽ là: $0/1000 = 0$; $990/1000 = 0.990$ và $10/1000 = 0.010$. Khi sử dụng ước lượng Laplace thì các xác suất sẽ tương ứng là: $1/1003 = 0.001$; $991/1003 = 0.998$ và $11/1003 = 0.011$, như vậy ta đã giải quyết được vấn đề của công thức (6.12).

6.4. THUẬT TOÁN PHÂN LỚP MÁY VECTOR HỖ TRỢ SVM

Tương tự thuật toán Bayes, thuật toán máy vector hỗ trợ (Support Vector Machines – SVM) là một thuộc lớp giải thuật phân lớp thống kê. Nó có khả năng xử lý cả dữ liệu tuyến tính và dữ liệu không tuyến tính. Bản chất của giải thuật này là nó xây dựng một siêu phẳng để phân chia dữ liệu thành 2 nửa. Trong trường hợp nếu dữ liệu là không tuyến tính thì nó sẽ sử dụng một hàm nhân (kernel function) để chuyển đổi tập dữ liệu ban đầu sang một không gian mới có số chiều lớn hơn để xử lý.

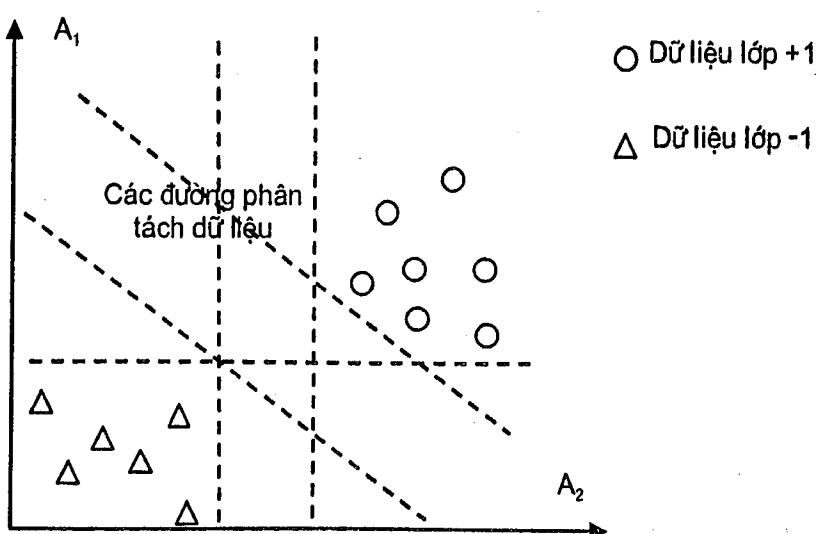
6.4.1. Trường hợp dữ liệu có thể phân loại tuyến tính

Để làm quen với thuật toán phân lớp SVM ta xét trường hợp đơn giản là tập dữ liệu huấn luyện chỉ có 2 lớp và nó phân bố ở dạng mà ta có thể phân tách chúng một cách tuyến tính. Gọi D là tập dữ liệu huấn luyện: $(X_1, y_1), (X_2, y_2), \dots, (X_{|D|}, y_{|D|})$, trong đó X_i là các phần tử dữ liệu và y_i là nhãn tương ứng của nó. Giá trị của y_i có thể nhận là một trong 2 giá trị $\{-1, +1\}$ giống như tập dữ liệu trong bảng 6.1 có 2 lớp cho trường mua máy tính là *yes* hay *no*. Để có thể hiển thị được dữ liệu ta lấy trường hợp dữ liệu được biểu diễn bằng 2 thuộc tính A_1 và A_2 , và các phần tử dữ liệu của tập D được minh họa bằng hình 6.3. Từ hình vẽ cho chúng ta thấy dữ liệu có thể phân tách thành 2 nửa bằng một đường thẳng. Tuy nhiên, số lượng các đường thẳng có thể dùng để phân tách tập dữ liệu trên thành 2 nửa là vô hạn (hình 6.3 minh họa một số đường thẳng vẽ bằng đường đứt nét có thể dùng để phân tách dữ liệu thành 2 lớp riêng biệt). Trong trường hợp dữ liệu được biểu diễn

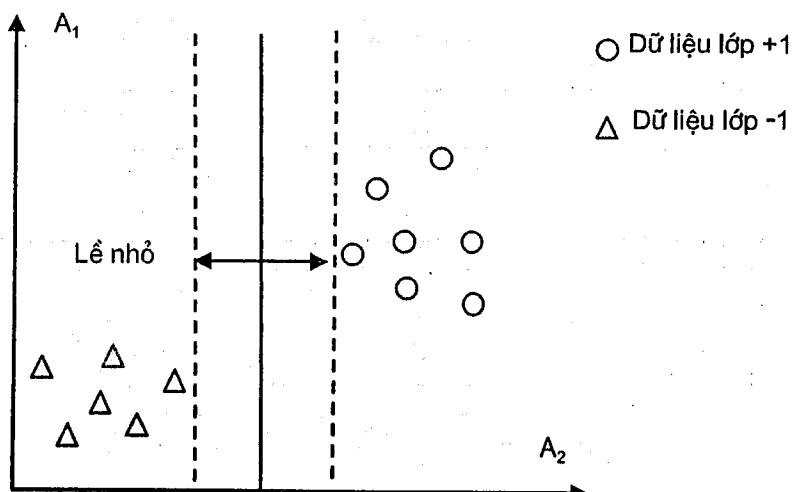
bảng 8 thuộc tính (3 chiều) thì đường thẳng sẽ được thay thế bằng mặt phẳng (plane), và trường hợp tổng quát (n chiều) thì ta dùng siêu phẳng (hyperplane) để thay thế đường thẳng. Ta sẽ dùng thuật ngữ siêu phẳng từ đoạn này về sau trong thuật toán SVM.

Để tìm ra siêu phẳng tốt nhất, giải thuật SVM tìm *siêu phẳng có lề lớn nhất* (maximum marginal hyperplane - MMH). Khái niệm lề có thể được minh họa trên hình 6.4, lề của siêu phẳng h là tổng khoảng cách từ h đến 2 siêu phẳng là tiếp tuyến với 2 miền dữ liệu (ở hai bên siêu phẳng) và song song với siêu phẳng h . Hay nói một cách khác, lề của siêu phẳng h là tổng khoảng cách của 2 phần tử dữ liệu (ở 2 mặt của siêu phẳng) trong tập dữ liệu huấn luyện gần với h nhất. Hình 6.5 minh họa một siêu phẳng khác có lề lớn hơn so với lề của siêu phẳng trong hình 6.4. Lý do của việc tìm siêu phẳng có lề lớn nhất là ta hy vọng nó sẽ có thể phân lớp tốt nhất, nó cho chúng ta tỉ lệ lỗi phân lớp thấp nhất. Một siêu phẳng phân lớp có thể biểu diễn bằng công thức:

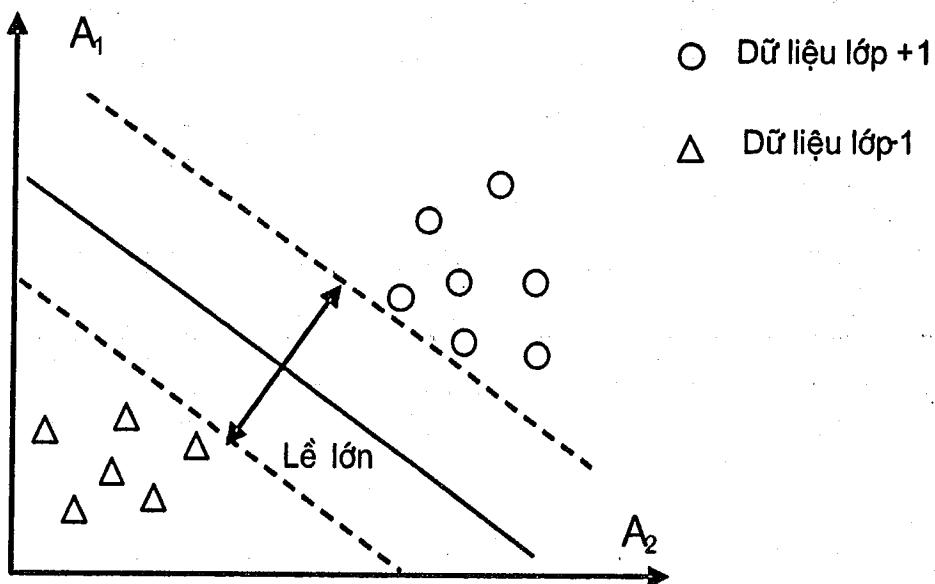
$$W \cdot X + b = 0 \quad (6.18)$$



Hình 6.3. Minh họa dữ liệu có thể phân tách một cách tuyến tính



Hình 6.4. Lề của một siêu phẳng



Hình 6.5. Siêu phẳng có lề lớn

trong đó, W là vector trọng số $W = \{w_1, w_2, \dots, w_n\}$; và n là số lượng các thuộc tính mô tả tập dữ liệu D ; b là một số thực được gọi là độ lệch. Trong trường hợp đơn giản nhất, ta xét số lượng thuộc tính là 2 ký hiệu A_1 và A_2 . Khi đó phần tử dữ liệu X được biểu diễn bằng $X = (x_1, x_2)$ với x_1, x_2 là giá trị tương ứng của thuộc tính A_1 và A_2 . Nếu ta coi b cũng là một trọng số thì công thức (6.18) sẽ có dạng:

$$w_0 + w_1 x_1 + w_2 x_2 = 0 \quad (6.19)$$

Khi đó các điểm nằm phía trên siêu phẳng sẽ thỏa mãn điều kiện:

$$w_0 + w_1 x_1 + w_2 x_2 > 0 \quad (6.20)$$

Các điểm nằm phía dưới siêu phẳng sẽ thỏa mãn điều kiện:

$$w_0 + w_1 x_1 + w_2 x_2 < 0 \quad (6.21)$$

Hai siêu phẳng tiếp tuyến với dữ liệu và song song với siêu phẳng phân lớp h có thể được biểu diễn bằng công thức:

$$H_1 : w_0 + w_1 x_1 + w_2 x_2 \geq +1, \text{ với } y_i = +1 \text{ và} \quad (6.22)$$

$$H_2 : w_0 + w_1 x_1 + w_2 x_2 \leq -1, \text{ với } y_i = -1 \quad (6.23)$$

Do đó, nói một cách chính xác hơn thì các điểm ở trên siêu phẳng H_1 sẽ được phân vào lớp +1 và các điểm ở dưới siêu phẳng H_2 sẽ được phân vào lớp -1. Bằng cách nhân cả 2 vế của 2 bất đẳng thức (6.22) và (6.23) với y_i ta được bất đẳng thức chung:

$$y_i (w_0 + w_1 x_1 + w_2 x_2) \geq 1, \text{ với } \forall i \quad (6.24)$$

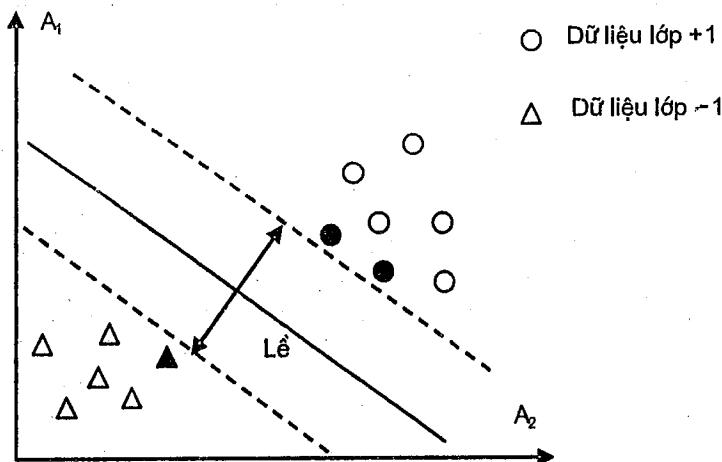
Để xác định 2 siêu phẳng H_1 và H_2 ta chỉ cần dựa vào các phần tử dữ liệu huấn luyện nằm trên 2 siêu phẳng (các phần tử dữ liệu thỏa mãn $y_i (w_0 + w_1 x_1 + w_2 x_2) = 1$).

Các phần tử dữ liệu này được gọi là các vector hỗ trợ (support vector). Chúng cũng chính là các phần tử dữ liệu nằm gần siêu phẳng phân chia h nhất. Hình 6.6 minh họa các vector hỗ trợ (chúng là các hình được bôi đen). Trong trường hợp tổng quát thì các vector hỗ trợ chính là các phần tử khó phân lớp nhất nhưng lại cung cấp nhiều thông tin nhất cho việc phân lớp (giúp ta xác định các siêu phẳng tiếp tuyến). Từ công thức (6.24) ở trên chúng ta có thể suy ra công thức tính độ lớn của lề. Khoảng cách từ một điểm bất kỳ từ siêu phẳng H_1 đến siêu phẳng phân lớp h là $\frac{1}{\|W\|}$, trong đó $\|W\|$ là chuẩn Euclidean của W :

$$\|W\| = \sqrt{W \bullet W} = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2} \quad (6.25)$$

Tương tự khoảng cách từ một điểm bất kỳ từ siêu phẳng H_2 đến siêu phẳng phân lớp h cũng là $\frac{1}{\|W\|}$, và độ lớn của lề sẽ là $\frac{2}{\|W\|}$. Việc tìm ra siêu phẳng có lề lớn nhất người ta dựa vào việc giải công thức (6.24), việc này có thể giải quyết bằng bài toán tối ưu toàn phương lồi

(convex quadratic optimization). Chi tiết cách giải bài toán này sẽ không được trình bày trong khuôn khổ cuốn giáo trình này.



Hình 6.6. Minh họa vector hỗ trợ

Sau khi tìm được siêu phẳng có lề lớn nhất MMH , siêu phẳng này có thể được viết lại dựa trên công thức Lagrangian như sau:

$$f(X^T) = \sum_{i=1}^l y_i \alpha_i X_i X^T + b_0 \quad (6.26)$$

trong đó, y_i là nhãn của các vector hỗ trợ X_i ; X^T là một phần tử dữ liệu kiểm tra; α_i và b_0 là các số thực, chúng là các tham số được xác định thông qua quá trình tối ưu; và l là số lượng các vector hỗ trợ.

Cho một phần tử dữ liệu mới X^T nếu $sign(f(X^T)) = +1$ thì phần tử X^T nằm trên siêu phẳng MMH , SVM sẽ dự đoán nhãn của X^T là $+1$, ngược lại nó sẽ dự đoán X^T thuộc lớp -1 .

6.4.2. Trường hợp dữ liệu không thể phân tách tuyến tính

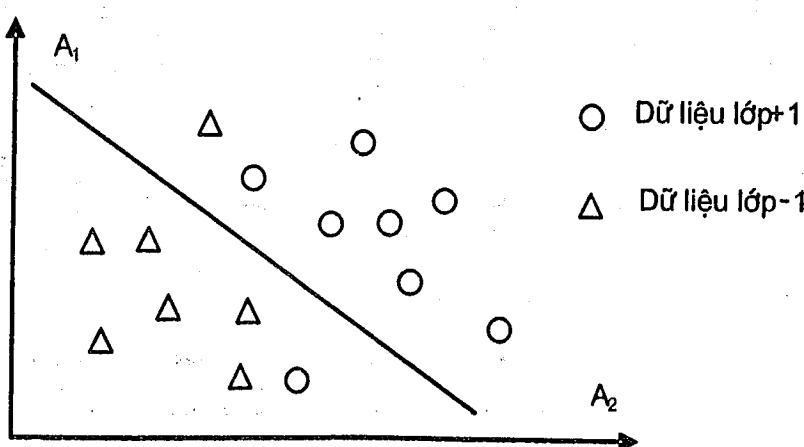
Trong thực tế ta có thể gặp nhiều miền dữ liệu không thể phân tách một cách tuyến tính như trong hình 6.7. Với ví dụ minh họa này, ta thấy không thể tồn tại một siêu phẳng nào có thể phân tách tập dữ liệu (được ký hiệu bằng các hình tròn rỗng và hình tròn được tô đen) thành 2 nửa. Tuy nhiên SVM có thể mở rộng để phân lớp được các *dữ liệu không thể phân tách tuyến tính* (*linearly inseparable data* hay *non-linearly separable data*) hay gọi đơn giản là *dữ liệu không tuyến tính* (*nonlinear data*) hay *dữ liệu phi tuyến*. SVM mở rộng này có khả năng tìm được ranh giới (boundary) phân

lớp, hay siêu diện không tuyến tính (nonlinear hypersurface) (hay siêu diện phi tuyến) từ không gian dữ liệu đầu vào.

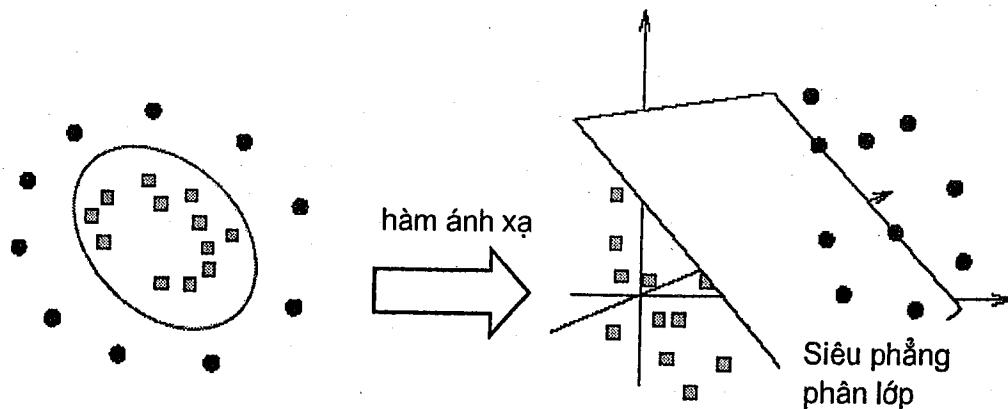
SVM được mở rộng để xử lý dữ liệu phi tuyến theo 2 bước chính như sau:

1. Bước đầu tiên chúng ta chuyển không gian dữ liệu đầu vào thành một không gian dữ liệu có số chiều lớn hơn bằng một ánh xạ không tuyến tính (ánh xạ phi tuyến). Có rất nhiều ánh xạ phi tuyến có thể được sử dụng trong bước này (sẽ được trình bày ở dưới).
2. Khi dữ liệu đã được chuyển sang không gian có số chiều lớn hơn, bước tiếp theo ta tìm siêu phẳng tuyến tính để phân lớp dữ liệu trên không gian mới.

Để minh họa cho phương pháp xử lý của SVM ta có thể xem minh họa trong hình 6.8, trong đó hình 6.8 a) mô tả không gian của dữ liệu đầu vào (nó được biểu diễn bằng không gian 2 chiều), rõ ràng với phân bố dữ liệu như thế này thì ta không thể dùng một siêu phẳng để phân tách 2 lớp thành 2 phần độc lập nhau. Sau khi sử dụng hàm ánh xạ, không gian dữ liệu đầu vào sẽ được chuyển sang không gian mới có số chiều cao hơn (3 chiều), đặc biệt trong không gian dữ liệu mới này, ta có thể sử dụng một siêu phẳng để phân tách dữ liệu thành 2 lớp.



Hình 6.7. Trường hợp dữ liệu không thể phân tách bằng một siêu phẳng



a) Không gian ban đầu (2 chiều) b) Không gian mới (3 chiều)

Hình 6.8. Hàm ánh xạ từ dữ liệu phi tuyến sang dữ liệu tuyến tính

Ví dụ trong một miền dữ liệu 3 chiều, một phần tử dữ liệu sẽ được biểu diễn bằng vector $X = (x_1, x_2, x_3)$, sau khi sử dụng một hàm ánh xạ Φ sang không gian mới có 6 chiều, phần tử X sẽ biến thành Z , sao cho $Z = \Phi(X) = (x_1, x_2, x_3, x_1*x_1, x_1*x_2, x_1*x_3)$. Giả sử sau khi biến đổi, dữ liệu trong không gian mới sẽ có thể phân lớp tuyến tính, và ta có thể dùng một siêu phẳng để phân tách dữ liệu thành 2 nửa, khi đó siêu phẳng h sẽ được biểu diễn bằng công thức $h(Z) = W^*Z + b$, trong đó W là vector trọng số và Z là vector biểu diễn dữ liệu trong không gian mới và b là một số thực giống như công thức biểu diễn siêu phẳng (6.18). Khi diễn giải công thức này ra, ta có công thức biểu diễn siêu phẳng là:

$$\begin{aligned} h(Z) &= w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_1 * x_1 + w_5 z_1 * x_2 + w_6 z_1 * x_3 + b \\ &= w_1 z_1 + w_2 z_2 + w_3 z_3 + w_4 z_4 + w_5 z_5 + w_6 z_6 + b \end{aligned}$$

Tuy chúng ta đã mở rộng thêm sức mạnh của SVM, nhưng chúng ta lại có thêm vấn đề. Cụ thể là độ phức tạp thuật toán sẽ tăng lên bởi vì ta phải sử dụng thêm hàm ánh xạ. Rất may là tồn tại giải pháp cho vấn đề này, chú ý công thức (6.25), ta phải thực hiện phép nhân tích vô hướng $X_i X^T$ (trong đó X_i và X^T đều là các vector trong không gian dữ liệu ban đầu) hay viết $X_i X_j$ cho đơn giản: $X_i \bullet X_j = \sum_k x_{ik} * x_{jk}$, trong đó x_{ik} là các giá trị biểu diễn phần tử dữ liệu X_i và x_{jk} là các giá trị biểu diễn phần tử dữ liệu X_j .

Khi chuyển sang không gian mới, tích vô hướng trên sẽ được tính toán bằng $\Phi(X_i)\Phi(X_j)$ trong đó Φ là hàm ánh xạ. Tuy nhiên, một mẹo toán học rất hay ở đây là, thay vì tính tích vô hướng trên dữ liệu ở không gian dữ liệu mới, thì ta có thể sử dụng một *hàm nhân* (kernel function) K cho kết quả tương tự như sau:

$$K(X_i, X_j) = \Phi(X_i)\Phi(X_j) \quad (6.27)$$

Bằng cách sử dụng hàm tương đương này, thì ở bất kỳ đâu xuất hiện $\Phi(X_i)\Phi(X_j)$ ta thay thế bằng hàm $K(X_i, X_j)$. Do đó, việc tính toán về bản chất sẽ được thực hiện trên không gian dữ liệu ban đầu – không gian có khả năng có số chiều nhỏ hơn nhiều. Sau khi sử dụng hàm nhân thay thế, ta có thể sử dụng thuật toán tìm kiếm siêu phẳng phân lớp mà cũng không cần quan tâm đến ánh xạ biến đổi cụ thể là gì. Các đặc điểm của hàm nhân có thể sử dụng để thay thế hàm nhân tích vô hướng đã được nghiên cứu. Dưới đây xin trình bày một số hàm nhân phổ biến, nó thường được cài đặt trong các gói phần mềm cài đặt giải thuật SVM (chẳng hạn như thư viện libSVM²⁰, hay thư viện Weka²¹):

1. Hàm nhân đa thức cấp h :

$$K(X_i, X_j) = (X_i \bullet X_j + 1)^h \quad (6.28)$$

2. Hàm nhân Gaussian radial cơ bản:

$$K(X_i, X_j) = e^{\|X_i - X_j\|^2 / 2\sigma^2} \quad (6.29)$$

3. Hàm nhân đa sigmoid

$$K(X_i, X_j) = \tanh(\kappa X_i \bullet X_j - \delta) \quad (6.30)$$

Một số hàm nhân khác ta có thể tham khảo và thử nghiệm từ bộ phần mềm cài đặt giải thuật SVM có tên là Accord.NET²².

²⁰ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

²¹ <http://sourceforge.net/projects/weka/>

²² <http://crsouza.blogspot.com/2010/03/kernel-functions-for-machine-learning.html>

Vấn đề thứ 2 là liệu có tồn tại một hàm nhân nào có thể biến các tập dữ liệu phi tuyến bất kỳ sang không gian dữ liệu tuyến tính. Câu trả lời có lẽ là không, tùy vào từng loại dữ liệu mà sẽ có một hoặc một số hàm nhân phù hợp. Trong nhiều trường hợp ta phải chọn thử nhiều hàm nhân khác nhau để chọn ra hàm nhân phù hợp với tập dữ liệu đang xử lý nhất.

6.4.3. Phân lớp đa lớp với SVM

Vấn đề cuối cùng là thuật toán SVM trình bày ở trên chỉ hoạt động với dữ liệu có 2 lớp, trong thực tế số lượng lớp của dữ liệu có thể rất lớn. Rất may là cũng đã có giải pháp để mở rộng SVM cho bài toán phân lớp có nhiều lớp.

Bài toán phân lớp câu hỏi yêu cầu một bộ phân lớp đa lớp do đó cần cải tiến SVM cơ bản (phân lớp nhị phân) thành bộ phân lớp đa lớp.

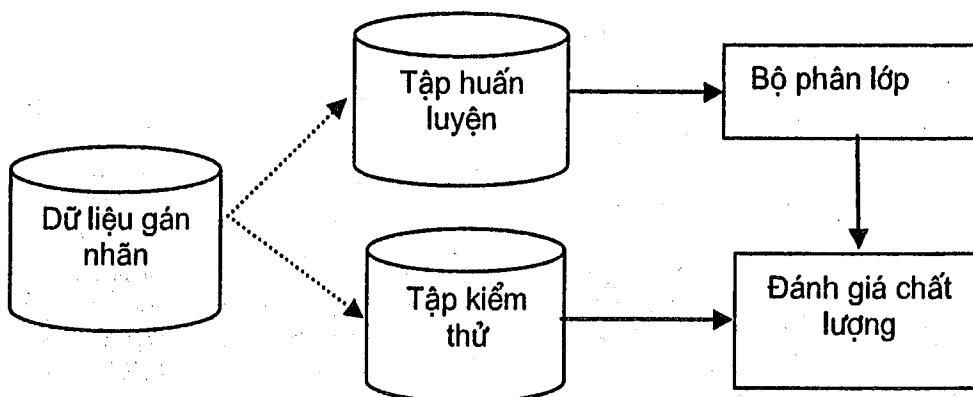
Một trong những phương pháp cải tiến đó là sử dụng thuật toán *1-against-all* [Hsu02, Milgram06]. Ý tưởng cơ bản là chuyển bài toán phân lớp nhiều lớp thành nhiều bài toán phân lớp nhị phân như sau:

- Giả sử tập dữ liệu mẫu $(x_1, y_1), \dots, (x_m, y_m)$ với x_i là một vector n chiều và $y_i \in Y$ là nhãn lớp được gán cho vector x_i (có m nhãn lớp khác nhau)
- Biến đổi tập Y ban đầu thành m tập có hai lớp con $Z_i = \{y_i, \{Y - y_i\}\}$
- Áp dụng SVM phân lớp nhị phân cơ bản với m tập Z_i để xây dựng siêu phẳng cho phân lớp này. Như vậy ta sẽ có m bộ phân lớp nhị phân.

Bộ phân lớp với sự kết hợp của m bộ phân lớp trên được gọi là bộ phân lớp đa lớp mở rộng với SVM. Ngoài ra còn có một giải pháp phân lớp đa lớp khác là one-against-one, độc giả có thể tham khảo chi tiết tại [Hsu02, Milgram06].

6.5. THUẬT TOÁN PHÂN LỚP kNN

Mô hình chung của các thuật toán học có giám sát là giải thuật sẽ phân tích dữ liệu huấn luyện để tìm ra mô hình biểu diễn dữ liệu, sau đó ta có thể dùng một tập dữ liệu khác để kiểm thử độ chính xác của giải thuật như minh họa trên hình 6.9. Như mô tả ở trên hình, tập dữ liệu huấn luyện sẽ được sử dụng để tạo ra mô hình (trong quá trình huấn luyện giải thuật). Có một số giải thuật lại không hề tồn tại giai đoạn học để tạo ra mô hình, mà nó chỉ đơn thuần là sử dụng tập dữ liệu huấn luyện phục vụ cho giai đoạn dự đoán nhãn của dữ liệu sau này. Hay nói một cách khác mô hình của giải thuật thuộc lớp này chính là tập dữ liệu huấn luyện. Những giải thuật này được liệt kê vào lớp *giải thuật lười học* (lazy learner). Đặc điểm của lớp giải thuật này là nó không tồn thời gian để học, tuy nhiên giai đoạn phân lớp của nó lại bị “trả giá”. Thông thường các giải thuật lười học sẽ cần phải tính toán nhiều trong quá trình phân lớp. Có thể đây là nhược điểm lớn nhất của lớp giải thuật lười học, vì khi tập dữ liệu huấn luyện là rất lớn thì chi phí khi phân lớp sẽ càng cao.



Hình 6.9. Các bước trong mô hình học máy có giám sát

Tuy nhiên một trong những ưu điểm của việc “lười học” là nó hỗ trợ xử lý dữ liệu một cách gia tăng (incremental). Cụ thể là với các giải thuật cần phải huấn luyện thì khi dữ liệu huấn luyện thay đổi, thì ta phải huấn luyện lại giải thuật để tạo ra mô hình mới tương ứng với dữ liệu mới. Tuy nhiên với giải thuật lười học thì

cho dù dữ liệu huấn luyện có thay đổi thì cũng không phải mất công huấn luyện.

Một trong những giải thuật thuộc lớp giải thuật lười học là giải thuật *k người láng giềng gần nhất* (*k nearest neighbors*) viết tắt là kNN và giải thuật case-based reasoning. Giáo trình này sẽ trình bày chi tiết giải thuật kNN.

Khi đưa một phần tử dữ liệu mới, giải thuật sẽ tìm *k* phần tử dữ liệu láng giềng gần nó nhất (*k nearest neighbors*), sau đó dựa trên nhãn (lớp) của các láng giềng này mà nó sẽ quyết định nhãn (lớp) của phần tử dữ liệu mới là thuộc lớp nào. Trường hợp đơn giản nhất là ta chỉ tìm một phần tử gần phần tử mới nhất, nhãn của phần tử mới sẽ được gán là nhãn của phần tử tìm được. Để tìm các phần tử láng giềng gần nhất ta cần định nghĩa độ đo nào đó, một trong các độ đo điển hình là độ đo khoảng cách Euclide. Giả sử có 2 phần tử dữ liệu $X_1 = (x_{11}, x_{12}, \dots, x_{1n})$ và $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$, độ đo khoảng cách Euclide được tính bằng công thức:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (6.31)$$

Từ công thức 6.31, ta nhận thấy nếu các thuộc tính khác nhau có miền giá trị khác nhau thì có thể độ chính xác của độ đo sẽ không chính xác. Ví dụ thuộc tính thu nhập có miền giá trị lớn hơn nhiều so với thuộc tính tuổi, hay thuộc tính số lượng con. Khi đó chỉ cần một độ chênh lệch nhỏ của thuộc tính thu nhập cũng làm thay đổi giá trị của độ đo khoảng cách. Để làm cho các thuộc tính có “ảnh hưởng” ngang nhau đến độ đo khoảng cách, ta có thể chuẩn hóa dữ liệu các thuộc tính sử dụng công thức sau để chuyển giá trị v của một thuộc tính A sang giá trị v' có miền giá trị nằm trong khoảng $[0, 1]$:

$$v' = \frac{v - min_A}{max_A - min_A} \quad (6.32)$$

trong đó, min_A và max_A là giá trị nhỏ nhất và lớn nhất của thuộc tính A .

Trường hợp thuộc tính biểu diễn dữ liệu không phải là dữ liệu liên tục mà là dữ liệu rời rạc (chẳng hạn thuộc tính màu có miền giá trị là một danh sách các loại màu). Khi đó ta có thể giải quyết như sau: giả sử x_{1i} và x_{2i} là giá trị rời rạc (biểu diễn thuộc tính A) của 2 phần tử dữ liệu X_1 và X_2 , thì:

$$x_{1i} - x_{2i} = \begin{cases} 0 & \text{khi } x_{1i} = x_{2i} \\ 1 & \text{khi } x_{1i} \neq x_{2i} \end{cases} \quad (6.33)$$

Rõ ràng với công thức này thì ta có thể áp dụng công thức (6.31) với dữ liệu rời rạc. Trong nhiều trường hợp ta cũng có thể sử dụng độ đo tương tự (thay vì độ đo khoảng cách) để tìm ra các phần tử lóng giềng gần nhất.

Vấn đề tiếp theo là xác định giá trị k như thế nào để ta có thể thu được kết quả phân lớp tốt nhất. Với trường hợp đơn giản nhất thì $k = 1$ (khi đó giải thuật kNN sẽ được ký hiệu là 1-NN). Khi xác định được phần tử dữ liệu gần phần tử dữ liệu cần phân lớp nhất thì bài toán xác định nhãn lại rất đơn giản vì nó chính là nhãn của phần tử gần nhất vừa tìm được. Tuy nhiên có một vấn đề khi ta chỉ dựa vào 1 phần tử lóng giềng để quyết định nhãn của phần tử dữ liệu cần phân lớp: đó là trường hợp phần tử lóng giềng gần nó nhất lại là phần tử nhiễu (noise), khi đó nhãn thu được sẽ không chính xác. Để giải quyết vấn đề này thì ta có thể dùng các phương pháp để lọc các dữ liệu nhiễu, thậm chí là các thuộc tính nhiễu đi.

Tuy nhiên cũng có một giải thuật mở rộng của giải thuật 1-NN, đó là tăng giá trị của k lên để tạo khả năng ra quyết định dựa trên nhiều phần tử dữ liệu. Thông thường các giá trị của k được chọn sẽ là các giá trị lẻ (để tránh trường hợp các lóng giềng của phần tử dữ liệu cần phân lớp lại thuộc 2 lớp khác nhau, và số lượng các lóng giềng trong mỗi lớp lại bằng nhau). Với $k = 3$ và có 3 phần tử dữ liệu lóng giềng gần nhất có nhãn là $\{A, B, A\}$, khi đó ta có thể kết luận là phần tử dữ liệu cần phân lớp là thuộc lớp A. Với $k = 5$, các phần tử lóng giềng có nhãn là $\{A, B, A, B, B\}$, thì ta có thể kết luận là phần tử dữ liệu mới thuộc lớp B. Tuy nhiên việc phân lớp dựa vào việc đếm số nhãn như thế này sẽ có vấn đề. Cụ thể với trường hợp $k = 5$, và giả sử độ tương tự tương ứng của 5

láng giềng này là $\{0.98, 0.67, 0.56, 0.34, 0.23\}$. Ta có thể nhận thấy các phần tử láng giềng 4 và 5 có độ tương tự rất thấp, do đó nếu ta dựa vào các phần tử dữ liệu này để kết luận nhãn của phần tử dữ liệu mới thuộc lớp A sẽ không tin cậy.

Do đó người ta đề xuất là sử dụng trọng số cho nhãn của các phần tử láng giềng, chúng ta có giải thuật mới có tên là: k người láng giềng gần nhất có đánh trọng số khoảng cách (distance-weighted kNN). Cụ thể nhãn của k láng giềng sẽ được gán trọng số, lớp có tổng trọng số lớn nhất sẽ được dùng để gán cho phần tử cần phân lớp. Trọng số đơn giản chính là độ tương tự giữa phần tử dữ liệu cần phân lớp X với phần tử láng giềng X_i , là $sim(X, X_i)$. Với ví dụ $k = 5$ ở trên thì tổng trọng số của các láng giềng thuộc lớp A là $0.98 + 0.56 = 1.54$, và tổng trọng số các nhãn thuộc lớp B là $0.67 + 0.34 + 0.23 = 1.24$, kết quả này cho ta quyết định là phần tử cần phân lớp thuộc lớp A. Một số công thức tính trọng số khác là: $1/(1-sim(X, X_i))$ hay $1/(1-sim(X, X_i))^2$. Các công thức này đều có đặc điểm chung là giá trị của chúng sẽ tăng lên khi độ tương tự giữa chúng tăng lên. Tuy có rất nhiều đề xuất cải tiến so với giải thuật 1-NN nhưng trong nhiều trường hợp thì 1-NN vẫn tỏ ra là có chất lượng tốt hơn cả.

Một nhược điểm của giải thuật kNN là rất chậm khi kích thước của tập dữ liệu huấn luyện D tăng lên. Ta phải sử dụng $|D|$ phép so sánh để tìm ra các láng giềng gần nhất, hay độ phức tạp của nó là $O(n)$. Có rất nhiều đề xuất để làm giảm độ phức tạp của giải thuật, một số phương pháp được liệt kê ở dưới:

- Sắp xếp tập dữ liệu D đầu vào và tổ chức nó dưới dạng 1 cây tìm kiếm, khi đó độ phức tạp của nó giảm xuống còn $O(\log(n))$.
- Sử dụng các phương pháp song song hóa
- Lấy mẫu tập dữ liệu D để tạo một tập dữ liệu D' có kích thước nhỏ hơn
- Sử dụng 1 phần độ đo khoảng cách (partial distance), việc tính toán khoảng cách chỉ dựa trên một tập con các thuộc

tính, nếu giá trị thu được lớn hơn 1 ngưỡng nào đó thì ta sẽ không tính toán tiếp phần tử dữ liệu hiện tại nữa (vì nó có khoảng cách quá xa), và phần tử dữ liệu tiếp theo sẽ được xử lý.

- Phương pháp hiệu chỉnh (editing): chúng ta loại bỏ các phần tử dữ liệu (đã được chứng minh) là vô nghĩa trong quá trình phân lớp. Phương pháp này còn có các tên khác là tia (pruning) hay cô đọng hóa (condensing) vì chúng làm giảm số lượng phần tử dữ liệu trong tập huấn luyện.

6.6. ĐÁNH GIÁ CÁC GIẢI THUẬT PHÂN LỚP

Như đã đề cập ở trên, trước khi đưa bộ phân lớp vào ứng dụng, chúng ta cần phải biết được độ chính xác của nó có đáp ứng được yêu cầu trong miền dữ liệu cụ thể nào đó hay không. Để tính toán các độ đo đánh giá, ta sử dụng ma trận lẩn lộn như bảng 6.2, trong đó TP (*true positive*) là số lượng các phần tử được dự đoán đúng lớp +1; FN (*false negative*) là số lượng các phần tử đoán nhầm từ -1 sang +1; FP (*false positive*) là số lượng các phần tử bị đoán nhầm từ lớp +1 sang -1; và TN (*true negative*) là số lượng phần tử được dự đoán đúng thuộc lớp -1. Chúng ta có các công thức đánh giá như sau:

- Tỉ lệ lỗi tổng thể:

$$Error = \frac{FP + FN}{TP + FP + TN + FN} \times 100\% \quad (6.34)$$

Bảng 6.2. Ma trận lẩn lộn

		Lớp được dự đoán bởi giải thuật phân lớp	
Lớp thực tế		+1	-1
+1	TP	FN	
	FP		TN

- Độ chính xác tổng thể:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \times 100\% \quad (6.35)$$

Đối với từng lớp thì ta có thể sử dụng thêm 2 độ đo đánh giá sau:

- Độ chính xác (precision): $P = \frac{TP}{TP + FP} \times 100\%$ (6.36)

- Độ hồi tưởng (recall): $R = \frac{TP}{TP + FN} \times 100\%$ (6.37)

- Độ đo tổng hợp (F-measure) $F = \frac{2 \times P \times R}{P + R}$ (6.38)

Trong trường hợp bài toán phân lớp K lớp, các độ đo trung bình được sử dụng bao gồm trung bình mịn (*microaveraging*) và trung bình thô (*macroaveraging*).

- Độ chính xác trung bình thô (macro-averaging precision):

$$P^M = \frac{1}{K} \sum_{c=1}^K P_c \quad (6.39)$$

- Độ hồi tưởng trung bình thô (macro-averaging recall):

$$R^M = \frac{1}{K} \sum_{c=1}^K R_c \quad (6.40)$$

- Độ chính xác trung bình mịn (micro-averaging precision):

$$P^\mu = \frac{\sum_{c=1}^K TP_c}{\sum_{c=1}^K (TP_c + TN_c)} \quad (6.41)$$

- Độ hồi tưởng trung bình mịn (micro-averaging recall):

$$R^\mu = \frac{\sum_{c=1}^K TP_c}{\sum_{c=1}^K (TP_c + FN_c)} \quad (6.42)$$

trong đó, P_c và R_c lần lượt là độ chính xác và độ hồi tưởng của lớp C. Các độ đo trung bình mịn được coi là các độ đo tốt hơn để đánh giá chất lượng thuật toán phân lớp.

Theo mô hình được mô tả trong hình 6.9, tập dữ liệu gán nhãn sẽ được chia làm 2 phần: một dành cho huấn luyện giải thuật và phần còn lại để kiểm thử được. Phương pháp chia đơn giản nhất là lấy ngẫu nhiên khoảng 2/3 dữ liệu làm dữ liệu huấn luyện và phần 1/3 còn lại được dùng làm dữ liệu kiểm thử. Phương pháp chia này gọi là phương pháp holdout. Phương pháp holdout có thể cho chúng ta kết quả kiểm thử không chính xác vì có thể một cách chia nào đó làm cho chất lượng của giải thuật rất tốt, hoặc cũng có thể là rất kém. Lý do là việc lấy ngẫu nhiên có thể làm cho tập dữ liệu huấn luyện không đại diện đúng cho miền dữ liệu ta đang xét: chẳng hạn có trường hợp không có một phần tử dữ liệu thuộc vào lớp A nào đó nằm trong tập huấn luyện, và kết quả là chất lượng của giải thuật kém. Để làm tăng độ chính xác khi đánh giá một giải thuật ta có thêm một số phương pháp sau:

- Phương pháp *lấy mẫu ngẫu nhiên* (random subsampling): Đây là phương pháp mở rộng của phương pháp holdout, ta thực hiện việc chia dữ liệu k lần, trong mỗi lần ta thực hiện việc huấn luyện giải thuật và kiểm thử. Kết quả kiểm thử của giải thuật được tính bằng giá trị trung bình của kết quả kiểm thử trong k lần lặp. Phương pháp này còn có tên khác là Repeated holdout.
- Phương pháp chia theo tỉ lệ (stratification): Trong phương pháp này, ngoài việc chia dữ liệu là ngẫu nhiên, nó còn bổ sung thêm ràng buộc là tỉ lệ dữ liệu của các lớp trong cả tập dữ liệu huấn luyện và kiểm thử là giống nhau.
- Phương pháp *thẩm định chéo k-tập* (k-fold cross-validation): Thay vì chia dữ liệu gán nhãn thành 2 tập (một tập dành cho huấn luyện và tập kiểm thử), tập dữ liệu huấn luyện ban đầu D sẽ được chia ngẫu nhiên thành k tập con (được gọi là fold) không giao nhau: D_1, D_2, \dots, D_k , kích thước của các tập này là xấp xỉ nhau. Quá trình huấn luyện và kiểm thử sẽ được thực hiện (lặp) k lần. Tại mỗi lần lặp thứ i tập dữ liệu D_i sẽ được dùng làm tập dữ liệu kiểm thử và $(k-1)$ tập dữ liệu còn lại sẽ được gộp lại

$\bigcup_j D_j, 1 \leq j \leq k, j \neq i$ làm tập dữ liệu huấn luyện. Việc làm này sẽ đảm bảo tính ngẫu nhiên của dữ liệu, hơn nữa bất kỳ phần tử nào cũng được làm dữ liệu kiểm thử 1 lần và làm dữ liệu huấn luyện trong $(k-1)$ lần. Ta có thể kết hợp phương pháp thẩm định chéo với phương pháp chia theo tỉ lệ để có thể thu được kết quả thẩm định chính xác hơn.

- Phương pháp Leave-one-out: là trường hợp đặc biệt của phương pháp thẩm định chéo k tập, trong đó số tập $k = n$ với n là số lượng các phần tử dữ liệu trong tập D . Với phương pháp này ta thấy chi phí cho việc kiểm thử là rất lớn nên nó không phải là phương pháp đánh giá phổ dụng.

6.7. MỘT SỐ ỨNG DỤNG CỦA CÁC GIẢI THUẬT PHÂN LỚP

Giải thuật phân lớp có lẽ được liệt kê là giải thuật được sử dụng nhiều nhất, hay có tính ứng dụng cao nhất trong thực tế. Dưới đây chỉ xin liệt kê một số ứng dụng của nó:

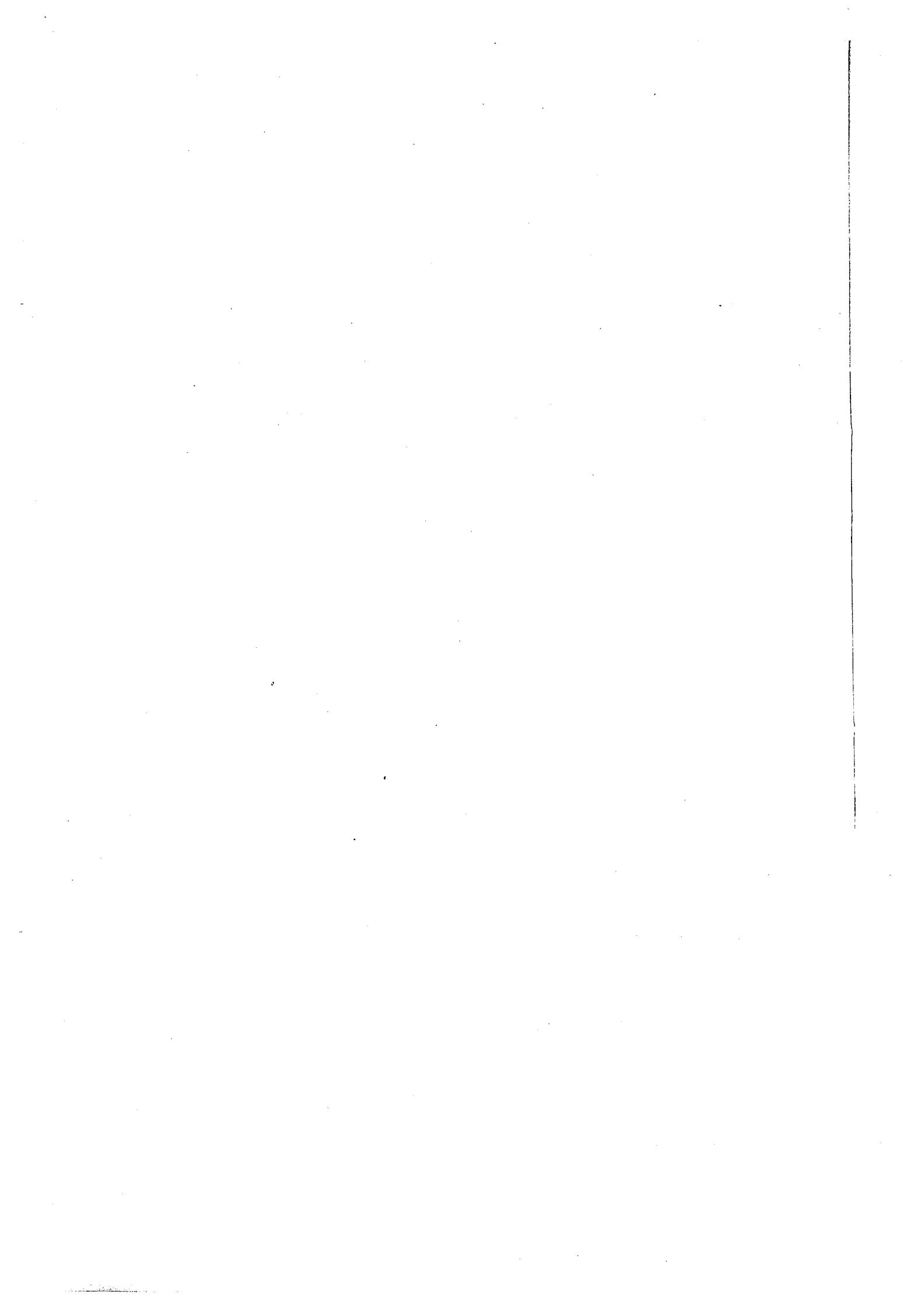
- Trong ngân hàng, khi xem xét hồ sơ của một khách hàng cần vay vốn, nếu ta có thể phân lớp được khách hàng này thuộc lớp “an toàn” hay “mạo hiểm” thì sẽ có ý nghĩa rất quan trọng cho người ra quyết định cho vay vốn.
- Trong chứng khoán, nếu phân lớp được các mã chứng khoán sẽ lên hay xuống thì có thể sẽ là bài toán sống còn đối với một nhà đầu tư.
- Trong các mail server (như gmail hay yahoo), chúng ta vẫn thấy các hệ thống lọc thư rác, nó có khả năng phân loại được các thư rác (spam mail) và đưa vào thùng rác. Chức năng này làm cho người dùng thấy rất thuận tiện và tránh được bức mìn.
- Trong các hệ thống thu thập tin (như trang baomoi.com) thì ta cần một hệ thống tự động phân lớp các bản tin thu được và đưa vào các chuyên mục phù hợp.

- Trong một hệ thống thư viện điện tử, các giải thuật phân lớp rất quan trọng vì nó giúp ta phân loại tự động được các tài liệu vào các lớp phù hợp, từ đó người dùng sẽ tìm ra tài liệu mình quan tâm được dễ dàng hơn.
- Trong quá trình xử lý dữ liệu, các máy tìm kiếm sẽ muốn phát hiện ra các trang rác (spam) để loại bỏ trong quá trình đánh chỉ mục.
- Các dịch vụ trực tuyến (chia sẻ ảnh, tin hay video) rất cần có một hệ phân lớp có khả năng phát hiện ra các bản tin, các hình ảnh hay video có nội dung không phù hợp như các nội dung dung tục, hay không phù hợp với văn hóa, chính trị, ...
- Rất nhiều bài toán trong xử lý ngôn ngữ tự nhiên như phân đoạn (chunking), gán nhãn từ loại (part of speech tagging), thậm chí là nhận dạng thực thể tên (named entity recognition) cũng đều có thể biến đổi thành bài toán phân lớp.

CÂU HỎI VÀ BÀI TẬP

- 6.1.** Tính toán tường minh độ lợi thông tin cho các thuộc tính còn lại không được tính tường minh ở mục 6.2.1.
- 6.2.** Tính toán tường minh tỉ số độ lợi cho các thuộc tính còn lại không được tính tường minh ở mục 6.2.2.
- 6.3.** Tính toán tường minh tỉ số Gini cho các thuộc tính còn lại không được tính tường minh ở mục 6.2.3.
- 6.4.** Dùng bộ phân lớp DecisionTable trong phần mềm weka để phân lớp tập dữ liệu đi kèm và đánh giá sử dụng phương pháp thẩm định chéo (10-folds cross-validation).
- 6.5.** Giả sử trong bảng dữ liệu 6.1, ta lấy dòng đầu tiên làm dữ liệu kiểm thử, toàn bộ các dòng còn lại làm dữ liệu huấn luyện. Dùng thuật toán Naive Bayes để phân lớp và kiểm tra xem nó có phân lớp đúng hay không?

- 6.6. Dùng bộ phân lớp Naive Bayes trong phần mềm weka để phân lớp tập dữ liệu đi kèm và đánh giá sử dụng phương pháp thẩm định chéo (10-folds cross-validation).
- 6.7. Giả sử trong bảng dữ liệu 6.1, ta lấy dòng đầu tiên làm dữ liệu kiểm thử, toàn bộ các dòng còn lại làm dữ liệu huấn luyện. Dùng thuật toán kNN với $k=1$ để phân lớp và kiểm tra xem nó có phân lớp đúng hay không?
- 6.8. Dùng bộ phân lớp KStar trong phần mềm weka để phân lớp tập dữ liệu đi kèm và đánh giá sử dụng phương pháp thẩm định chéo (10-folds cross-validation).
- 6.9. Dùng bộ phân lớp LibSVM trong phần mềm weka để phân lớp tập dữ liệu đi kèm và đánh giá sử dụng phương pháp thẩm định chéo (10-folds cross-validation).
- 6.10. Cài đặt giải thuật cây quyết định sử dụng độ lợi thông tin, sau đó áp dụng phân lớp dữ liệu trong bảng 6.1 với dòng đầu tiên làm dữ liệu kiểm thử và các dòng còn lại làm dữ liệu huấn luyện.
- 6.11. Cài đặt giải thuật cây quyết định sử dụng tỉ số độ lợi, sau đó áp dụng phân lớp dữ liệu trong bảng 6.1 với dòng đầu tiên làm dữ liệu kiểm thử và các dòng còn lại làm dữ liệu huấn luyện.
- 6.12. Cài đặt giải thuật cây quyết định sử dụng tỉ số Gini, sau đó áp dụng phân lớp dữ liệu trong bảng 6.1 với dòng đầu tiên làm dữ liệu kiểm thử và các dòng còn lại làm dữ liệu huấn luyện.
- 6.13. Cài đặt giải thuật Naive Bayes cho dữ liệu rời rạc, sau đó áp dụng phân lớp dữ liệu trong bảng 6.1 với dòng đầu tiên làm dữ liệu kiểm thử và các dòng còn lại làm dữ liệu huấn luyện.
- 6.14. Cài đặt giải thuật Naive Bayes cho dữ liệu liên tục, sau đó áp dụng phân lớp dữ liệu đi kèm với phần mềm weka. Chia file dữ liệu ra thành 2 nửa theo tỉ lệ 70%/30% làm dữ liệu huấn luyện và dữ liệu kiểm thử.
- 6.15. Cài đặt giải thuật kNN (với $k = 1$), sau đó áp dụng phân lớp dữ liệu đi kèm với phần mềm weka. Chia file dữ liệu ra thành 2 nửa theo tỉ lệ 70%/30% làm dữ liệu huấn luyện và dữ liệu kiểm thử.



Chương 7.

PHƯƠNG PHÁP HỌC BÁN GIÁM SÁT

7.1. GIỚI THIỆU

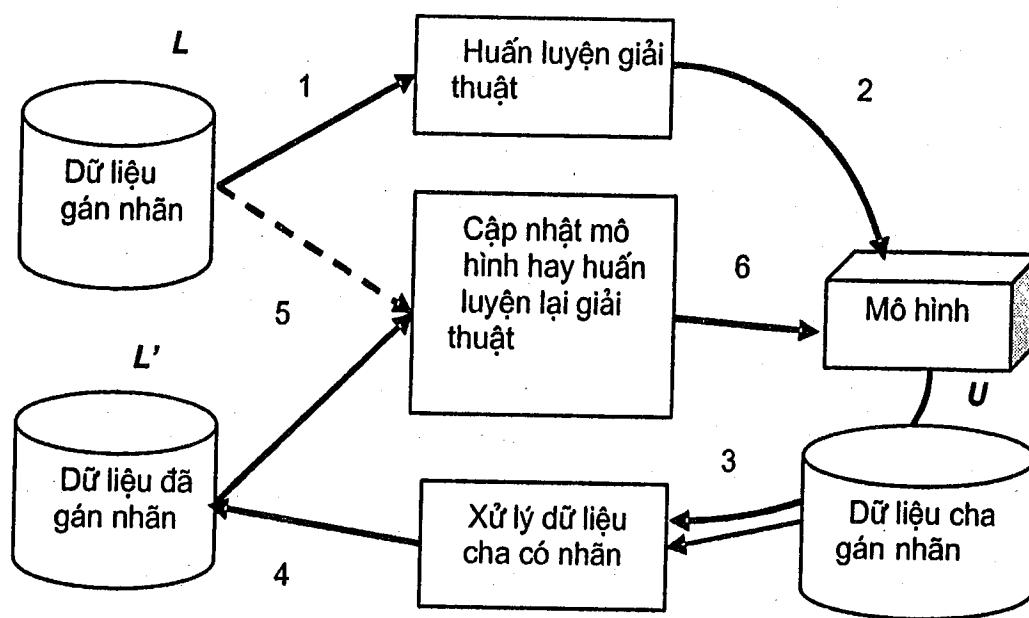
Các thuật toán đã trình bày ở chương 6 có đặc điểm là chỉ có thể học từ dữ liệu đã gán nhãn, việc tạo ra các dữ liệu gán nhãn thường là công việc buồn tẻ nhưng lại tốn công sức. Trong nghiên cứu của mình Lang [Lang95] đã chứng minh rằng: khi một người đọc 1000 bài báo để gán nhãn lớp cho chúng, thì một giải thuật phân lớp dựa trên tập dữ liệu gán nhãn này có thể đạt được độ chính xác là 50%. Trong nhiều hệ thống trong thực tế, ít người có đủ kiên nhẫn để thực hiện công việc gán nhãn dữ liệu như trên và đặc biệt là thu được một chất lượng phân lớp thấp như vậy. Chúng ta rất muốn có một giải thuật có thể chỉ cần vài chục dữ liệu gán nhãn (thay vì vài nghìn dữ liệu gán nhãn) mà vẫn có thể cho chúng ta một bộ phân lớp với độ chính xác chấp nhận được. Ngoài dữ liệu gán nhãn thì trong thực tế các dữ liệu chưa gán nhãn thường tồn tại với số lượng lớn, chẳng hạn với bài toán phân lớp văn bản, thì nguồn dữ liệu các trang web từ Internet là rất lớn. Nếu tận dụng được các nguồn dữ liệu chưa gán nhãn thì sẽ làm giảm được công sức tạo dữ liệu cũng như làm tăng được chất lượng của các bộ phân lớp. Hiện tại đã có rất nhiều nghiên cứu và đề xuất các giải thuật có khả năng sử dụng dữ liệu gán nhãn, đồng thời tận dụng cả dữ liệu chưa gán nhãn để làm giàu (augment) thêm dữ liệu huấn luyện nhằm làm tăng chất lượng phân lớp. Các giải thuật có đặc điểm này được phân vào lớp giải thuật học bán giám sát [Goldman00, Dempster77, Nigam00, Zhu05, Zhu07].

Để có thể phân biệt được các loại (lớp) giải thuật học có giám sát, không giám sát và bán giám sát ta có thể xem xét các đặc điểm sau của các loại giải thuật:

- Giải thuật học có giám sát: Đầu vào của nó là một tập dữ liệu đã được gán nhãn $\{x_i, y_i\}$, trong đó y_i là các nhãn tương ứng của phần tử dữ liệu x_i , hay nói cách khác mỗi một phần tử dữ liệu x_i đã được phân vào các lớp cụ thể y_i . Nhiệm vụ của các giải thuật này là tìm mối quan hệ giữa dữ liệu và nhãn để có thể dùng để dự đoán nhãn của một phần tử dữ liệu x mới chưa có nhãn.
- Giải thuật học không giám sát: Đầu vào của nó là một tập chỉ chứa các phần tử dữ liệu $\{x_i\}$ không có nhãn, hay nói cách khác chúng ta không biết trước nhãn các phần tử dữ liệu x_i . Nhiệm vụ của các giải thuật này là tìm ra cấu trúc quan trọng của dữ liệu, và phân dữ liệu thành các nhóm có các đặc điểm chung.
- Các giải thuật học bán giám sát: Về yêu cầu đầu ra nó cũng giống giải thuật học có giám sát tức là nó cũng phải tìm ra quan hệ giữa dữ liệu và nhãn để có thể dự đoán được các phần tử dữ liệu mới. Tuy nhiên sự khác biệt ở đây là đầu vào cho các giải thuật bán giám sát là một tập nhỏ các dữ liệu có gán nhãn $\{x_i, y_i\}$ và một tập lớn các dữ liệu không gán nhãn $\{x_j\}$, các giải thuật phải học ra quan hệ giữa dữ liệu và nhãn từ 2 tập dữ liệu này. Do đó ta có thể định nghĩa một cách không hình thức (1) *giải thuật học bán giám sát = giải thuật học có giám sát + dữ liệu không gán nhãn*, hoặc (2) *giải thuật học bán giám sát = dữ liệu có gán nhãn + giải thuật học không giám sát*. Tùy vào loại giải thuật học bán giám sát khác nhau mà nó thuộc định nghĩa không hình thức (1) hay (2).

Giải thuật học bán giám sát thuộc định nghĩa không hình thức (1) có mô hình chung như minh họa ở hình 7.1. Đầu vào cho giải thuật bán giám sát là một tập nhỏ dữ liệu gán nhãn L và một

tập dữ liệu chưa gán nhãn U . Tập dữ liệu gán nhãn L (1) sẽ được sử dụng để huấn luyện các giải thuật để tạo ra mô hình ban đầu (2). Mô hình này sẽ được dùng để gán nhãn các dữ liệu chưa được gán nhãn U (3) và ta thu được tập dữ liệu đã gán nhãn L' (4). Tùy theo từng giải thuật mà toàn bộ tập dữ liệu L' hay một tập con của L' , kết hợp với tập L (5) được dùng để huấn luyện hay cập nhật lại mô hình của thuật toán để tạo ra mô hình mới (6). Quá trình 3, 4, 5 và 6 sẽ được lặp đi lặp lại để làm tăng chất lượng phân lớp của giải thuật. Tùy theo từng loại giải thuật học bán giám sát mà đầu ra của nó sẽ là mô hình phân lớp hay là một tập dữ liệu huấn luyện L đã được bổ sung thêm các phần tử dữ liệu gán nhãn có độ tin cậy. Phần tiếp theo của chương 7 sẽ trình bày một số thuật toán phân lớp bán giám sát thông dụng.



Hình 7.1. Mô hình chung của các giải thuật bán giám sát dựa trên giải thuật giám sát

Các thuật toán học bán giám sát thường giả định (assumption) rằng tập dữ liệu có nhãn L và tập dữ liệu chưa gán nhãn U là có cùng phân bố. Với giả định này thì ta mới có thể khai thác được các phần tử dữ liệu chưa có nhãn để làm giàu tập dữ liệu có nhãn, hay nói một cách khác với giả định này thì ta mới có thể sử dụng mô hình thu được từ tập dữ liệu gán nhãn L để phân lớp các phần tử dữ liệu trong tập dữ liệu chưa có nhãn U .

Ngoài cách phân loại giải thuật học bán giám sát theo định nghĩa không hình thức (1) và (2) ở trên, ta còn có một số cách phân loại giải thuật học bán giám sát khác:

- Giải thuật học bán giám sát dựa trên bộ phân lớp: bắt đầu bằng giải thuật phân lớp yếu (weak), là giải thuật phân lớp có hiệu năng thấp, ta sẽ dần dần cải thiện chất lượng của giải thuật phân lớp để cuối cùng thu được giải thuật phân lớp có hiệu năng cao. Các giải thuật thuộc lớp này có thể kể tên là thuật toán cực đại kỳ vọng, giải thuật học cộng tác, ... sẽ được giới thiệu ở dưới.
- Giải thuật học bán giám sát dựa trên dữ liệu: phát hiện các cấu trúc hình học (geometry) của dữ liệu, từ đó xây dựng các bộ phân lớp dựa trên các cấu trúc này. Một giải thuật thuộc phân lớp này là giải thuật manifold.
- Ngoài ra ta có thể phân lớp giải thuật học bán giám sát dựa vào đặc điểm của dữ liệu đầu vào: khi dữ liệu đầu vào là một cấu trúc phức tạp (chẳng hạn như một cây hay một đồ thị) thì giải thuật phân lớp được gọi là phân lớp cấu trúc (structured learning) và giải thuật học giám sát sẽ có tên là học bán giám sát cho phân lớp cấu trúc (semi-supervised learning for structure prediction).

7.2. THUẬT TOÁN CỰC ĐẠI KỲ VỌNG EM

Giải thuật *cực đại kỳ vọng* (Expectation Maximization - EM) được đề cập ở chương 5 đã được sử dụng để thực hiện bài toán phân cụm (thuộc lớp bài toán học không giám sát) [Dempster77]. Tuy nhiên giải thuật này cũng có thể sử dụng để làm giải thuật phân lớp dưới dạng giải thuật học bán giám sát, hay nói cách khác, thuật toán học bán giám sát cực đại kỳ vọng thuộc định nghĩa không hình thức (2): giải thuật học không giám sát + dữ liệu huấn luyện. Nhắc lại một số điểm liên quan đến mô hình sinh (generative model) được sử dụng trong giải thuật EM:

- Mỗi lớp trong tập dữ liệu đang xét có thể được biểu diễn bằng một phân bố xác suất, nếu ta có k lớp thì sẽ có k

phân bố xác suất được gọi là phân bố (xác suất) thành phần (component distribution)

- Toàn bộ tập dữ liệu (hay miền dữ liệu đang xét) là sự trộn (hay một phân bố trộn) hữu hạn (finite mixture) của các phân bố này, từ hữu hạn ở đây thể hiện số lượng các thành phần (component) là hữu hạn. Hình 6.2 minh họa mô hình trộn gồm 2 phân bố xác suất của 2 lớp. Do đó ta có thể phân lớp toàn bộ tập dữ liệu đầu vào bằng cách sử dụng mô hình mật độ trộn (mixture density model) của k phân bố xác suất, trong đó một phân bố biểu diễn một lớp. Như vậy, nhiệm vụ của giải thuật phân lớp là đi tìm (ước lượng) các tham số của các phân bố xác suất sao cho phù hợp với tập dữ liệu đầu vào nhất bằng cách khai thác tập dữ liệu không có nhãn U .

Nhắc lại rằng ở chương 5 chúng ta đã được biết là giải thuật EM có thể dùng kết hợp với giải thuật phân lớp Bayes cho bài toán phân cụm. Trong chương này chúng ta sẽ tìm hiểu giải thuật EM kết hợp với giải thuật phân lớp Bayes trong bài toán học bán giám sát để phân lớp. Do chúng ta có 2 bài toán khác nhau, nên giải thuật EM trong chương này sẽ có sửa đổi so với bài toán phân cụm.

Với tập dữ liệu gán nhãn đầu vào L gồm $|L|$ phần tử dữ liệu, và được chia thành k lớp. Để đơn giản, ta xét trường hợp mỗi phần tử dữ liệu được biểu diễn bằng 1 số thực. Gọi m_C , σ_C và $P(C)$ tương ứng là giá trị trung bình, độ lệch chuẩn và xác suất lấy mẫu của lớp C , các giá trị trên được tính như sau:

$$m_C = \frac{1}{|C|} \sum_{p \in C} p, \quad \sigma_C = \sqrt{\frac{1}{|C|} \sum_{x \in C} (x - m_C)^2} \quad \text{và} \quad P(C) = \frac{|C|}{|L|} \quad (7.1)$$

trong đó, $|C|$ là lực lượng của tập các phần tử dữ liệu thuộc lớp C . Khi đó bộ ba $\langle m_C, \sigma_C, P(C) \rangle$ được gọi là mô hình sinh của lớp C , theo phân bố chuẩn Gauss (Gaussian distribution).

Sau khi đã có bộ ba $\langle m_C, \sigma_C, P(C) \rangle$ cho từng lớp, ta có thể xác định xác suất mà một phần tử dữ liệu thuộc vào lớp C là bao nhiêu. Trường hợp nếu phần tử dữ liệu được biểu diễn bằng giá trị

rồi rạc (chỉ gồm các giá trị 0 và 1) thì xác suất của một phần tử dữ liệu x thuộc vào lớp C được tính bằng công thức:

$$P(C|x) = \frac{P(C)P(x|C)}{P(x)} \quad (7.2)$$

và $P(x|C)$ được tính bằng số lần xuất hiện của x trong lớp C chia cho tổng số phần tử dữ liệu trong lớp C :

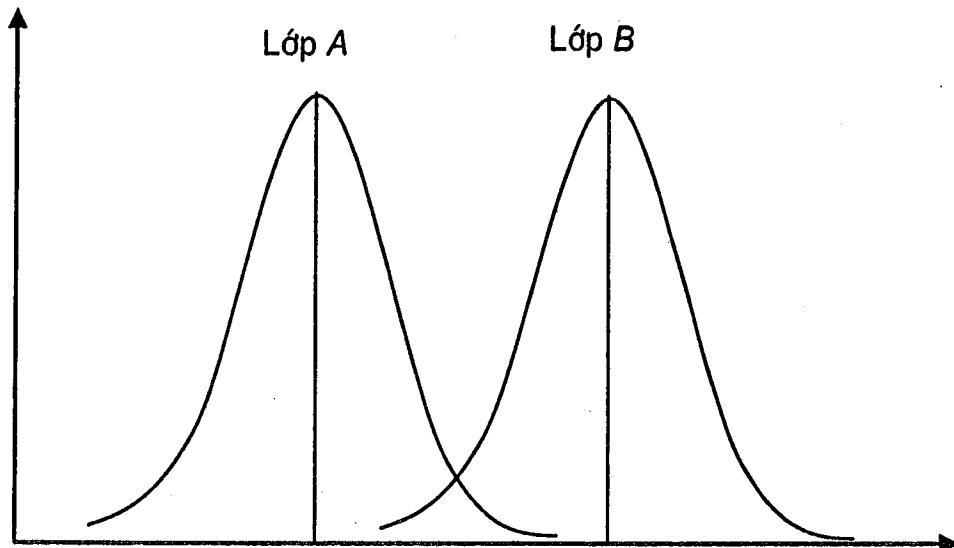
$$P(x|C) = \frac{|\{x_i \in C | x_i = x\}|}{|C|}. \text{ Trường hợp dữ liệu được biểu diễn}$$

bằng số thực, khi đó xác suất $P(x|C)$ được tính bằng công thức

$$P(C|x) \approx \frac{f_C(x)P(C)}{P(x)} \quad (7.3)$$

$$\text{trong đó } f_C(x) = \frac{1}{\sqrt{2\pi}\sigma_C} e^{-(x-m_C)/(2\sigma_C^2)} \quad (7.4)$$

Do $P(x)$ xuất hiện trong tất cả các công thức tính xác suất của x thuộc vào các lớp khác nhau nên ta có thể bỏ qua không cần tính. Nhưng khi đó các giá trị tính toán được $P(C|x)$ có thể không thỏa mãn điều kiện $\sum_C P(C|x) = 1$, do đó ta có thể cần phải chuẩn hóa lại để đảm bảo điều kiện này.



Hình 6.2. Mô hình trộn hữu hạn (trường hợp có 2 lớp)

Khi sử dụng công thức trên thì ta cần phải chú ý là nếu có trường hợp có một đặc trưng nào đó có giá trị độ lệch chuẩn σ là 0

thì ta không thể tính được giá trị $f_C(x)$. Để xử lý trường hợp này ta có thể lấy một xác suất ngầm định nào đó, chẳng hạn là 0.05 để thay vào các xác suất $P(x | C)$.

Trong trường hợp tổng quát thì một phần tử dữ liệu có thể được biểu diễn bằng nhiều đặc trưng (nhiều chiều). Giả sử mỗi một phần tử dữ liệu x được biểu diễn bằng một vector d chiều (x_1, x_2, \dots, x_d), khi đó với giả thiết là các đặc trưng là độc lập nhau thì ta có thể sử dụng công thức sau:

$$P(x | C) = P((x_1, x_2, \dots, x_d) | C) = \prod_{i=1}^d P(x_i | C) = \prod_{i=1}^d f_C^i(x_i) \quad (7.5)$$

Dựa trên các lý thuyết ở trên, giải thuật học bán giám sát cực đại kỳ vọng hoạt động như sau:

Đầu vào: Tập dữ liệu gán nhãn L và tập dữ liệu không gán nhãn $U = \{x_1, x_2, \dots, x_n\}$

Đầu ra: Mô hình phân lớp M

Giải thuật cực đại kỳ vọng

Bước 1: Dùng tập dữ liệu có gán nhãn L để xây dựng mô hình M gồm các tham số $\langle m_C, \sigma_C, P(C) \rangle$ ban đầu cho k lớp. Các tham số này sẽ được cập nhật thông qua việc khai thác các phần tử dữ liệu không có nhãn trong tập U bằng vòng lặp 2 bước sau:

Bước kỳ vọng: Với từng phần tử $x_i \in U$ ($1 \neq i \neq n$) tính giá trị $w_i^C = P(C | x_i)$ là xác suất x_i thuộc vào lớp C . Chuẩn hóa giá trị w_i^C trên toàn bộ k lớp để đảm bảo $\sum_{C=1}^k w_i^C = 1$. Giá trị w_i^C thu được tại thời điểm này chính là giá trị kỳ vọng phần tử x_i thuộc vào lớp C (đây là nguyên nhân bước này có tên là bước kỳ vọng).

Bước cực đại: Trong bước này ta cần tìm giá trị lớn nhất (cực đại hóa) xác suất mô hình M trên tập dữ liệu D ($D = L \cup U$): $P(M | D)$. Theo công thức Bayes $P(M | D) = \frac{P(D | M)P(M)}{P(D)}$, do $P(D)$

không phụ thuộc vào mô hình M , nên thay vì cực đại hóa $P(M | D)$ trực tiếp, ta có thể cực đại hóa biểu thức $P(D | M)P(M)$, với $P(D | M)$ được tính theo công thức:

$$P(D | M) = \prod_{x_i \in U} \sum_{j=1}^k P(c_j) P(x_i | c_j) \times \\ \prod_{x_i \in L} \sum_{j=1}^k P(y_i = c_j) P(x_i | y_i = c_j) \quad (7.6)$$

trong đó, $P(D | M)$ là xác suất của tập dữ liệu D trên mô hình M ; Chú ý trong công thức trên ta có tính xác suất của các phần tử dữ liệu trong tập dữ liệu có nhãn L và tập chưa có nhãn U . Trong tập dữ liệu huấn luyện L , ký hiệu y_i là nhãn của các phần tử x_i .

Vì công thức (7.6) có chứa tích các xác suất nên tích của chúng sẽ rất nhỏ, do đó ta có thể dùng hàm $\log()$ của nó để thay thế, ký hiệu:

$$L(P(D | M)P(M)) \equiv \log(P(M | D) \times P(M)) \\ = \log(P(M)) + \log(P(M | D)) \\ = \log(P(M)) + \sum_{x_i \in U} \log\left(\sum_{j=1}^k P(c_j) P(x_i | c_j)\right) \quad (7.7) \\ + \sum_{x_i \in L} \log\left(\sum_{j=1}^k P(y_i = c_j) P(x_i | y_i = c_j)\right)$$

Ta có thể tìm được giá trị cực đại địa phương (local maximum) của $L(P(D | M)P(M))$ bằng giải thuật leo đồi (hill climbing) khi lặp lại 2 bước kỳ vọng và cực đại.

Trong một số trường hợp cụ thể (hay miền dữ liệu cụ thể), để làm tăng vai trò (trọng số) của dữ liệu có gán nhãn và dữ liệu không có nhãn, công thức (7.7) có thể được viết lại như sau:

$$L(P(D | M)P(M)) = \\ \log(P(M)) + \lambda \sum_{x_i \in U} \log\left(\sum_{j=1}^k P(c_j) P(x_i | c_j)\right) \quad (7.8) \\ + (1 - \lambda) \sum_{x_i \in L} \log\left(\sum_{j=1}^k P(y_i = c_j) P(x_i | y_i = c_j)\right)$$

Bằng cách thay đổi giá trị của λ , ta sẽ làm thay đổi trọng số của phần dữ liệu gán nhãn và không gán nhãn nhằm thu được mô hình tối ưu nhất.

7.3. THUẬT TOÁN HỌC CỘNG TÁC (CO-TRAINING)

Giải thuật học cộng tác (co-training) là giải thuật thuộc lớp giải thuật bán giám sát được đề xuất bởi Blum và Mitchell [Blum98] năm 1998. Dựa trên giải thuật này đã có rất nhiều giải thuật học cộng tác khác được đề xuất. Mục này sẽ giới thiệu giải thuật học cộng tác gốc của Blum và Mitchell và một số giải thuật học cộng tác khác.

7.3.1. Thuật toán học cộng tác dựa trên nhiều khung nhìn

Đây là giải thuật học cộng tác đầu tiên được Blum và Mitchell đề xuất [Blum98]. Giải thuật học cộng tác này dựa trên giả thiết rằng: trong miền dữ liệu đang xét, các phần tử dữ liệu có thể được biểu diễn bằng hai hay nhiều tập các đặc trưng (features); tập các đặc trưng này là độc lập điều kiện với nhau (conditionally independent). Mỗi tập đặc trưng này là đủ (sufficient) để phân lớp tập dữ liệu trong miền dữ liệu đang xét với lượng dữ liệu huấn luyện đủ lớn. Ví dụ trong miền dữ liệu web, một trang web có thể được biểu diễn bằng một vector, trong đó mỗi phần tử trong vector biểu diễn một từ trong từ điển tương ứng với miền dữ liệu các trang web đang xét. Trong các trang web thì có một đối tượng rất quan trọng là các liên kết (hyperlink), một liên kết trong trang web hiện tại $q.html$ đến trang web $p.html$ khi biểu diễn bằng cú pháp HTML có dạng:

< a href = "p.html" > cụm từ mô tả sơ lược về trang web $p.html$ </ a >

Ta cũng có thể sử dụng các từ xuất hiện trong các liên kết trỏ đến trang web p để biểu diễn trang web p . Chú ý rằng, các từ xuất hiện trong liên kết trên đến trang web $p.html$ sẽ độc lập (ở một mức độ nào đó) với các từ xuất hiện trong bản thân trang web $p.html$ (vì các từ này xuất hiện ở trang $q.html$). Do đó hai tập đặc trưng biểu diễn các trang web sẽ độc lập nhau. Mỗi tập đặc trưng biểu diễn trang web trên được gọi là một *khung nhìn* (view).

Định nghĩa một cách hình thức thì mô hình học cộng tác sẽ như sau:

- Gọi X là không gian biểu diễn một miền dữ liệu đang xét $X = X^1 \times X^2$, trong đó X^1 và X^2 tương ứng là một khung nhìn của một phần tử dữ liệu, và giả sử mỗi khung nhìn là đủ để phân lớp dữ liệu với một lượng dữ liệu huấn luyện đủ lớn. Hay nói một cách khác, một phần tử dữ liệu x sẽ được biểu diễn bằng một cặp $x = (x_1, x_2)$, trong đó x_1 và x_2 là các vector biểu diễn dữ liệu tương ứng với khung nhìn X^1 và X^2 .
- Gọi D là một phân bố trên X , C_1 và C_2 là tập các lớp được định nghĩa tương ứng trên X^1 và X^2 . Gọi f_1 và f_2 là hai hàm phân lớp tương ứng trên 2 khung nhìn X^1 và X^2 : $f_1 : X^1 \rightarrow C_1$ và $f_2 : X^2 \rightarrow C_2$.
- Giả sử tất cả các phần tử dữ liệu có nhãn và có xác suất khác 0 trên D là nhất quán (consistent) với f_1 và f_2 . Nói một cách khác nếu gọi f là hàm phân trên X : $f : X \rightarrow C$ (với C là tập các lớp trên X), thì với mọi phần tử dữ liệu $x = (x_1, x_2)$ có nhãn quan sát được là l , thì $f(x) = f_1(x_1) = f_2(x_2) = l$; và D sẽ gán xác suất bằng 0 cho tất cả các phần tử $x = (x_1, x_2)$ nếu $f_1(x_1) \neq f_2(x_2)$. Với giả sử này thì với một phần tử dữ liệu không có nhãn và nó vẫn có đặc điểm là $f(x) = f_1(x_1) = f_2(x_2)$, do đó ta có thể tận dụng các phần tử dữ liệu không có nhãn làm dữ liệu có gán nhãn (để làm dữ liệu huấn luyện) bằng cách gán nhãn cho nó.

Giải thuật học cộng tác được mô tả như sau:

Đầu vào: + Tập dữ liệu gán nhãn L

+ Tập dữ liệu không gán nhãn U , mỗi phần tử dữ liệu x thuộc tập L và U đều có dạng $x = (x_1, x_2)$

+ Số vòng lặp k

Đầu ra: là tập dữ liệu gán nhãn L đã được làm giàu và hai bộ phân lớp có chất lượng đã được cải thiện

Thuật toán co-training

1. Tạo một tập U' gồm u phần tử dữ liệu lấy ngẫu nhiên từ U
2. Lặp k lần các bước sau
 - 2.1 Dùng tập L để huấn luyện bộ phân lớp h_1 bằng cách sử dụng phần x_1 của x
 - 2.2 Dùng tập L để huấn luyện bộ phân lớp h_2 bằng cách sử dụng phần x_2 của x
 - 2.3 Dùng h_1 và h_2 để phân lớp các phần tử dữ liệu nằm trong tập U'
 - 2.4 Giả sử h_1 và h_2 đều nhất trí phân lớp, cụ thể là $h_1(x) = h_2(x)$ với độ tin cậy cao (hay xác suất phân lớp là cao) cho p phần tử dữ liệu thuộc lớp dương và n phần tử thuộc lớp âm
 - 2.5 Thêm $p + n$ phần tử dữ liệu đã gán nhãn ở trên vào tập dữ liệu L
 - 2.6 Chọn ngẫu nhiên $2(p + n)$ phần tử không có nhãn trong tập U để tạo lại tập U'

Trong đó h_i là một giải thuật phân lớp có giám sát nào đó, chẳng hạn như Naive Bayes hay SVM. Để làm tăng chất lượng của các phần tử dữ liệu trong tập U được đưa vào tập L , thì ta chỉ chọn các phần tử đã được 2 bộ phân lớp đồng thuận và có xác suất phân lớp là cao.

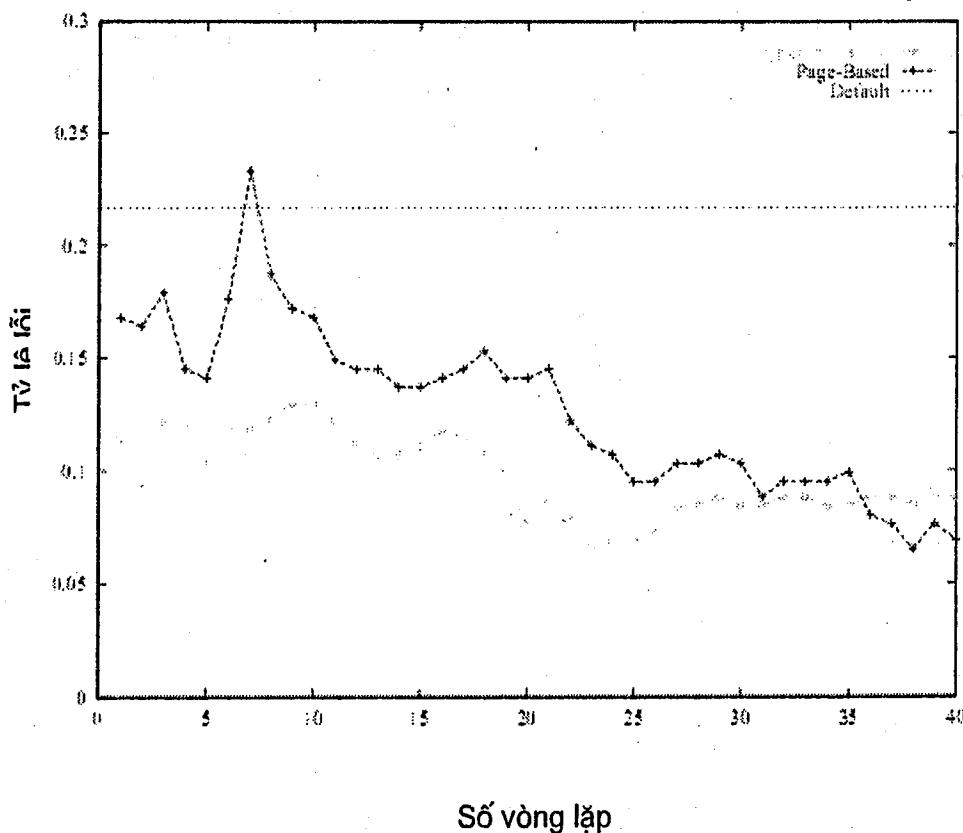
Đầu ra của giải thuật học cộng tác là tập dữ liệu đã được làm giàu và 2 bộ phân lớp có chất lượng đã được cải tiến so với tập dữ liệu huấn luyện L ban đầu. Ta có thể dùng tập dữ liệu đã được làm giàu này để huấn luyện các bộ phân lớp khác. Một phương pháp khác tận dụng đầu ra của giải thuật học cộng tác này là kết hợp hai bộ phân lớp này thành một bộ phân lớp mới $h(x) = h_1(x_1)h_2(x_2)$.

Trong thực nghiệm, Blum và Michell đã áp dụng thuật toán học cộng tác để phân lớp web, giá trị cho các tham số dùng trong thực nghiệm tương ứng là: $u = 75$; h_1 và h_2 là bộ phân lớp Bayes; $p = 1$; và $n = 3$.

Khi làm việc với các thuật toán học cộng tác ở trên cần lưu ý một số vấn đề sau đây. Thứ nhất, kích thước tập dữ liệu gán nhãn có ảnh hưởng lớn đến hiệu quả của thuật toán học cộng tác. Nếu tập này quá nhỏ thì sẽ làm cho các thuật toán học cộng tác có hiệu quả không cao, lý do là khi lượng dữ liệu huấn luyện quá ít thì tỉ lệ lỗi của các thuật toán học có giám sát sẽ cao, khả năng phân lớp nhầm các phần tử dữ liệu chưa gán nhãn sẽ cao, hệ quả là các phần tử khi đưa vào tập dữ liệu gán nhãn sẽ làm sai lệch tập dữ liệu huấn luyện ban đầu. Trong trường hợp quá nhiều thì thực sự chúng ta không thu được lợi ích từ co-training, lý do là tập dữ liệu huấn luyện đã chứa đủ dữ liệu để đại diện cho phân bố của dữ liệu, nên chúng ta không thể thu lợi gì được từ tập dữ liệu không gán nhãn. Kích thước của tập dữ liệu gán nhãn L là bao nhiêu là phù hợp thì hoàn toàn phụ thuộc vào từng bài toán cụ thể. Thứ hai, cơ sở tăng hiệu quả co-training là vấn đề thiết lập các tham số trong thuật toán như kích cỡ tập dữ liệu gán nhãn, kích cỡ tập dữ liệu chưa gán nhãn, số mẫu được thêm vào sau mỗi vòng lặp. Trong mọi trường hợp, việc chọn bộ phân lớp thành phần cho từng khung nhìn là rất quan trọng [Zhu05, Zhu07] vì nó ảnh hưởng đến chất lượng của giải thuật học cộng tác. Vấn đề cuối cùng là việc chọn các khung nhìn X^1 và X^2 cũng là một bài toán quan trọng vì nó ảnh hưởng đến chất lượng của giải thuật học cộng tác.

Một đặc điểm của giải thuật học cộng tác ở trên là trong vòng lặp nó không đảm bảo là vòng lặp sau sẽ làm cho chất lượng của các bộ phân lớp có chất lượng cao hơn vòng lặp trước đó. Lý do là trong các vòng lặp ta có bước thêm các phần tử dữ liệu lấy từ tập không có nhãn U' vào trong tập dữ liệu L , trong các phần tử dữ liệu này có thể xuất hiện các phần tử có nhãn bị lỗi, nếu điều này xảy ra thì tỉ lệ lỗi sẽ tăng lên. Hình 6.3 là sơ đồ tỷ lệ lỗi của hai bộ phân lớp Bayes trong thực nghiệm của tác giả Blum và Mitchell khi áp dụng thuật toán học cộng tác cho tập dữ liệu web. Trong đó hyperlink-page là giải thuật Bayes sử dụng tập đặc trưng là các từ xuất hiện trong liên kết và page-based là giải thuật Bayes sử dụng tập đặc trưng là các từ xuất hiện trong bản thân các trang web.

Một hệ quả thu được từ đặc điểm trên là kết quả của giải thuật học cộng tác trong các lần chạy khác nhau là khác nhau. Do đó trong thực tế để thu được kết quả tốt ta thường chạy giải thuật trên tập dữ liệu đầu vào một số lần, kết quả cuối cùng được chọn là cái nào cho chất lượng tốt nhất. Đây cũng là đặc điểm chung của hầu hết các giải thuật học bán giám sát.



Hình 6.3. Sơ đồ tỷ lệ lỗi của giải thuật giám sát trong các vòng lặp

7.3.2. Thuật toán học cộng tác co-EM

Nigram và Ghani đã tiến hành thực nghiệm để so sánh giải thuật học cộng tác và giải thuật EM [Nigram00]. Vì EM theo mô hình sinh, do đó nó hoạt động tốt khi dữ liệu không gán nhãn (unlabeled data) phù hợp với mô hình sinh. Và ngược lại, nếu mỗi cụm của dữ liệu trong tập dữ liệu không gán nhãn không tương ứng với các lớp, trong trường hợp này thì giải thuật học cộng tác có hiệu năng tốt hơn với EM. Do đó, tác giả đề xuất kết hợp 2 giải thuật học

bán giám sát: co-training và EM lại với nhau thành một giải thuật mới với kỳ vọng là nó kết hợp được ưu điểm của 2 phương pháp học. Giải thuật mới có tên là co-EM được mô tả như sau:

Đầu vào: + tập dữ liệu gán nhãn L

+ tập dữ liệu không gán nhãn U , mỗi phần tử dữ liệu x thuộc tập L và U đều có dạng $x = (x_1, x_2)$

Đầu ra: 2 bộ phân lớp có chất lượng đã được cải thiện

Thuật toán co-EM

1. Khởi tạo: dùng tập dữ liệu có nhãn L để huấn luyện bộ phân lớp $h_1(x)$ trên tập đặc trưng X^1
2. Lặp cho đến khi $h_1(x)$ và $h_2(x)$ hội tụ
 - 2.1 Dùng $h_1(x)$ phân lớp tập dữ liệu không có nhãn U để tạo ra tập dữ liệu có nhãn L'
 - 2.2 Huấn luyện bộ phân lớp $h_2(x)$ bằng tập dữ liệu $L \cup L'$ trên tập đặc trưng X^2
 - 2.3 Sử dụng $h_2(x)$ để phân lớp lại tập dữ liệu U để tạo ra tập dữ liệu có nhãn L'
 - 2.4 Huấn luyện bộ phân lớp $h_1(x)$ bằng tập dữ liệu $L \cup L'$ trên tập đặc trưng X^1

Chú ý là trong giải thuật co-EM thì ta không tạo một tập con $U' \subset U$ để gán nhãn (như trong giải thuật học cộng tác), ở đây toàn bộ tập dữ liệu không có nhãn U được sử dụng. Cũng giống giải thuật học cộng tác, chất lượng của giải thuật co-EM bị phụ thuộc nhiều vào việc chọn các khung nhìn (view) cho bộ phân lớp $h_i(x)$.

7.3.3. Thuật toán học cộng tác dựa trên nhiều giải thuật học giám sát

Việc chia tập đặc trưng như trên thành 2 tập con không giao nhau và độc lập có điều kiện với nhau ở trên có hạn chế là trong một số trường hợp khi số lượng đặc trưng rất ít thì việc tạo ra hai tập con có thể không khả thi. Hay nếu ta chia một tập đặc trưng thành hai tập con X^1, X^2 , thì các tập con này thường không biểu

diễn đầy đủ các thông tin cho các phần tử dữ liệu, do đó, chất lượng của bộ phân lớp trên tập con đặc trưng sẽ bị giảm (so với phân lớp khi sử dụng toàn bộ tập đặc trưng). Một trong các đề xuất giải pháp để khắc phục trường hợp trên là sử dụng các giải thuật phân lớp khác nhau thay vì một giải thuật như trên. Việc chọn các giải thuật phân lớp được xây dựng trên các mô hình học máy khác nhau sao cho việc phân lớp của các giải thuật phân lớp là độc lập nhau cho từng phần tử dữ liệu. Khi đảm bảo được điều này thì chúng ta có thể dùng được toàn bộ tập đặc trưng để huấn luyện cho các bộ phân lớp. Goldman và Zhou là tác giả của giải thuật co-training theo tư tưởng này, chi tiết của giải thuật có thể xem tại [Goldman00].

Nhóm tác giả Vincent Ng và Claire Cardie cũng đề xuất giải thuật học bán giám sát Multi-Classifier theo ý tưởng này [Vincent03]. Giải thuật của nhóm tác giả này được minh họa như sau:

Đầu vào: + tập dữ liệu gán nhãn L

+ tập dữ liệu không gán nhãn U

+ hai giải thuật phân lớp khác nhau $Learn_1(x)$ và $Learn_2(x)$

Đầu ra: 2 bộ phân lớp có chất lượng đã được cải thiện

Thuật toán Multi-Classifier

1. Khởi tạo:

1.1 Khởi tạo tập dữ liệu huấn luyện cho các bộ phân lớp
 $L_1 = L, L_2 = L$

1.2 Lấy ngẫu nhiên u phần tử dữ liệu từ tập dữ liệu không có nhãn U để tạo thành tập U'

2. Lặp cho đến khi U' rỗng

2.1 Dùng L_1 để huấn luyện bộ phân lớp thứ nhất
 $h_1 = Learn_1(L_1)$

2.2 Dùng L_2 để huấn luyện bộ phân lớp thứ hai
 $h_2 = Learn_2(L_2)$

- 2.3 Dùng bộ phân lớp thứ nhất gán nhãn cho tập dữ liệu U' , sau đó chọn k phần tử được phân lớp có độ tin cậy cao nhất để đưa vào tập huấn luyện L_2

$$L_2 = L_2 \cup \{(x, h_1(x)) \mid x \in U', h_1(x) \in k_top_confidence(h_1(x))\}$$
- 2.4 Dùng bộ phân lớp thứ hai gán nhãn cho tập dữ liệu U' , sau đó chọn k phần tử được phân lớp có độ tin cậy cao nhất để đưa vào tập huấn luyện L_1

$$L_1 = L_1 \cup \{(x, h_2(x)) \mid x \in U', h_2(x) \in k_top_confidence(h_2(x))\}$$
- 2.5 Tạo lại tập U' bằng cách chọn ngẫu nhiên các phần tử trong U

3. Output $h(x) \leftarrow h_1(x)h_2(x)$

Điểm khác nữa giữa giải thuật Multi-Classifier so với giải thuật học cộng tác là kết quả phân lớp tập dữ liệu của bộ phân lớp này sẽ được thêm vào tập huấn luyện của bộ phân lớp kia ($L_2 = L_2 \cup \{(x, h_1(x)) \mid x \in U', h_1(x) \in k_top_confidence(h_1(x))\}$)

7.4. THUẬT TOÁN TRI-TRAINING

Một đề xuất khác cho giải thuật học cộng tác để xóa bỏ hạn chế là phải dùng 2 giải thuật học khác nhau của tác giả Zhou [Zhou05]. Đề xuất của Zhou cho phép ta không phải chia tập đặc trưng ra làm 2 khung nhìn khác nhau cũng như cũng không phải sử dụng nhiều giải thuật học khác nhau. Trong giải thuật mới của Zhou, thay vì sử dụng 2 bộ phân lớp huấn luyện lẩn nhau thì Zhou đề xuất sử dụng 3 bộ phân lớp, và cứ 2 bộ phân lớp lại chịu trách nhiệm “huấn luyện” bộ phân lớp còn lại. Giải thuật mới của Zhou được đặt tên là Tri-Training. Chi tiết của giải thuật này được mô tả như sau:

- Đầu vào:* + tập dữ liệu gán nhãn L ;
 + tập dữ liệu chưa gán nhãn U ;
 + một giải thuật phân lớp có giám sát *Learn*;
- Đầu ra:* bộ phân lớp kết hợp đã được huấn luyện;

Thuật toán Tri-training

//Bước khởi tạo

for $i = 1..3$ do

$S_i \leftarrow \text{BootstrapSample}(L);$

$h_i \leftarrow \text{Learn}(S_i);$

$e_i' \leftarrow 0.5;$

$l_i' \leftarrow 0;$

endfor

//Bước học

repeat until không có h_i nào thay đổi

for $i = 1..3$ do

$L_i \leftarrow \emptyset;$

$update_i \leftarrow \text{false};$

$e_i \leftarrow \text{MeasureError}(h_j \& h_k) (j, k \neq i);$

if ($e_i < e_i'$) then

for every $x \in U$ do

if ($h_j(x) = h_k(x)$) ($j, k \neq i$) then

$L_i \leftarrow L_i \cup \{(x, h_i(x))\}$

endfor

if ($|L_i| = 0$) then $l_i' \leftarrow \left\lfloor \frac{e_i}{e_i' - e_i} + 1 \right\rfloor$

if ($|L_i| < |L_i'|$) then

if ($e_i | L_i | < e_i' | L_i' |$) then

$update_i \leftarrow \text{true};$

else if ($|L_i'| > \frac{e_i}{e_i' - e_i}$) then

$L_i \leftarrow \text{Subsample}(L_i, \left\lceil \frac{e_i' l_i'}{e_i} - 1 \right\rceil);$

$update_i \leftarrow \text{true};$

```

endfor

for  $i = 1..3$  do

    if ( $update_i = \text{true}$ ) then

         $h_i \leftarrow Learn(L \cup L_i);$ 

         $e'_i \leftarrow e_i;$ 

         $l_i \leftarrow |L_i|;$ 

    endfor

endrepeat

Output:  $h(x) \leftarrow \arg \max_{y \in \text{label}} \sum_{i: h_i(x) = y} 1$ 

```

- **Bước khởi tạo:** Vì không sử dụng các khung nhìn (tập con đặc trưng) khác nhau và cũng không sử dụng các giải thuật phân lớp khác nhau, do đó, để tạo cho các bộ phân lớp có khả năng dự đoán độc lập nhau, Tri-Training phải sử dụng phương pháp lấy mẫu (BootstrapSample) để tạo ra các tập huấn luyện ban đầu S_i khác nhau cho 3 bộ phân lớp. Bản chất của thủ tục BootstrapSample là lấy ngẫu nhiên $|L|$ phần tử dữ liệu trong tập L . Chú ý, với cách lấy ngẫu nhiên này thì một phần tử dữ liệu trong tập L có thể được lấy một số lần, ngược lại có phần tử dữ liệu lại không được lấy.
- **Bước học:** Quá trình học được lặp đi lặp lại trong một vòng lặp. Tập dữ liệu không có nhãn U sẽ được hai bộ phân lớp j và k gán nhãn, nếu một phần tử dữ liệu x ($x \in U$) nào đó được hai bộ phân lớp này “đồng thuận”, tức $h_j(x) = h_k(x)$ ($j, k \neq i$) thì phần tử dữ liệu x đó được đưa vào tập dữ liệu tiềm năng ký hiệu là L_i . Tập dữ liệu tiềm sau này sẽ được dùng kết hợp với tập dữ liệu gán nhãn đầu vào L để huấn luyện bộ phân lớp h_i . Một điểm khá thú vị ở đây là tập dữ liệu gán nhãn ban đầu L và tập dữ liệu chưa gán nhãn U sẽ không bị thay đổi trong quá trình học của giải thuật. Một điểm khác biệt nữa là số lượng dữ liệu mới được gán nhãn L_i dùng để kết hợp với dữ liệu gán nhãn

ban đầu L được giới hạn. Thủ tục Subsample sẽ đảm bảo việc loại bỏ một số lượng dữ liệu trong tập L_i để thu được số lượng dữ liệu phù hợp nhất định trước khi dùng để kết hợp với tập dữ liệu L dùng cho việc huấn luyện h_i ($h_i \leftarrow \text{Learn}(L \cup L_i)$). Kết quả cuối cùng của giải thuật là bộ 3 các bộ phân lớp đã được huấn luyện.

- Giải thuật dừng khi các bộ phân lớp không cần phải huấn luyện (update) lại (tức $\text{update}_i \leftarrow \text{false}$; với mọi i).
- Đầu ra của giải thuật là 3 bộ phân lớp đã được cải tiến và ta kết hợp cả 3 bộ phân lớp này để dự đoán nhãn của một phần tử dữ liệu mới. Xác suất một phần tử dữ liệu mới x thuộc vào lớp C sẽ là tổng xác suất $P(x | C)$ của 3 bộ phân lớp h_i , lớp nào có xác suất $P(x | C)$ lớn nhất sẽ được dùng làm nhãn cho phần tử x .

Giải thuật Tri-Training được đề xuất và chứng minh được tính hiệu quả của nó, tuy nhiên trong một số bài toán cụ thể, khi mà số lượng các phần tử dữ liệu trong một lớp C nào đó có thể có rất ít. Khi thực hiện việc lấy mẫu BootstrapSample thì các phần tử dữ liệu thuộc lớp C đó có thể bị bỏ qua, dẫn đến khả năng lớp C sẽ không có dữ liệu để huấn luyện, dẫn đến kết quả phân lớp của từng bộ phân lớp h_i sẽ bị giảm đáng kể, và kết quả cuối cùng sẽ không cao. Một đề xuất khác được đưa ra trong khi gặp phải trường hợp này là thay vì chỉ sử dụng 1 giải thuật phân lớp, ta có thể sử dụng nhiều giải thuật phân lớp khác nhau, do đó ta có thể sử dụng toàn bộ tập dữ liệu gán nhãn gốc L ban đầu để huấn luyện các bộ phân lớp [Nguyen08]. Đề xuất này đã được thử nghiệm trong bài toán phân lớp câu hỏi và đã đem lại kết quả khả quan. Một chú ý trong trường hợp này là phải chọn các giải thuật phân lớp sao cho chất lượng phân lớp của chúng phải tương đương nhau.

7.5. THUẬT TOÁN TỰ HUẤN LUYỆN (SHELF-TRAINING)

Giải thuật tự huấn luyện là một phương pháp được sử dụng phổ biến trong học bán giám sát vì tính đơn giản của nó. Ta có thể

coi giải thuật tự học là trường hợp đặc biệt của giải thuật học cộng tác, trong đó ta chỉ sử dụng một khung nhìn (view) cho cả 2 bộ phân lớp. Trong giải thuật tự huấn luyện một bộ phân lớp ban đầu được huấn luyện cùng với số lượng nhỏ dữ liệu gán nhãn L . Tập phân lớp sau đó sẽ được dùng để gán nhãn cho dữ liệu chưa gán nhãn U để thêm vào tập L dùng để huấn luyện giải thuật trong vòng lặp sau. Sau đó tập phân lớp sẽ được huấn luyện lại trên tập dữ liệu L mới, quy trình này được lặp đi lặp lại để làm tăng chất lượng của bộ phân lớp. Chú ý rằng tập phân lớp sử dụng các dự đoán của nó để dạy chính nó, do đó nó có tên là tự học. Quy trình này còn được gọi là shelf-teaching hay là bootstrapping. Thuật toán EM địa phương được trình bày phía trên là dạng đặc biệt của shelf-training.

Shelf-training được áp dụng để xử lý các bài toán của một số ngôn ngữ tự nhiên. Ngoài ra shelf-training còn được áp dụng để phân tách và dịch máy. Trong thuật toán shelf-training, sử dụng một thuật toán phân lớp giám sát h gọi là thuật toán "nền" của thuật toán shelf-training.

Đầu vào: + tập các dữ liệu gán nhãn L .

+ tập các dữ liệu chưa gán nhãn U

Đầu ra: tập dữ liệu L đã được làm giàu và bộ phân lớp có chất lượng được cải tiến

Shelf-training

Loop (cho đến khi $U = \emptyset$ hoặc đạt được đủ một số vòng lặp)

Huấn luyện bộ phân lớp giám sát h trên tập L

$h = Learn(L)$

Sử dụng h để phân lớp dữ liệu trong tập U

Tìm tập con $U' \subseteq U$ có độ tin cậy cao nhất:

$$L + U' \Rightarrow L$$

$$U - U' \Rightarrow U$$

Trong nội dung thuật toán tổng quát trên đây còn một vấn đề cần xem xét đây chính là vấn đề tìm tập con U' có "độ tin cậy cao nhất". Trong một số trường hợp có thể sử dụng thủ tục bootstrapping trong thuật toán shelf-training.

7.6. MỘT SỐ ỨNG DỤNG CỦA CÁC GIẢI THUẬT HỌC BÁN GIÁM SÁT

Các giải thuật học bán giám sát ở trên được giới thiệu chủ yếu là dùng cho các bài toán phân lớp. Trong thực tế, các giải thuật học bán giám sát có thể áp dụng cho nhiều bài toán khác nữa chẳng hạn như bài toán nhận dạng thực thể tên (named entity recognition), phân tích cú pháp (syntax parser),... Việc ứng dụng các giải thuật học bán giám sát là ở những trường hợp ta chỉ có thể thu được một tập nhỏ dữ liệu gán nhãn và có rất nhiều dữ liệu không gán nhãn. Ngoài ra trong một số trường hợp khác ta có một tập dữ liệu gán nhãn đủ lớn nhưng vẫn áp dụng các giải thuật học bán giám sát để làm tăng chất lượng của hệ thống. Chương này sẽ không đi chi tiết vào từng ứng dụng cụ thể của giải thuật này.

CÂU HỎI VÀ BÀI TẬP

- 7.1. Viết giải thuật tự huấn luyện shelf-training với bộ phân lớp Naive Bayes (được viết trong chương 6). Chọn một tập dữ liệu đi kèm với phần mềm weka, chia tập dữ liệu này thành 3 phần: một phần làm dữ liệu có nhãn, một phần làm dữ liệu không gán nhãn và phần còn lại làm dữ liệu kiểm thử. Thực nghiệm giải thuật trên với tập dữ liệu vừa tạo và đánh giá xem chất lượng của nó có được cải thiện so với ban đầu hay không. Thay đổi tỉ lệ giữa dữ liệu có nhãn và dữ liệu không có nhãn để thấy sự ảnh hưởng của nó đến chất lượng giải thuật.
- 7.2. Thực hiện công việc tương tự như bài 1 nhưng dùng giải thuật co-training.

- 7.3. Thực hiện công việc tương tự như bài 1 nhưng dùng giải thuật co-EM.
- 7.4. Thực hiện công việc tương tự như bài 1 nhưng dùng giải thuật Multi-Classifier.
- 7.5. Thực hiện công việc tương tự như bài 1 nhưng dùng giải thuật Tri-Training.

Chương 8.

KHAI PHÁ DỮ LIỆU BẢO VỆ TÍNH RIÊNG TƯ

Trước hết, yêu cầu bảo vệ tính riêng tư trong khai phá dữ liệu có xuất phát điểm từ nhận thức của người sử dụng. Rakesh Agrawal và Ramakrishnan Srikant [AS00] cho biết, theo thống kê của một số tác giả trước đó, hầu hết người sử dụng Web quan tâm tới việc bảo vệ tính riêng tư, trong đó có 17% số người tuyệt đối không cung cấp thông tin riêng tư, 56% số người bằng lòng cung cấp với điều kiện biện pháp bảo vệ tính riêng tư và chỉ có 27% số người sẵn sàng cung cấp thông tin. Có tới 86% người sử dụng web cho rằng việc con người cung cấp thông tin nhằm thu nhận một lợi ích nào đó là sự lựa chọn cá nhân. Có tới 82% người dùng Web coi trọng chính sách bảo mật tính riêng tư trong các hệ thống. Đồng thời, yêu cầu về mức độ bảo mật tính riêng tư đối với các thuộc tính khác nhau (họ tên, nghề nghiệp, lứa tuổi, sở thích, nơi cư trú...) là khác nhau.

Sau nữa, và có thể là khía cạnh quan trọng hơn, tính riêng tư (còn được gọi là quyền tự do riêng tư) của cá nhân và tổ chức được pháp luật bảo vệ. Yêu cầu bảo vệ tính riêng tư quy định hoạt động khai phá dữ liệu cần được tiến hành đúng pháp luật về việc không vi phạm quyền tự do riêng tư được pháp luật bảo vệ.

Trong nhiều miền ứng dụng liên quan tới thông tin cá nhân, chẳng hạn như các CSDL về công dân và dữ liệu mạng xã hội, nảy sinh các quan ngại của cộng đồng về khả năng khai phá dữ liệu có thể vi phạm tính riêng tư của cá nhân và tổ chức. Quan ngại nói trên càng được tăng thêm khi xảy ra một số tình huống khuếch trương quá mức các kết quả khai phá dữ liệu mà sự khuếch trương đó đã đụng chạm tới tính riêng tư của người dùng. Gregory

Piatetsky-Shapiro [Shap95] đưa ra hai ví dụ cho tình huống nói trên. Trong ví dụ thứ nhất, khi mẫu khách hàng mua loại sản phẩm cụ thể được phát hiện, các chiến dịch quảng cáo khuyến mại định hướng khách hàng quá mức đã gây ra sự phiền toái rất khó chịu cho khách hàng. Trong ví dụ thứ hai, kế hoạch bán một CD-ROM chứa dữ liệu của 100 triệu hộ gia đình Mỹ (hơn 120 triệu khách hàng) của hãng LOTUS bị phá sản vì gấp phải lùn sóng (được ví như "bão") phản đối.

Chính vì lý do đó, ngay từ đầu những năm 1990, bảo vệ tính riêng tư trong khai phá dữ liệu được đặt ra như là một chủ đề nghiên cứu quan trọng. Vào năm 1995, tạp chí IEEE Expert đã tiến hành một diễn đàn nhỏ về chủ đề này, trong đó Gregory Piatetsky-Shapiro [Shap95] và Daniel O'Leary [Leary95] đưa ra một số luận điểm có tính khái quát về chủ đề bảo vệ tính riêng tư trong khai phá dữ liệu. Mục này trình bày các nội dung liên quan tới yêu cầu bảo vệ tính riêng tư trong khai phá dữ liệu theo hai khía cạnh: khía cạnh pháp luật và khía cạnh công nghệ.

8.1. KHÍA CẠNH PHÁP LUẬT BẢO VỆ TÍNH RIÊNG TƯ VÀ KHAI PHÁ DỮ LIỆU

Mỗi quốc gia đều có quy định về bảo vệ tính riêng tư phù hợp với truyền thống văn hóa, đạo đức xã hội và thể chế chính trị của quốc gia đó. Trong thời đại hội nhập quốc tế hiện nay, nội dung cốt lõi về bảo vệ tính riêng tư của các quốc gia trên thế giới là tương đồng nhau. Dưới đây trình bày một số nội dung theo tiếp cận pháp luật về yêu cầu bảo vệ tính riêng tư trong khai phá dữ liệu của Tổ chức Hợp tác kinh tế và Phát triển (the Organization for Economic Cooperation and Development: OECD) và các quốc gia Bắc Mỹ, những khu vực có lĩnh vực khai phá dữ liệu rất phát triển.

8.1.1. Hướng dẫn của OECD về dữ liệu riêng tư và tác động tới hoạt động phát hiện tri thức từ dữ liệu

Hướng dẫn OECD về dữ liệu riêng tư cho phép sử dụng dữ liệu xuyên biên giới với tám điều khoản chính (nguyên tắc) bảo vệ

tính riêng tư trong dữ liệu. Nội dung bảo vệ tính riêng tư trong khai phá dữ liệu cần được đối chiếu với tám nguyên tắc dưới đây theo hướng dẫn OECD.

Daniel O'Leary [Leary95] khảo sát hướng dẫn của OECD về dữ liệu riêng tư và liên hệ hướng dẫn này với khuynh hướng phát hiện tri thức. Hướng dẫn của OECD gồm tám nguyên tắc:

- (1). Nguyên tắc giới hạn thu thập: Dữ liệu nên được thu được hợp pháp và công bằng, trong đó một số dữ liệu rất nhạy cảm không nên nắm bắt. Tính "nhạy cảm" có thể chứa các thông tin về tôn giáo, chủng tộc, quốc gia gốc... nhưng thường không được xác định tường minh. Nguyên tắc giới hạn thu thập có thể làm hạn chế phạm vi của phát hiện tri thức. Nếu dữ liệu được cho là "nhạy cảm" thì không có quyền truy nhập dữ liệu này, nếu dữ liệu có tính nhạy cảm tự nhiên (nhưng không được xếp vào nhạy cảm) thì khai phá dữ liệu có thể dẫn tới hậu quả.
- (2) Nguyên tắc chất lượng dữ liệu: Dữ liệu có liên quan đến mục đích sử dụng, chính xác, đầy đủ và cập nhật; các biện pháp thích hợp cần được thực hiện để đảm bảo chính xác, nguyên tắc này. Nguyên tắc chất lượng dữ liệu lưu ý phân biệt dữ liệu gốc và dữ liệu chế biến. Nhìn chung dữ liệu chế biến không nên lưu trữ, nếu như cần lưu trữ thì cần phải cập nhật theo dữ liệu gốc. Thêm nữa, cần định ra các tiêu chí chuẩn đảm bảo chất lượng dữ liệu cho phát hiện tri thức.
- (3). Nguyên tắc đặc tả mục đích: Mục đích sử dụng dữ liệu cần được xác định và các dữ liệu bị phá hủy nếu chúng không còn phục vụ cho mục đích. CSDL chỉ được sử dụng cho mục đích đã được tuyên bố và điều này hạn chế khả năng phát hiện tri thức, đặc biệt trong trường hợp sử dụng các thông tin riêng tư.
- (4). Nguyên tắc giới hạn sử dụng: Sử dụng dữ liệu cho các mục đích khác so với đặc tả chỉ có thể được tiến hành khi

có đồng ý của đối tượng dữ liệu hoặc của cơ quan pháp luật. Do đó, cần có các tuyên bố về việc sử dụng dữ liệu cho phát hiện tri thức trong các thỏa thuận của đối tượng dữ liệu ngay từ khi họ tham gia vào hệ thống CSDL.

- (5). Nguyên tắc bảo vệ: Thủ tục bảo vệ chống lại mất mát, hư hỏng, tiêu hủy, hoặc sử dụng lạm dụng dữ liệu. Trong một số tình huống thì khai phá dữ liệu có vẻ như là một sự lạm dụng dữ liệu. Việc xác định tính bản quyền dữ liệu trong khai phá dữ liệu là rất quan trọng.
- (6). Nguyên tắc mở: Cần thông báo mở về việc thu thập, lưu trữ và sử dụng dữ liệu cá nhân. Phát hiện tri thức từ dữ liệu cũng cần được tuyên bố cho đối tượng dữ liệu. Nguyên tắc mở cho phép kiểm soát hoạt động phát hiện tri thức không phù hợp.
- (7). Nguyên tắc sự tham gia của cá nhân: Các đối tượng dữ liệu có quyền truy cập và phản đối các dữ liệu liên quan đến họ. Nguyên tắc này cũng yêu cầu tăng cường tính tin cậy của các phương pháp phát hiện tri thức được sử dụng.
- (8). Nguyên tắc trách nhiệm: Cần có một cơ chế thi hành (bộ điều khiển) chịu trách nhiệm thi hành tất cả các nguyên tắc trên đây. Nguyên tắc này cũng đòi hỏi một mức độ kiểm soát đối với hoạt động phát hiện tri thức từ dữ liệu.

Cũng liên quan tới hướng dẫn của OECD trên đây, Willi Kloesgen [Kloe95] chỉ ra một số vấn đề cần quan tâm để bảo vệ tính riêng tư trong phát hiện tri thức từ dữ liệu. Vấn đề thứ nhất là quyền truy nhập dữ liệu nguyên thủy liên quan đến thông tin cá thể (người, công ty, giao dịch). Vấn đề thứ hai là đầu ra của phát hiện tri thức (mẫu/tri thức phát hiện được) có thể đưa tới các hành vi của bên thứ ba ảnh hưởng tới một cá nhân thuộc nhóm người liên quan tới mẫu đó. Chẳng hạn, một mẫu phát hiện được về nhóm người có nguy cơ cao về bệnh tật có thể dẫn tới tình huống người quản lý có hành động buộc thôi việc một nhân viên có thể thuộc nhóm tương ứng với mẫu nói trên.

8.1.2. Tiếp cận pháp luật bảo vệ tính riêng tư tại nước Mỹ và tác động tới khai phá dữ liệu

Đảm bảo tính riêng tư trong khai phá dữ liệu còn dẫn tới cuộc chiến pháp lý ở nước Mỹ. Đầu năm 2003, hai Thượng nghị sỹ Mỹ Jon Corzine và Ron Wyden đề xuất dự luật S.188 năm 2003 về việc nghiêm cấm khai phá dữ liệu. Tiếp đó, Tiểu ban chính sách công thuộc Hiệp hội máy tính (the Association for Computing Machinery's U.S. Public Policy Committee: USACM) của Mỹ có bài viết bày tỏ sự lo ngại về rủi ro an ninh (Security Risks), rủi ro riêng tư (Privacy Risks), rủi ro kinh tế (Economic Risks), rủi ro cá nhân (Personal Risks) đối với *Chương trình dự đoán thông tin toàn bộ* (the Total Information Awareness (TIA) Program) do Bộ Quốc phòng Mỹ đầu tư.

Quan ngại tác động tiêu cực của hai sự kiện trên đối với sự phát triển của lĩnh vực khai phá dữ liệu và phát hiện tri thức, Ban điều hành Tiểu ban Phát hiện tri thức và Khai phá dữ liệu của ACM (Executive Committee on ACM Special Interest Group on Knowledge Discovery and Data Mining) đã công bố một bài viết [Kim03] nhấn mạnh việc phân biệt công nghệ khai phá dữ liệu từ tập hợp dữ liệu với các ứng dụng cụ thể trong một miền ứng dụng cụ thể và khẳng định rằng khai phá dữ liệu không chống lại sự tự do cá nhân. Bởi nhiều lý do, dự luật S.133 không được thông qua, tuy nhiên, sự kiện về dự luật này cũng đặt ra yêu cầu cấp thiết là phải tăng cường nhận thức và hành động để bảo vệ tính riêng tư trong khai phá dữ liệu.

Dưới đây là một số luận điểm chính trong khẳng định khai phá dữ liệu không vi phạm quyền tự do cá nhân của Tiểu ban Phát hiện tri thức và Khai phá dữ liệu của ACM:

- Một dự án phát hiện tri thức lớn đòi hỏi sự tham gia của rất nhiều công nghệ mà công nghệ khai phá dữ liệu chỉ là một trong số đó. Đối với chương trình TIA trên đây, các công nghệ tham gia bao gồm quản trị CSDL, phân tích xử lý trực tuyến, nhận dạng giọng nói, xác nhận hình ảnh (khuôn mặt, mống mắt, vân tay...),

hiểu ngôn ngữ tự nhiên, lưu trữ dữ liệu, tích hợp dữ liệu... và khai phá dữ liệu. Nếu có xảy ra hiện tượng vi phạm quyền riêng tư thì điều đó có nguyên nhân từ mọi công nghệ trên đây mà không thể kết luận vi phạm đó là chỉ do công nghệ khai phá dữ liệu;

- Công nghệ khai phá dữ liệu dựa trên nền tảng có xuất xứ lâu dài của phân tích thống kê và trí tuệ nhân tạo song không phải đã hoàn hảo. Tính không hoàn hảo cũng đặt ra đối với mọi công nghệ. Giải pháp tốt để giảm thiểu những điểm chưa hoàn hảo của công nghệ khai phá dữ liệu là cần tăng cường nghiên cứu và triển khai ứng dụng công nghệ này.

- Để phù hợp với yêu cầu đảm bảo quyền riêng tư thì càng cần phát triển các nghiên cứu và triển khai ứng dụng về khai phá dữ liệu liên quan, đặc biệt là khai phá dữ liệu bảo mật dữ liệu và đảm bảo quyền riêng tư (data security and privacy-preserving data mining).

Khái niệm về khai phá dữ liệu bảo vệ tính riêng tư

Trong hầu hết các nghiên cứu, các tác giả chấp nhận tính riêng tư có tính khái quát với sự đa dạng ngữ nghĩa. Một số nghiên cứu cố gắng nhận diện nội dung ngữ nghĩa chi tiết hơn của tính riêng tư trong các miền ứng dụng cụ thể. Mina Deng [Deng10], Fahriye Seda Gurses [Guses10] cung cấp các khía cạnh khác nhau của tính riêng tư hoặc trong hệ thống bảo vệ nội dung hoặc trong miền ứng dụng mạng xã hội trực tuyến.

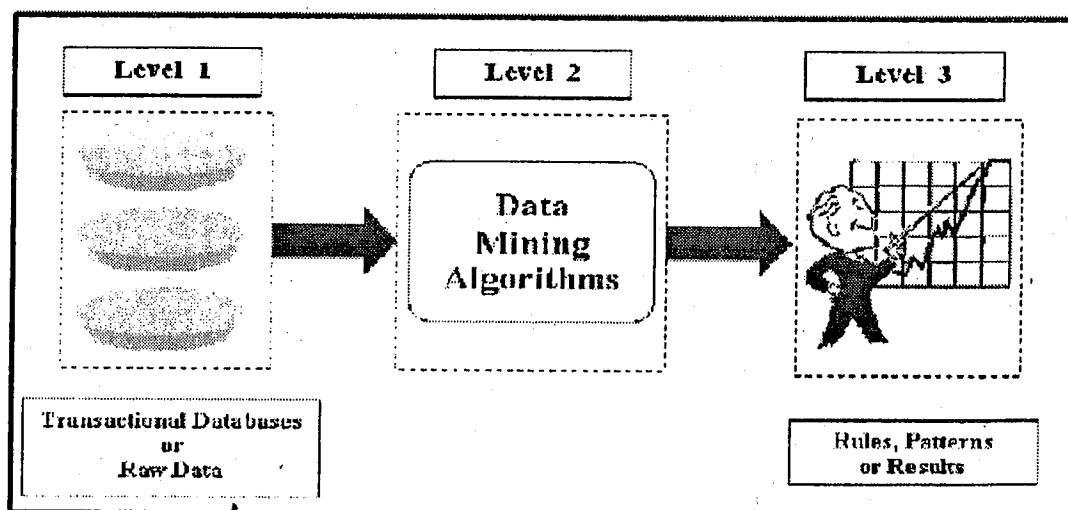
Khái niệm về khai phá dữ liệu bảo vệ tính riêng tư được Chris Clifton và cộng sự [CKV04] xác định qua một số định nghĩa và phân tích có căn cứ. Theo Verykios V. S. và cộng sự [VBFPS04], khai phá dữ liệu bảo vệ tính riêng tư là khai phá dữ liệu trong đó thuật toán khai phá dữ liệu (thuật toán khai phá dữ liệu bảo vệ tính riêng tư) phải được phân tích để giải quyết đối với các tác dụng phụ nảy sinh từ dữ liệu riêng tư. Thuật toán khai phá dữ liệu quan tâm tới tính riêng tư theo hai mức chính. Ở mức thứ nhất, dữ liệu thô nhạy cảm từ CSDL nguồn như định danh, tên, địa chỉ... nên được sửa đổi hoặc cắt bỏ trong giai đoạn chọn lọc dữ

liệu để thuật toán khai phá dữ liệu không có khả năng làm tổn thương tính riêng tư của cá nhân người khác. Ở mức thứ hai, tri thức phát hiện được mà có khả năng làm tổn thương tới sự riêng tư của người khác cũng phải được loại trừ. Như vậy, khai phá dữ liệu bảo vệ tính riêng tư là khai phá dữ liệu mà dữ liệu nguồn được biến đổi theo một cách nào đó để dữ liệu riêng tư và tri thức riêng tư đạt được độ tin cậy sau quá trình khai phá dữ liệu. Khai phá dữ liệu riêng tư còn giải quyết vấn đề "thừa kế CSDL" về việc rò rỉ các thông tin bí mật từ sự truy cập của người sử dụng không được phép. Khai phá dữ liệu bảo vệ tính riêng tư còn bao gồm đảm bảo tin cậy tính riêng tư của một tổ chức.

8.2. PHƯƠNG PHÁP KHAI PHÁ DỮ LIỆU BẢO VỆ TÍNH RIÊNG TƯ

8.2.1. Mô hình và phương pháp khai phá dữ liệu bảo vệ tính riêng tư

Các giải pháp công nghệ thi hành khai phá dữ liệu bảo vệ tính riêng tư cần được nghiên cứu, phát triển và sử dụng nhằm đáp ứng độ tin cậy của dữ liệu và tri thức trong quá trình khai phá dữ liệu. Trong [Yasien07], Ahmed HajYasien trình bày ba mức chính thi hành việc bảo vệ tính riêng tư trong thuật toán khai phá dữ liệu là mức dữ liệu, mức thuật toán khai phá dữ liệu, và mức lựa chọn và trình diễn tri thức kết quả (Hình 8.1).



Hình 8.1. Ba mức chính khai phá dữ liệu bảo vệ tính riêng tư [Yasien07]

Jaideep Shrikant Vaidya [Vaidya04] trình bày bốn mô hình tin cậy trong khai phá dữ liệu bảo vệ tính riêng tư:

(i) *mô hình tin cậy bên thứ ba (Trust Third Party Model)*: Trong mô hình này, việc bảo vệ tính riêng tư (mọi tính toán liên quan) được giao phó cho một bên thứ ba. Đây là một mô hình lý tưởng và điều quan trọng nhất là tìm ra được một bên thứ ba được tất cả mọi người tin cậy. Bên thứ ba nói trên có thể là từ một bên tham gia được lựa chọn theo cơ chế nào đó, chẳng hạn cơ chế tương tự như cơ chế bầu thủ lĩnh trong truyền thông phân tán; (ii) *mô hình nửa tin cậy (Semi-honest Model)*: Trong mô hình này, các bên tham gia khai phá dữ liệu thi hành hoạt động bảo vệ tính riêng tư theo giao thức của thuật toán khai phá dữ liệu. Mỗi bên tham gia thực hiện các bước của giao thức theo quy trình đã định. Giao thức bảo vệ tính riêng tư cho phép các bên tham gia tự do sử dụng dữ liệu đầu vào nhận được (tuy nhiên, dữ liệu này đã được biến đổi tại nơi cung cấp). Mô hình nửa tin cậy là mô hình phổ biến hiện thời; (iii) *mô hình độc hại (Malicious Model)*: Trong mô hình này, không có hạn chế nào đối với các bên tham gia, mỗi bên tham gia thực hiện bất kỳ điều gì mà nó có lợi nhất và phải tự chịu trách nhiệm bảo vệ tính riêng tư. Có ba khả năng đi chệch khỏi giao thức: Một bên từ chối tham gia vào giao thức khi giao thức này được gọi lần đầu tiên, một bên có thể thay đổi đầu vào cục bộ của nó bằng cách thâm nhập giao thức với một đầu vào khác với mà nó được cung cấp, một bên có thể hủy bỏ sớm hơn dự kiến. Xây dựng giao thức theo mô hình độc hại là rất khó khăn, song mô hình độc hại hướng tới các ứng dụng mà mô hình nửa tin cậy không thi hành được; (iv) *Mô hình tương hợp khuyến khích (Incentive Compatibility)*: Mô hình này dựa trên giao thức tương hợp khuyến khích là giao thức có khả năng phát hiện bên gian lận hoặc bên chịu thiệt khi cùng tham gia bảo vệ tính riêng tư.

Theo Verykios V. S. và cộng sự [VBFPS04], khai phá dữ liệu bảo vệ tính riêng tư được tham chiếu theo năm chiều sau đây:

- Phân tán dữ liệu: Xem xét nguồn dữ liệu là tập trung hay phân tán, phân tán theo chiều dọc (mỗi thành phần chứa toàn bộ

đối tượng song chỉ với một bộ phận thuộc tính) hay chiều ngang (mỗi thành phần chứa toàn bộ các thuộc tính song chỉ chứa một bộ phận đối tượng) hay theo cả chiều dọc lẫn chiều ngang. Lưu ý rằng khai phá dữ liệu bảo vệ tính riêng tư khi phân tán dữ liệu thì yêu cầu bảo vệ tính riêng tư đòi hỏi với cả tác nhân khai phá dữ liệu lẫn các tác nhân thành phần dữ liệu. Nội dung chính của các thuật toán khai phá dữ liệu bảo vệ tính riêng tư phân tán là xây dựng các giao thức bảo vệ tính riêng tư khi khai phá dữ liệu tại các thành phần và sau đó bảo vệ tính riêng tư khi kết hợp kết quả.

- Biến đổi dữ liệu: Sử dụng các sơ đồ sửa đổi dữ liệu mà theo đó giá trị ban đầu của dữ liệu trong CSDL được sửa đổi nhằm nhận được sự bảo vệ tính riêng tư cao. Kỹ thuật sửa đổi dữ liệu phải phù hợp với chính sách bảo mật của tổ chức quản lý dữ liệu. Một số phương pháp sửa đổi điển hình là: (i) Làm nhiễu (perturbation) dữ liệu bằng cách thay giá trị thuộc tính thành giá trị mới (chẳng hạn, giá trị 1 thay đổi thành giá trị 0, hoặc bổ sung nhiễu), (ii) chặn (blocking): thay một giá trị thuộc tính hiện có bằng ký hiệu "?", (iv) tổng hợp/hợp nhất: nhóm một số giá trị thuộc tính vào một lớp thô có độ trừu tượng, (v) tráo đổi: tráo đổi các giá trị của các bản ghi cá nhân, và (vi) chọn mẫu để cho phép phát hành dữ liệu.

- Thuật toán khai phá dữ liệu bảo vệ tính riêng tư: Ở đây, thuật toán che giấu dữ liệu được bao gói như một thành phần của thuật toán khai phá dữ liệu. Chiều này cho phép thi hành linh hoạt việc sửa đổi dữ liệu ngay trong thuật toán khai phá dữ liệu nhưng sẽ gặp khó khăn không nhỏ khi thi hành giải pháp nhúng thuật toán che giấu dữ liệu.

- Che giấu dữ liệu hoặc luật: Đề cập đến việc che giấu dữ liệu nguồn hoặc dữ liệu tổng hợp đã qua biến đổi. Do tính phức tạp của việc che giấu dữ liệu tổng hợp cho nên các kỹ thuật thi hành chiều này thường chỉ được thi hành theo kiểu chẩn đoán chính sách che giấu thông tin.

- Bảo vệ tính riêng tư: Thay vì tiến hành sửa đổi theo bộ quy tắc nhất quán trên toàn bộ CSDL đầu vào, kỹ thuật bảo vệ tính

riêng tư tiến hành sửa đổi một cách chọn lọc các dữ liệu có tính riêng tư trực tiếp hoặc gián tiếp. Các tác giả cho rằng đây là chiêu quan trọng nhất và kỹ thuật khai phá dữ liệu tiếp cận theo chiều này có khả năng đảm bảo tốt nhất tính riêng tư. Một thuật toán khai phá dữ liệu bảo vệ tính riêng tư có thể thi hành theo một hoặc một số chiêu trên đây. Các thuật toán khai phá dữ liệu bảo vệ tính riêng tư được tiếp cận theo các phương pháp là dựa theo kinh nghiệm (Heuristic-Based Methods), dựa trên mã hóa (Cryptography-Based Methods), dựa trên tái thiết kế (Reconstruction-Based Methods).

Phương pháp dựa theo kinh nghiệm được xem xét dựa trên một quan niệm rằng biến đổi hoặc tinh chế dữ liệu chọn lọc là một bài toán NP-khó và vì vậy việc chuẩn đoán dựa trên kinh nghiệm sẽ có lợi thế trong việc giải quyết vấn đề loại bỏ một số mẫu (tri thức) nhạy cảm từ kết quả của thuật toán khai phá dữ liệu.

Phương pháp dựa trên mã hóa được đặt ra trong tình huống thuật toán khai phá dữ liệu được tiến hành trên cơ sở cộng tác của nhiều bên tham gia. Mỗi bên có dữ liệu riêng của mình, có đủ năng lực thi hành thuật toán khai phá dữ liệu cho dữ liệu riêng của mình song không muốn công bố kết quả khai phá dữ liệu trọn vẹn của bản thân cho các bên còn lại. Đây là một trường hợp của bài toán tính toán an toàn nhiều bên (Secure Multiparty Computation - SMC). Bài toán khai phá dữ liệu bảo vệ tính riêng tư thuộc trường hợp này có tính phổ biến. Một vài thuật toán theo phương pháp này sẽ được giới thiệu.

Trong phương pháp tái thiết kế, dữ liệu nguồn được tạo nhiều và đầu vào cho khai phá dữ liệu là các bản tổng hợp từ dữ liệu nguồn đã được gộp.

8.2.2. Một số thuật toán khai phá dữ liệu bảo vệ tính riêng tư

Là một nội dung nghiên cứu được đề cập rất sớm song do tính "nhạy cảm" của dữ liệu riêng tư cho nên chỉ tới mấy năm gần

đây với sự phát triển mạnh mẽ của phương tiện xã hội và truyền thông không dây, sự quan tâm tới khai phá dữ liệu bảo vệ tính riêng tư có sự tăng trưởng nhảy vọt. Một số hội nghị khoa học tầm cỡ thế giới dành riêng cho chủ đề này đã được tổ chức, chẳng hạn như the ACM 2011 Workshop on Privacy in the Electronic Society²³. Tạp chí VLDB dành một số phát hành tháng 11/2006 (số 15(4)) cho quản lý tính riêng tư trong CSDL với hai bài báo chuyên về khai phá dữ liệu bảo vệ tính riêng tư. Tạp chí KDD Letter năm 2011 dành một số cho chủ đề Privacy in Mobility Data Mining²⁴. Chủ đề này thu hút sự quan tâm nhiều năm của một số nhóm nghiên cứu trên thế giới, trong đó có nhóm nghiên cứu tại Purdue University (Christopher W. Clifton, Jaideep Vaidya²⁵ và cộng sự). Đã có sáng chế khai phá dữ liệu bảo vệ tính riêng tư được đăng ký, chẳng hạn, sáng chế "Method, Apparatus and Computer Program Product for Preserving" Patent US 7,904,471 B2 (Mar 8 2011) của Charu Aggarwal và Philip S. Yu.

Nhìn chung, thuật toán khai phá dữ liệu bảo vệ tính riêng tư có nội dung cơ bản như thuật toán khai phá dữ liệu tương ứng ngoại trừ cơ chế bảo vệ tính riêng tư dựa trên các giao thức. Vì vậy, thuật toán khai phá dữ liệu tập phổ biến bảo vệ tính riêng tư theo mô hình nửa tin cậy [Vaidya04] được chọn làm mẫu cho các thuật toán khai phá dữ liệu bảo vệ tính riêng tư.

Bài toán được phát biểu như sau. Một CSDL giao dịch được phân tán theo chiều dọc với k ($k > 2$) thành phần P_1, P_2, \dots, P_k . Cần tìm tất cả các tập phổ biến với cơ chế bảo vệ tính riêng tư theo mô hình nửa tin cậy.

Hình 8.2 mô tả thuật toán tìm tất cả các tập phổ biến bảo vệ tính riêng tư [Vaidya04]. Nội dung cơ bản của thuật toán này

²³ <http://wpes11.rutgers.edu/>

²⁴ <http://www.sigkdd.org/explorations/issue.php?volume=13&issue=1&year=2011&month=07>

²⁵ <http://www.cs.purdue.edu/homes/clifton/> và <http://cimic.rutgers.edu/~jsvaidya/>

chính là thuật toán Apriori ngoại trừ việc phải sử dụng giao thức 17 hoặc giao thức 18 (dòng 12) để tính toán độ hỗ trợ (số lượng bản ghi) của một tập ứng viên. Như đã trình bày trên đây, giao thức truyền thông giữa các bên tham gia là nội dung cốt lõi của thuật toán khai phá dữ liệu bảo vệ tính riêng tư, vì vậy nội dung chính của thuật toán là giao thức 17 hoặc giao thức 18.

1. $L_1 = \{\text{large 1-itemsets}\}$
2. **for** ($k = 2; L_{k-1} \neq \{\}; k++$) **do**
3. $L_k = \{\}$
4. $C_k = \text{apriori-gen}(L_{k-1})$
5. **for all candidate** $c \in C_k$ **do**
6. **if** all the attributes in c are entirely at any one party P_i **then**
7. party P_i independently calculates $c.count$
8. **else**
9. let P_1 have l_1 of the attributes, ..., P_k have l_k attributes
 $(\sum_{i=1}^k l_i = |c|)$
10. construct S_1 on P_1 's side, ..., S_k on P_k 's side
11. where $S_i = S_{u_1} \cap \dots \cap S_{u_{l_i}}, 1 \leq i \leq k$
12. compute $c.count = |\bigcap_{j=1..k} S_j|$ using Protocol 17 or 18
13. **end if**
14. $L_k = L_k \cup \{c \mid c.count \geq \text{minsup}\}$
15. **end for**
16. **end for**
17. Answer = $\bigcup_k L_k$

Thuật giải 8.1. Thuật toán tìm tập phổ biến bảo vệ riêng tư [Vaidya04]

Điều kiện: số lượng site $k > 2$, mỗi site có tập dữ liệu cục bộ S_i ; Kích thước tối đa của tập cục bộ m , và ngưỡng r được sử dụng để chống lại hiện tượng thăm dò

for all sites i {thao tác được tiến hành song song} **do**

Sinh khóa hash E_i

for $j = |S_i|$ to m do

$S_i \leftarrow S_i \cup \{\text{prefix_not_in_}U.i.j\}$ {chèn vào S_i các item duy nhất trong site đó và không thể nằm trong tập giao}

end for

{Bước 1 – Hashing}

$M \leftarrow \text{Encryp AndPermute}(S_i, E_i)$

Gửi M đến site $i+1 \bmod k$

for $p=1..k-2$ do

$M' \leftarrow \text{Receive from site } i-1 \bmod k$

$M'' \leftarrow \text{EncryptAndPermute}(M', E_i)$

Gửi M'' đến site $i+1 \bmod k$

end for

$M' \leftarrow \text{Receive from site } i-1 \bmod k$

$M'' \leftarrow \text{EncryptAndPermute}(M', E_i)$

Gửi M'' đến mọi site trừ site $i+1 \bmod k$

{Bước 2 Tập giao ban đầu của các tập}

$TS_i \leftarrow \text{nhận từ site } j \text{ với } j \neq i-1$

$TS'_i \leftarrow \bigcap_{p=0, p \neq i-1}^{k-1} TS_p$

if $|TS'_i| < r$ then

Thông báo Hủy (ABORT) cho mọi site {Phát hiện/chống thăm dò}

else

{Bước 3 – Tập giao cuối cùng để tính toán kết quả}

Gửi TS'_i đến site $i+1 \bmod k$

Nhận $TS'_{i-1 \bmod k}$ từ site $i-1 \bmod k$

$TS'_i \leftarrow TS'_i \bigcap TS'_{i-1 \bmod k}$

return $|TS'_i|$

end if

end for

Thuật giải 8.2. Giao thức tính toán an toàn độ hỗ trợ
của các tập mục ứng viên [Vaidya04]

Function EncryptAndPermute(Set M , Key E_k)

Require: M is the input array to be hashed, E_k is the hash key

$C \leftarrow \emptyset$

for all $j \in M$ do

$C \leftarrow C \cup \{E_k(j)\};$

end for

randomly permute C to prevent tracking values

return C

Thuật giải 8.3. Hàm mã hóa và đổi chỗ trong giao thức [Vaidya04]

Thuật giải 8.2 và thuật giải 8.3 mô tả nội dung cơ bản của giao thức 17 được đề cập. Các bên tham gia được kết nối vòng tròn theo thứ tự liệt kê như mô tả bài toán. Mã hóa và giải mã thực hiện theo cơ chế bất đối xứng.

Theo trình bày tại thuật giải 8.2, ngoại trừ giai đoạn sinh khóa khởi thủy (thực hiện song song), thuật toán gồm 3 giai đoạn: băm, chuyển giao đầu và chuyển giao cuối.

Trong giai đoạn băm, các bên tham gia sẽ băm mọi tập mục ứng viên của mình với một khóa riêng chỉ do tự mỗi bên biết và thứ tự các mục dữ liệu được sắp xếp ngẫu nhiên. Như vậy, không một bên tham gia nào xác định được ánh xạ mà bên tham gia phía trước đã thực hiện.

Trong giai đoạn chuyển giao đầu, các bên đi tìm phần giao của các mục dữ liệu (ngoại trừ các mục được xác định cục bộ). Việc băm ngăn ngừa việc dùng học máy để nhận ra giá trị thực từ các tập mục nhận được. Lưu ý rằng chỉ cần phát hiện một tình huống nhỏ thua ngưỡng hỗ trợ thì giao thức kết thúc (tập mục ứng viên bị loại bỏ).

Trong giai đoạn chuyển giao cuối, các bên gửi kết quả tính toán của mình tới hàng xóm kế tiếp và nhận kết quả tính toán từ hàng xóm phía trước. Từ các kết quả này, mỗi viên thành tính toán được lực lượng của tập mục ứng viên về phần mình và trả về kết quả.

Thuật giải 8.4 mô tả thuật toán phân lớp cây quyết định PPID3 bảo vệ tính riêng tư [VCKP08] được phát triển trên cơ sở thuật toán

phân lớp cây quyết định ID3 và một nhóm hàm thành phần IsREmpty(), DistributionCount(), IsSameClass(), AttribMaxInfoGain(), ComputeInfoGain().

PPIID3(): Privacy-Preserving Distributed ID3

Require: Transaction set T partitioned between sites P_1, \dots, P_k

Require: p class values: c_1, c_2, \dots, c_p with P_k holding the class attribute

1. **if** IsREmpty() **then**
2. Continue at site P_k up to the return:
3. $(cnt_1, \dots, cnt_p) \leftarrow \text{DistributionCounts}()$
4. Build a leaf node with distribution (cnt_1, \dots, cnt_p)
5. $\{\text{class} \leftarrow \arg \max_{i=1 \dots p} cnt_i\}$
6. return ID of the constructed node
7. **else if** $\text{clsNode} \leftarrow (\text{at } P_k: \text{IsSameClass}())$ **then**
8. return leaf node as clsNode
9. **else**
10. $\text{Bestsite} \leftarrow \text{AttribMaxInfoGain}()$
11. Continue execution at BestSite
12. Create Interior Node Nd with attribute $Nd.A \leftarrow \text{BestAtt}_{\text{Bestsite}}$
{This is best locally (from AttribMaxInfoGain()), and
globally from line 8}
13. **for** each attribute value $a_i \in Nd.A$ **do**
14. Constraints.set($Nd.A, a_i$) {Update local constraint tuple}
15. $\text{nodeID} \leftarrow \text{PPIID3}()$ {Recurse}
16. $Nd.a_i \leftarrow \text{nodeID}$ {Add appropriate branch to interior node}
17. **end for**
18. Constraints.set($A, ?$) {Returning to parent: should no longer filter transactions with A }
19. Store Nd locally keyed by Node ID
20. return Node ID of interior node Nd {Execution time at site owning parent node}
21. **end if**

Giao thức thu thập dữ liệu ẩn danh k phía (k-anonymity) bốn pha do Sheng Zhong và cộng sự [ZYC09] đề xuất có tiềm năng phát triển các giao thức khai phá dữ liệu bảo vệ tính riêng tư (thuật giải 8.5).

k-Anonymous Data Collection Protocol – The basis solution

Let the miner's private key be x and his public key be $y=g^x$

Let the DCH's private key be u and hist public key be $v=g^u$

1. Phase 1: Data submission

2. **for** each respondent i **do**

3. i picks r_i^+, r_i^- uniformly and independently

4. i encrypts her data using public key yv :
 $d_i^+ = E_{yv}(d_i^+, r_i^+); d_i^- = E_{yv}(d_i^-, r_i^-)$

5. i submits d_i^+, d_i^- to the miner

6. **endfor**

7. Phase 2. Miner's randomization operations

8. **for** each pair (i, j) **do**

9. The miner computes $\bar{q}_{ij} = (\overline{d_{ij}^+} / \overline{d_{ij}^-})^{r_{ij}}$, where each r_{ij} is chosen uniformly and independently

10. **endfor**

11. **for** each i **do**

12. The miner chooses a permuation θ_i on $\{1, \dots, N\}$ uniformly at random

13. **for** each $j \in \{1, 2, \dots, N\}$ **do**

14. The miner computes $\bar{q}_{i,j} = \overline{q_{i,\theta_i(j)}} / \overline{q_{i,j}} [1] = \overline{q_{i,j}} [1] / (\overline{q_{i,j}} [2])^x$ and sets $\overline{q_{i,j}} [2] = \overline{q_{i,j}} [2]$

15. **endfor**

16. **endfor**

17. The miner sends the DCH: $\{\overline{d_i^+}, \overline{d_i^-}\}_{i=1, \dots, N}, \{\overline{q_{i,j}}\}_{i=1, \dots, N, j=1, \dots, N}$

18. Phase 3. DCH's randomization operations

19. The DCH computes $\{\overline{q_{i,j}}\} = \{\overline{q_{i,j}} [1] / (\overline{q_{i,j}} [2])^u\}$ for each pair

20. **for** each I **do**

21. The DCH counts $c_i = |\{j : \overline{q_{i,j}} = 1\}|$

22. if $c_i < k - 1$ then

23. The DCH sets to an encryption of $(\star, \star, \dots, \star)$ under public key

24. **else**
25. The DCH sets \overline{d}_i^*
26. **end if**
27. **end for**
28. **if** $1 \leq |\{i : c_i < k - 1\}| < k$ **then**
29. The DCH lets C be the smallest c_i that is greater than $k - 1$
30. For all i s.t. $c_i = C$, the DCH sets \overline{d}_i^* to an encryption $(\star, \star, \dots, \star)$ under public key yv ; for all other i , the DCH sets $\overline{d}_i^* = \overline{d}_i^{\dagger}$
31. **end if**
32. **if** $|\{i : c_i < k - 1\}| \geq k$ or $|\{i : c_i < k - 1\}| = 0$ **then**
33. The DCH defines $\overline{d}_i^* = \overline{d}_i^{\dagger}$
34. **end if**
35. **for each** i **do**
36. The DCH computes $\overline{d}_i^{**}[1] = \overline{d}_i^*[1]/(\overline{d}_i^*[2])^u$ and $\overline{d}_i^{**}[1] = \overline{d}_i^{\dagger}[1]/(\overline{d}_i^{\dagger}[2])^u$
37. The DCH defines $\overline{d}_i^{**}[2] = \overline{d}_i^*[2]$ and $\overline{d}_i^{**}[2] = \overline{d}_i^{\dagger}[2]$
38. The DCH chooses a permutation π on $\{1, \dots, N\}$ uniformly at random and computes $\overline{d}_i^{***} = \overline{d}_{\pi(i)}^*$ and $\overline{d}_i^{\diamond} = \overline{d}_{\pi(i)}^{\dagger}$
39. **end for**
40. The DCH sends \overline{d}_i^{***} and $\overline{d}_i^{\diamond}$ to the miner for all i
41. **Phase 3. Decryption**
42. **for each** i **do**
43. The miner decrypts \overline{d}_i^{***} and $\overline{d}_i^{\diamond}$ using his own private key x , where the decryption of \overline{d}_i^{***} is the part of data containing identifying information; the decryption of $\overline{d}_i^{\diamond}$ is the part of data without identifying information
44. **end for**
45. If the data needs to be published, the miner must publish it in a randomized order

CÂU HỎI VÀ BÀI TẬP

- 8.1.** Khái niệm về tính riêng tư và khai phá dữ liệu bảo vệ tính riêng tư.
- 8.2.** Trình bày thuật toán khai phá tập phổ biến bảo vệ tính riêng tư.
- 8.3.** Trình bày và phân tích thuật toán phân lớp cây quyết định bảo vệ tính riêng tư.

Chương 9.

TẬP MỜ, TẬP THÔ VÀ TẬP MỜ – THÔ TRONG KHAI PHÁ DỮ LIỆU

Các chương 4-7 đã trình bày các phương pháp chung nhất đối với các bài toán khai phá dữ liệu. Chương này tập trung vào một số khía cạnh chuyên sâu hơn về việc áp dụng lý thuyết tập mờ, lý thuyết tập thô vào các phương pháp khai phá dữ liệu để nâng cao hiệu quả của thuật toán khai phá dữ liệu. Các phương pháp và thuật toán được giới thiệu ở chương này chỉ tập trung vào các nội dung tới các thuật toán khai phá dữ liệu cơ bản được trình bày trong các chương 4-7.

Lý thuyết tập mờ (fuzzy set) và lý thuyết tập thô (rough set) hướng tới lập luận khả năng, suy diễn không đầy đủ và vì vậy các lý thuyết này tạo nền tảng lý thuyết tốt cho một lớp các phương pháp khai phá dữ liệu và phát hiện tri thức trong dữ liệu.

Hai mục đầu tiên của chương này trình bày một số phương pháp tập mờ và tập thô trong khai phá dữ liệu. Một lớp các phương pháp kết hợp tập mờ và tập thô (tập mờ-thô: fuzzy-rough set hoặc tập thô-mờ: rough-fuzzy set) trong khai phá dữ liệu được giới thiệu trong mục 9.3. Mỗi mục của chương đều được bắt đầu bằng việc giới thiệu một số kiến thức lý thuyết cơ bản nhất, và sau đó, một số phương pháp khai phá dữ liệu dựa trên từng lý thuyết được trình bày.

9.1. PHƯƠNG PHÁP TẬP MỜ TRONG KHAI PHÁ DỮ LIỆU

Lý thuyết tập mờ [Zadeh65] và lý thuyết khả năng [Zadeh78] ngày càng được ứng dụng rộng rãi trong nhiều lĩnh vực của khoa

học máy tính và toán học. Một lớp ứng dụng điển hình của lý thuyết tập mờ là các hệ thống tự động giám sát và điều khiển bao chứa các thao tác dựa trên lập luận.

Ngay thời kỳ phát triển đầu tiên của khai phá dữ liệu, các phương pháp dựa trên lý thuyết tập mờ đã được đề cập, trong đó lớp phương pháp khai phá luật là một trong những lớp điển hình nhất. Theo đánh giá của Eyke Hullermeier [Hyl11] thì các phương pháp mờ khai phá dữ liệu đặc biệt hữu dụng để trình diễn các mẫu "thô sơ" (vague pattern), điểm quan trọng mẫu chốt trong rất nhiều lĩnh vực ứng dụng. Đồng thời, các phương pháp mờ cũng đặc biệt hữu ích trong tiền xử lý dữ liệu (preprocessing) và hậu xử lý kết quả (postprocessing).

Trước khi đi tới một số bài toán khai phá dữ liệu điển hình dựa trên lý thuyết tập mờ, chúng ta xem xét các nội dung cơ bản nhất của lý thuyết này.

9.1.1. Một số kiến thức cơ sở của lý thuyết tập mờ

9.1.1.1. Một số khái niệm cơ bản

Ký hiệu X là không gian các đối tượng (điểm, dữ liệu) đang được quan tâm, ký hiệu x là phần tử tổng quát của X . Như vậy, $X = \{x\}$.

Định nghĩa 9.1.1 [Zadeh65]. Một tập mờ A trong X được biểu diễn bằng một hàm thành viên (hay hàm đặc trưng) $f_A(x)$ tương ứng mỗi đối tượng $x \in X$ với một giá trị $f_A(x)$ thuộc đoạn $[0,1]$, trong đó $f_A(x)$ biểu diễn “mức độ thành viên” của x trong A .

Như vậy, độ thành viên (membership, cũng gọi độ thuộc) của các phần tử $x \in X$ tới một tập mờ A có giá trị thuộc một miền liên tục từ 0 tới 1. Giá trị $f_A(x)$ càng gần 1 thì mức độ thành viên của x trong A càng cao. Khi A là một tập thông thường thì $f_A(x)$ chỉ nhận một trong hai giá trị 0 và 1, với $f_A(x) = 1$ ($f_A(x) = 0$) tương ứng với tình huống đối tượng x thuộc (không thuộc) tập A . Như vậy, trong trường hợp này, hàm thành viên $f_A(x)$ rút gọn thành “hàm đặc trưng” của tập A (thường được ký hiệu là λA trong lý

thuyết tập hợp). Tập thông thường $Y \subseteq X$ còn được gọi là **tập đơn giản** (hay **tập rõ**: crisp set). Nói chung đối với tập mờ, không có quan hệ một phần tử x "thuộc" một tập mờ A mà chỉ có thể có quan hệ "phần tử x thuộc tập mờ A với mức độ $f_A(x)$ ". Nói một cách chặt chẽ, tập mờ A không phải là một tập hợp theo nghĩa thông thường.

Ví dụ, cho X là tập các số nguyên là tuổi của nhân viên trong công ty ($X = [18, 60]$: theo quy định của luật lao động). Trong công ty không có quy định về người già, song trong bài toán khai phá luật kết hợp với thuộc tính tuổi, chúng ta đưa vào một tập mờ A "các nhân viên già" trong công ty. Khi đó, $f_A(x) = 0 \forall x \leq 30$, $f_A(x) = 1 \forall x \geq 55$ và $f_A(x)$ đơn điệu không giảm trong khoảng x từ 30 tới 55 và nhận giá trị từ 0 tới 1, chẳng hạn, $f_A(x) = ((x-30)/25)$: $\forall x \in [30, 55]$.

Tập mờ A được gọi là **rỗng** nếu $f_A(x) = 0 \forall x \in X$.

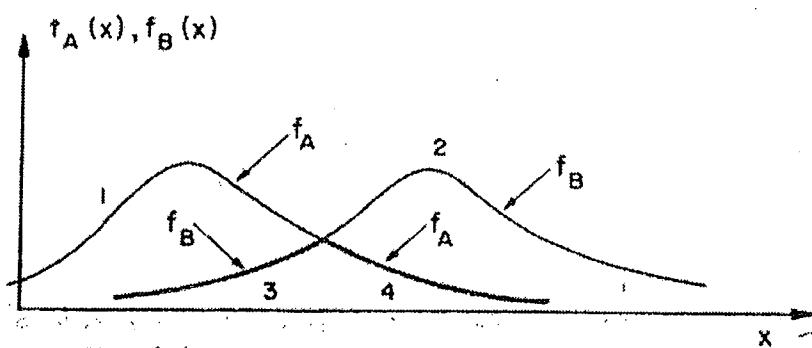
Tập mờ cũng có các phép toán tương tự như tập thông thường, bao gồm các phép toán quan hệ và các phép toán đại số.

Hai tập mờ A và B được gọi là **bằng nhau**, ký hiệu $A = B$, khi và chỉ khi $f_A(x) = f_B(x) \forall x \in X$ (ta viết $f_A = f_B$ thay cho $f_A(x) = f_B(x) \forall x \in X$; tương tự viết $f_A \leq f_B$ thay cho $f_A(x) \leq f_B(x) \forall x \in X$; và mở rộng cho mọi dấu phép toán so sánh khác).

Phần bù của tập mờ A , ký hiệu là A' , là tập mờ có hàm thành viên $f_{A'} = 1 - f_A$.

Tập mờ A bị chứa trong tập mờ B (A "là tập con" của B hoặc A "nhỏ hơn" B), ký hiệu $A \subset B$, khi và chỉ khi $f_A \leq f_B$: $A \subset B \Leftrightarrow f_A \leq f_B$. Tương ứng, ta định nghĩa được mọi quan hệ giữa hai tập mờ và mở rộng cho nhiều tập mờ như "chứa" ("lớn hơn"), "chứa thực sự" (lớn hơn thực sự"), "nhỏ nhất", "lớn nhất"...

Hợp của hai tập mờ A (với hàm thành viên $f_A(x)$) và B (với hàm thành viên $f_B(x)$) là một tập mờ C , ký hiệu $C = A \cup B$, với hàm thành viên $f_C(x)$ có giá trị là $f_C(x) = \max [f_A(x), f_B(x)] \forall x \in X$: $f_C = f_A \vee f_B$. Phép hợp hai tập mờ \cup có tính kết hợp: $A \cup (B \cup C) = (A \cup B) \cup C$. Có thể chứng minh được khẳng định "hợp giữa hai tập mờ là tập mờ nhỏ nhất chứa cả hai tập mờ đó".



Hình 9.1. Hai tập mờ A, B trên tập số thực và hợp (1+2), giao (3+4) của chúng [Zadeh65]

Giao của hai tập mờ A (với hàm thành viên $f_A(x)$) và B (với hàm thành viên $f_B(x)$) là tập mờ C, ký hiệu $C = A \cap B$, với hàm thành viên $f_C(x)$ có giá trị là $f_C(x) = \min [f_A(x), f_B(x)] \quad \forall x \in X: f_C = f_A \wedge f_B$. Phép giao hai tập mờ \cap có tính kết hợp: $A \cap (B \cap C) = (A \cap B) \cap C$. Tương tự như trên, cũng chứng minh được khẳng định “giao giữa hai tập mờ là tập mờ lớn nhất bị chứa bởi cả hai tập mờ đó”.

Tập mờ cũng đáp ứng Luật De Morgan như tập thông thường.

Hình 9.1 biểu diễn hai tập mờ A và B trên tập các số thực với các hàm thành viên f_A và f_B tương ứng. Hàm thành viên $f_{A \cup B}$ là đường cong nối hai nửa đường cong 1 và 2, hàm thành viên $f_{A \cap B}$ là đường cong nối hai nửa đường cong 3 và 4.

Hợp và giao là hai phép toán có vai trò đặc biệt quan trọng trong lý thuyết tập mờ vì chúng là nền tảng cho việc kết hợp các tập mờ. Chính vì lý do đó, nhằm tạo nên cơ chế linh hoạt tích hợp các tập mờ, người ta đã tổng quát hóa công thức tính hàm thành viên của hợp (giao) hai tập mờ thông qua hai chuẩn s-norm và t-norm.

Hàm $f: [0,1] \times [0,1] \rightarrow [0,1]$ được gọi là chuẩn s-norm (*t-norm*) nếu thỏa mãn bốn tính chất:

- Tính giao hoán: $f(x,y) = f(y,x), \forall x,y \in [0,1]$;
 - Tính đơn điệu: nếu $x \leq y$ thì $f(x,z) \leq f(y,z), \forall z \in [0,1]$;
 - Kết hợp: $f(x,f(y,z)) = f(f(x,y),z), \forall x,y,z \in [0,1]$;
 - Trung tính với 0: $f(x,0) = x, \forall x \in [0,1]$
- (Trung tính với 1: $f(x,1) = x, \forall x \in [0,1]$).

Hàm $\max(x,y)$ là một hàm chuẩn s-norm, hàm $\min(x,y)$ là một hàm chuẩn t-norm. Nối một cách tổng quát, hợp (giao) của hai tập mờ A và B là tập mờ có hàm thành viên $f_{A \cup B}$ ($f_{A \cap B}$) đáp ứng chuẩn s-norm (t-norm) của hai hàm thành viên f_A và f_B . Chuẩn s-norm còn được gọi là chuẩn t-conorm.

Các phép toán đại số trên tập mờ được xác định tương ứng trên hàm thành viên, chẳng hạn, tích đại số AB ($f_{AB} = f_A f_B$), tổng đại số A+B ($f_{A+B} = f_A + f_B$), hiệu tuyệt đối $|A-B|$ ($f_{|A-B|} = |f_A - f_B|$).

Một quan hệ mờ trên tập X là một tập mờ trên $X \times X$, có nghĩa là quan hệ mờ R trên X tương ứng với hàm thành viên f_R trên tập $X \times X$.

9.1.1.2. Tập mờ các mức

Mở rộng khái niệm tập mờ theo định nghĩa 9.1.1, Zadeh [Zadeh75] đưa ra khái niệm tập mờ theo các loại (mức) trong đó tập mờ nói trong định nghĩa 9.1.1 được gọi là tập mờ loại 1. Khái niệm tập mờ loại cao được L.A. Zadeh giới thiệu trong một bộ ba tài liệu của ông về biến ngôn ngữ (linguistic variable) và ứng dụng.

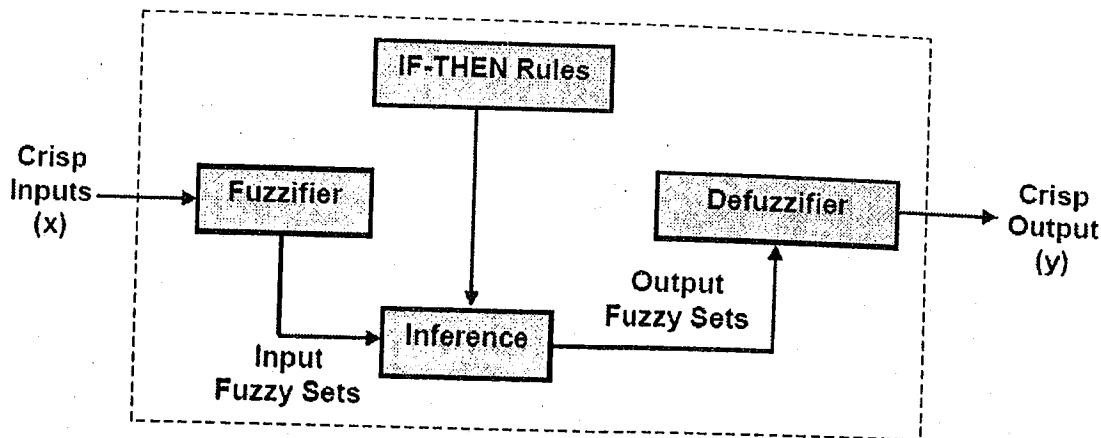
Định nghĩa 9.1.2 [Zadeh75]. Một tập mờ được gọi là loại n (type-n fuzzy set) nếu hàm thành viên của nó xác định trên các tập mờ loại n-1. Hàm thành viên của tập mờ loại 1 xác định trên các giá trị thuộc đoạn $[0, 1]$.

Tồn tại hai góc độ tiếp cận tới khái niệm tập mờ loại cao. Theo góc độ công nghệ thông tin (thuần túy), tập mờ loại 1 (tập mờ truyền thống) đã có tính trừu tượng cao và phải trải qua một quá trình lâu dài, tập mờ loại 1 mới được ứng dụng rộng rãi như ngày nay. Tập mờ loại cao có độ trừu tượng cao hơn về chất so với tập mờ loại 1 và làm tăng đáng kể độ phức tạp để hiểu và thi hành ứng dụng. Thực tiễn gần 40 năm ra đời, tập mờ mức loại 2 chưa đạt được tốc độ ứng dụng như tập mờ loại 1.

Theo góc độ công nghệ tri thức (Chương 2) thì hình thức phổ biến nhất của tri thức hiện trong chu kỳ sống là tri thức dưới dạng văn bản với các khái niệm là những biến ngôn ngữ. Một thành phần cơ bản của công nghệ tri thức là thu nhận và tích hợp thông tin. Tri thức chuyên gia miền lĩnh vực là một nguồn tri thức quan trọng trong thu nhận và tích hợp tri thức. Tuy nhiên, các chuyên gia miền lĩnh vực thường phát biểu các nhận định theo kinh nghiệm (tri thức miền ứng dụng của bài toán) của mình bằng lời ở các mức trừu tượng khác nhau. Tương ứng với các mức độ trừu tượng đó chính là các tập mờ các loại khác nhau. Khái niệm tập mờ đa loại cung cấp một cách diễn đạt về một quá trình làm mịn dần mức độ trừu tượng của các đối tượng biểu thị tri thức từ biến ngôn ngữ có tính trừu tượng cao nhất (tập mờ ở mức cao tương ứng) tới các tập mờ ở các mức trừu tượng thấp hơn và cuối cùng là tập mờ loại 1 được biểu diễn bằng các giá trị rõ mà các giá trị này có thể được xử lý tự động bằng các công cụ tính toán.

9.1.1.3. Hệ logic mờ, tập luật mờ, mờ hóa và khử mờ

Hình 9.2 trình bày sơ đồ cấu trúc thành phần điển hình của một hệ thống logic mờ (fuzzy logic system), một loại hệ thống ứng dụng tập mờ rất phổ biến. Hệ thống logic mờ nhận một giá trị đầu vào rõ x , mờ hóa giá trị này để chuyển thành một giá trị đầu vào mờ, lựa chọn các luật mờ tương ứng với giá trị mờ đầu vào mờ này để đưa ra một giá trị đầu ra mờ và cuối cùng khử mờ cho giá trị đầu ra mờ để có được một giá trị đầu ra rõ. Chẳng hạn, xem xét luật IF nhiệt-do-phong is "CAO-HON" THEN thao-tac is "HA-NHIET" trong bộ điều khiển của một điều hòa nhiệt độ giả định. Khi bộ đo nhiệt độ phòng hiện thời cho một giá trị cụ thể (35°C) là cao hơn so với nhiệt độ phòng cần giữ (28°C) thì động cơ sẽ chọn một giá trị rõ cho mức độ thao tác hạ nhiệt độ. Ở đây sử dụng cách ký hiệu luật mờ theo dạng IF – THEN để biểu thị rõ ngữ nghĩa thay cho cách viết thông dụng của luật kết hợp như (nhiet-do-phong is "CAO-HON") → (thao-tac is "HA-NHIET").



Hình 9.2. Hệ thống lôgic mờ (loại 1) [Chen07]

Hình 9.2 cho thấy tập các luật mờ (diễn hình là luật mờ dạng IF-THEN) là một thành phần cơ bản trong các hệ thống lôgic mờ.

Dạng đơn giản nhất của luật mờ là

IF *a* is A then *b* is B

trong đó, *a* và *b* là hai biến mờ (tập mờ), còn A và B là các giá trị mờ tương ứng với hai biến mờ *a* và *b*. Thêm một ví dụ khác cho luật mờ:

IF *sinh-vien* is "GIOI" THEN *diem-hoc-tap* is "CAO"

là một luật mờ, trong đó có "GIOI" là một giá trị mờ của biến mờ *sinh-vien* còn "CAO" là một giá trị mờ của biến mờ *diem-hoc-tap*. Các giá trị mờ ở đây chính là các tập mờ trên miền giá trị tương ứng (GIOI là một tập mờ trên tập các sinh viên, còn CAO là một tập mờ trên tập điểm học tập).

Tập luật mờ nói trên là kết quả của việc tích hợp tri thức có được của các chuyên gia miền ứng dụng, của các hệ thống khai phá tri thức từ dữ liệu. Một cách rất tự nhiên như đã được đề cập, chuyên gia miền ứng dụng thường mô tả tri thức của mình dưới dạng luật mờ. Để thích hợp với các hệ thống lôgic mờ, các hệ thống khai phá dữ liệu cũng cần cho các tri thức kết quả dưới dạng các luật mờ. Đây là một tình huống điển hình của khai phá dữ liệu dựa trên tập mờ.

Hình 9.2 cũng cho thấy hệ thống lôgic mờ có hai thành phần chức năng mà hoạt động của chúng có tính chất đối xứng nhau là thành phần “mờ hóa” (Fuzzifier) và thành phần “khử mờ” (Defuzzifier). Sự cộng tác của các chuyên gia miền ứng dụng có vai trò rất quan trọng trong việc thiết kế hai thành phần chức năng này.

Chẳng hạn, với thuộc tính tuổi của nhân viên của công ty trong bài toán khai phá luật kết hợp, công việc mờ hóa là xây dựng các tập mờ “già”, “trẻ”, “trung niên” trên tập X gồm các số nguyên [18,60]. Với sự cộng tác của các chuyên gia miền ứng dụng, hàm ánh xạ một giá trị rõ từ 18 tới 60 tới ba tập mờ “già”, “trẻ”, “trung niên” được xây dựng. Sau khi áp dụng luật mờ, nhận được một giá trị mờ đầu ra. Khi đó, cần khử mờ để có một giá trị rõ. Tồn tại nhiều phương pháp khử mờ, trong đó xây dựng các nhát cắt mờ thuộc dạng đơn giản nhất.

Cho một tập mờ A trên tập X và một số $\alpha \geq 0$. Nhát cắt α của tập mờ A là một tập thông thường $A_\alpha = \{x \in X : f_A(x) \geq \alpha\}$. Với mọi tập mờ A thì $A_0 = X$ và $A_\alpha = \emptyset \forall \alpha > 1$.

Với ví dụ về tập mờ “già” ở trên, nếu chọn ngưỡng $\alpha=0.8$ thì giá trị thuộc tính “già” của mọi nhân viên là 1 nếu nhân viên đó từ 50 tuổi trở lên và là 0 trong mọi trường hợp ngược lại.

9.1.2. Phương pháp tập mờ trong khai phá dữ liệu

Như đã đề cập, trong cuộc sống đời thường, để chỉ dẫn một tính chất, con người thường sử dụng một khái niệm hơn là một tập giá trị cụ thể biểu thị tính chất đó, chẳng hạn, khái niệm “già” thường được sử dụng nhiều hơn so với việc sử dụng tập các giá trị nguyên thuộc đoạn [50,60] do khái niệm “già” mang nhiều ngữ nghĩa hơn. Tương ứng, luật “IF thâm-niên NHIỀU, học-vị CAO THEN lương CAO” sẽ dễ hiểu hơn so với luật kết hợp “IF năm-dạy ≥ 30 , học-vi = TS OR học-vi = TSKH OR chức-danh = PGS OR chức-danh = GS THEN hệ-số-lương ≥ 6 ”. Trong trường hợp đầu tiên, tri thức phát hiện được có chứa giá trị ngôn ngữ NHIỀU, CAO, CAO.

Mặt khác, sử dụng tập mờ còn cho phép rút gọn tập giá trị thuộc tính hoặc để áp dụng được các thuật toán khai phá dữ liệu

đã biết trên miền hai giá trị (nhị phân) tới miền giá trị thực hoặc phân loại.

Lý thuyết tập mờ được áp dụng trong mọi giai đoạn của quá trình khai phá dữ liệu [Hul11], nổi bật nhất là áp dụng lý thuyết mờ trong giai đoạn chuẩn bị dữ liệu.

Các phương pháp tập mờ khai phá dữ liệu được sử dụng trong giai đoạn chuẩn bị dữ liệu là thuộc nhóm các phương pháp phân tích dữ liệu mờ (fuzzy data analysis). Các phương pháp này được phân chia thành hai nhóm theo hai hướng tiếp cận khác biệt rõ ràng. Hướng tiếp cận thứ nhất tiến hành việc mờ hóa chỉ trên các ánh xạ từ dữ liệu tới mô hình. Luật "IF thâm-niên NHIỀU, học-vị CAO THEN lương CAO" là kết quả của việc phân tích dữ liệu theo hướng tiếp cận này. Đây là hướng tiếp cận cho phép thi hành đơn giản hơn, vì vậy, các ứng dụng lý thuyết tập mờ được trình bày ở mục này là theo hướng tiếp cận này. Hướng tiếp cận thứ hai tiến hành việc nhúng toàn bộ dữ liệu vào các không gian mờ (thường là các không gian metric mờ) và thực hiện phân tích dữ liệu trên các không gian mờ này. Mặc dù hướng tiếp cận thứ hai là công phu và tinh vi hơn khi xây dựng các không gian metric mờ để nhúng dữ liệu vào song nó lại cho phép khai phá tốt các phương tiện của lý thuyết tập mờ vào bài toán khai phá dữ liệu. Lớp bài toán ứng dụng tập mờ khai phá dữ liệu điển hình theo hướng tiếp cận này là phân tích hồi quy mờ (Fuzzy regression analysis).

Các tình huống ứng dụng tập mờ vào khai phá dữ liệu còn cho thấy các phương pháp phân tích dữ liệu mờ đối với dữ liệu ban đầu có thể không được sử dụng trong giai đoạn chuẩn bị dữ liệu nhưng lại được sử dụng trong các giai đoạn sau đó. Chẳng hạn, thuật toán khai phá dữ liệu thi hành bước khai phá dữ liệu lại chứa các thành phần tích hợp là các giải pháp phân tích dữ liệu mờ.

9.1.2.1. Phương pháp tập mờ trong khai phá luật kết hợp

Lớp các phương pháp tập mờ khai phá luật kết hợp là một lớp phương pháp rộng lớn nhất. Chương 4 đã trình bày toàn diện

các nội dung về khai phá luật kết hợp, vì vậy, mục này chỉ tập trung vào lớp các phương pháp tập mờ khai phá luật kết hợp. Các phương pháp này được đi theo ba hướng tiếp cận điển hình: luật kết hợp với tập mờ trên miền giá trị của các mục dữ liệu (thường được phát biểu dưới dạng luật kết hợp mờ cho dữ liệu định lượng: Quantitative Data), luật kết hợp mờ với kiến trúc khái niệm (fuzzy taxonomy), luật kết hợp với tập mờ trên tập mục (thường được phát biểu dưới dạng bài toán khai phá luật kết hợp với giao dịch mờ: fuzzy transaction). Mục này sử dụng lại các khái niệm và ký hiệu đã được phát biểu tại Chương 4 nếu như không có sự chỉ dẫn khác.

Khai phá luật kết hợp mờ cho thuộc tính định lượng

Phát hiện luật kết hợp có xuất xứ từ các cơ sở dữ liệu giao dịch, trong đó miền giá trị của các thuộc tính chỉ có hai giá trị là 1 (mục đó xuất hiện trong giao dịch) và là 0 (trong trường hợp ngược lại). Trong trường hợp các thuộc tính có giá trị phân lớp hoặc có giá trị số thì khai phá luật kết hợp cần phải đi theo cách tiếp cận khác. Cần phải hình thành các tập mờ trên miền giá trị của các thuộc tính và luật kết hợp được biểu diễn theo các tập mờ đó.

Dạng đơn giản nhất luật kết hợp mờ trong trường hợp này là:

$$(X \text{ là } A) \rightarrow (Y \text{ là } B) \quad (9.1)$$

trong đó, X, Y là hai mục dữ liệu (thuộc tính) còn A, B là hai tập mờ tương ứng trên miền giá trị của hai thuộc tính này. Chẳng hạn, như luật mờ:

$$(thân-nhiệt \text{ là } CAO) \rightarrow (bệnh \text{ là } CAO)$$

Dạng (9.1) có thể được tổng quát hóa dưới dạng:

$$(X_1 \text{ là } A_1, X_2 \text{ là } A_2, \dots, X_p \text{ là } A_p) \rightarrow (Y \text{ là } B) \quad (9.2)$$

hoặc tổng quát hơn:

$$(X_1 \text{ là } A_1, \dots, X_p \text{ là } A_p) \rightarrow (Y_1 \text{ là } B_1, \dots, Y_q \text{ là } B_q) \quad (9.3)$$

trong đó, X_i , Y_j là các thuộc tính còn A_i , B_j là các tập mờ trên các miền giá trị của các thuộc tính tương ứng.

Dạng (9.2) được xem xét thông dụng hơn cả vì liên quan tới một thuộc tính quyết định, chẳng hạn

(*nhiệt-dộ* là CAO, *dộ-ẩm* là CAO) → (*số-diện* là NHIỀU)

Tuy nhiên, giá trị "CAO" của thuộc tính *nhiệt-dộ* hoặc *dộ-ẩm* hoặc giá trị "NHIỀU" của biến *số-diện* không có trong CSDL nguồn. Các giá trị này có được là kết quả của quá trình mờ hóa tương ứng với hình thành các tập mờ "CAO" hoặc "NHIỀU" trên miền giá trị của các thuộc tính tương ứng.

Lee J. H. và Hyung L. K. [LH97] đề xuất phương pháp áp dụng lý thuyết tập mờ vào khai phá luật kết hợp với thuộc tính nhận giá trị thực (được gọi là thuộc tính định lượng: quantitative attribute mà bao gồm các thuộc tính nhận giá trị phân lớp). Các tác giả đặt vấn đề tìm ra các luật có dạng (Hamburger, Medium) → (Coke, Small) trong đó Medium được coi là một tập mờ trên Hamburger còn Small được coi là một tập mờ trên Coke.

Phương pháp của Lee J. H. và Hyung L. K. hết sức đơn giản, bao gồm các bước chính yếu sau đây:

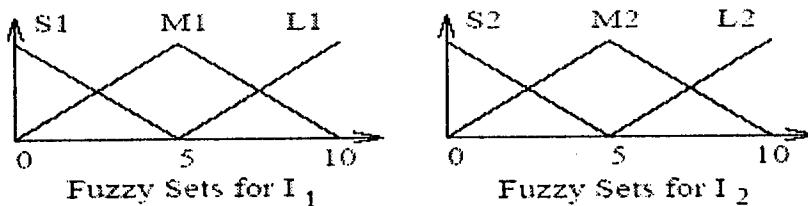
- Với mỗi thuộc tính định lượng, xác định một nhóm các tập mờ đủ để biểu thị ngữ nghĩa cần thiết cho bài toán khai phá luật (chẳng hạn, số lượng tiền mua Hamburger hay Coke có thể được tương ứng với nhóm tập mờ Large, Medium và Small). Với các tập mờ đã được xác định, xây dựng hàm chuyển giá trị định lượng thành giá trị hàm thành viên mờ.
- Chuyển CSDL gốc sang CSDL trung gian với tập thuộc tính là tập tất cả các tập mờ tương ứng với tất cả các thuộc tính gốc như được xác định ở trên. Bản ghi gốc được chuyển thành bản ghi của CSDL trung gian với các thành phần là các giá trị hàm thành viên mờ tương ứng với giá trị tại thuộc tính gốc.

Ví dụ, CSDL gốc gồm hai thuộc tính I_1 , I_2 và hai bản ghi $t_1 = (8, 7)$ và $t_2 = (3, 6)$. Xác định các tập mờ S_1 , M_1 , L_1 tương ứng với thuộc tính I_1 , các tập mờ S_2 , M_2 , L_2 tương ứng với thuộc tính I_2 (Hình 9.3).

Như vậy, CSDL trung gian gồm 2 bản ghi t'_1 và t'_2 với các thành phần như sau:

$$t_1' = (0, 0.4, 0.6, 0, 0.4, 0.6)$$

và $t_2' = (0.4, 0.6, 0, 0.8, 0.2)$.



Hình 9.3. Các tập mờ đối với hai thuộc tính [LH97]

- Một giá trị ngưỡng $\mu > 0$ được lựa chọn để chuyển các bản ghi thuộc CSDL trung gian sang bản ghi thuộc CSDL giao dịch. Ý nghĩa của ngưỡng μ là một giá trị thành phần của bản ghi trong CSDL trung gian được chuyển thành giá trị 1 hoặc 0 tùy thuộc vào giá trị đó có vượt qua giá trị ngưỡng μ hay không.
- Chuyển đổi CSDL trung gian về CSDL giao dịch: (i) tập thuộc tính (tập mục dữ liệu) vẫn giữ nguyên (chính là tập tất cả các tập mờ đã được xác định trên tập tất cả các tập thuộc tính định lượng ban đầu); (ii) các giá trị thành phần của các bản ghi được xác định là 1 hoặc 0 như đã nói ở trên.

Ví dụ, với giá trị ngưỡng $\mu = 0.3$ thì nhận được CSDL giao dịch có hai bản ghi:

$$t_1' = (0, 1, 1, 0, 1, 1)$$

và $t_2' = (1, 1, 0, 0, 1, 0)$.

- Thực hiện một phương pháp khai phá luật kết hợp cho CSDL giao dịch trên đây.
- Thu nhận kết quả và trực quan hóa.

Khai phá tập mục phổ biến theo CSDL giao dịch mờ

Khác với tình huống xem xét tập mờ trên miền giá trị của thuộc tính như trên đây, mục này không những xem xét bài toán khai phá luật kết hợp từ CSDL giao dịch mờ mà còn xem xét khai

phá luật kết hợp mờ với tập nền là tập tất cả các thuộc tính [DMSV03]. Miguel Delgado và cộng sự [DMSV03] đưa ra một số khái niệm liên quan phục vụ cho bài toán nói trên.

Định nghĩa 9.2.1. Cho I là tập tất cả các mục dữ liệu, giao dịch mờ là một tập mờ khác rỗng τ trên tập mục I . Với mọi $i \in I$, giá trị của giao dịch mờ tại thuộc tính i được ký hiệu là $\tau(i)$.

Ký hiệu tập FT là một tập các giao dịch mờ và ký hiệu $T = (I, FT)$ là một CSDL giao dịch mờ.

Miguel Delgado và cộng sự cũng đưa ra định nghĩa cho khái niệm luật kết hợp mờ, khái niệm và công thức xác định độ hỗ trợ và độ tin cậy của luật kết hợp mờ. Bài toán khai phá các tập mục phổ biến trong CSDL giao dịch mờ được phát biểu như sau:

Input: Cho một CSDL giao dịch mờ $D = (FT, I)$ trong đó FT là tập các giao dịch mờ, I là tập tất cả các mục dữ liệu. Cho độ hỗ trợ tối thiểu $minsup$ là một số dương.

Output: Tìm tất cả các tập mục phổ biến trong CSDL.

Thuật toán khai phá tập phổ biến trong CSDL giao dịch mờ có quy trình thực hiện tương tự như quy trình khai phá tập phổ biến với một số khác biệt về công thức tính toán độ hỗ trợ và độ tin cậy. Miguel Delgado và cộng sự trình bày chi tiết thuật toán khai phá luật kết hợp mờ.

9.1.2.2. Phương pháp tập mờ trong phân cụm dữ liệu

Thuật toán phân cụm mờ FCM (fuzzy c-means) do James C. Bezdek và cộng sự [BEF84] đề xuất là một thuật toán phân cụm mờ xuất hiện sớm nhất. Nội dung của thuật toán được mô tả tóm lược như dưới đây.

Cho trước một chuẩn $||x||$ trên không gian R^n . Thuật toán nhận đầu vào gồm: một tập dữ liệu $Y = \{y_i \in R^n, i = 1, N\}$, một hằng số số lượng cụm c (c dương, hữu hạn, $c > 1$), một số $m > 1$ là bậc trọng số, M_{fc} là tập tất cả các ma trận U cỡ $c \times N$, $U = \{u_{ij}\}: \sum_{k=1}^N u_{ik} > 0$,

$$\sum_{i=1}^c u_{ik} = 1; \text{ ngưỡng sai số } \varepsilon > 0 \text{ nhỏ.}$$

Thuật toán FCM

(A1) Cố định c , m , A , chuẩn $\| \cdot \|_A$. Chọn một ma trận khởi đầu $U \in M_{fc}$. Thực hiện lặp LMAX lần lượt, các bước sau đây.

(A2) Tính v_i theo công thức

$$v_i = \frac{\sum_{k=1}^N u_{ik}^m \times y_k}{\sum_{k=1}^N u_{ik}^m} \quad i = 1, 2, \dots, c$$

(A3) Tính toán lại giá trị của ma trận thành viên mới UN :

$$un_{ik} = \left(\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right)^{-1} \quad 1 \leq k \leq N; 1 \leq i \leq c$$

trong đó $d_{ik} = \| y_k - v_i \|_A$.

(A4) Nếu $\| U - UN \| < \epsilon$ thì dừng, ngược lại $U \leftarrow UN$ và quay lại (A2).

Cột i trong ma trận kết quả UN sẽ cho giá trị hàm thành viên của dữ liệu y_i tối c cụm.

9.1.2.3. Một số phương pháp tập mờ khai phá dữ liệu khác

Tập mờ còn được ứng dụng trong hầu hết các bài toán khai phá dữ liệu, chẳng hạn phân lớp (k-NN, SVM, cây quyết định), lựa chọn thuộc tính, dự báo vector hỗ trợ mờ...

Chẳng hạn, một mô tả thuật toán phân lớp k-NN mờ [JC11]:

FNN (U, C, y, K)

U : tập ví dụ học; C : tập các lớp;

y : đối tượng dữ liệu mới cần xác định nhãn;

K : số lượng láng giềng của mô hình.

Nội dung:

- (1) $N \leftarrow \text{getNearestNeighbours}(y, K)$; /tập N chứa k láng giềng gần y nhất.

(2) $\forall c \in C:$

$$(3) c'(y) = \sum_{x \in N} R(x,y) * c(x)$$

ở đây $R(x,y)$ là độ tương tự giữa x và y được tính theo công thức

$$R(x,y) = \frac{\|y - x\|^{-2/(m-1)}}{\sum_{j \in N} \|y - j\|^{-2/(m-1)}}$$

(4) output $\text{argmax}_{c \in C} (c'(y))$

9.2. PHƯƠNG PHÁP TẬP THÔ TRONG KHAI PHÁ DỮ LIỆU

Lý thuyết tập thô được N. Pawlack [Pawlack82] khởi xướng từ những năm đầu thập niên 1980 với đối tượng nghiên cứu là hệ thống thông tin và bảng quyết định. Xoay quanh hệ thống thông tin và bảng quyết định, bài toán được quan tâm trong lý thuyết tập thô là phát hiện ra các mẫu (các tri thức) từ dữ liệu có sẵn là một bài toán thuộc lĩnh vực phát hiện tri thức từ dữ liệu. Phát hiện mẫu và luật dựa trên tập thô đã trở thành một chủ đề được quan tâm ngay từ các hội nghị khoa học quốc tế đầu tiên về khai phá dữ liệu. Nền tảng toán học tốt của lý thuyết tập thô tạo nên một lợi thế tri thức miền ứng dụng cho khai phá dữ liệu. Tuy nhiên, nền tảng toán học tốt đó cũng đặt ra các yêu cầu về điều kiện đối với miền áp dụng lý thuyết tập thô trong khai phá dữ liệu.

Lý thuyết tập thô thu hút đông đảo các nhà nghiên cứu, điển hình là Cộng đồng nghiên cứu tập thô thế giới (the International Rough Set Society - IRSS, <http://roughsets.home.pl/www/>) được thành lập từ năm 2005 và Cộng đồng tập thô và tính toán mềm Trung Quốc (China Rough Set and Soft Computing Society: CRS&SCS, <http://cs.cqupt.edu.cn/crssc/>).

Hoạt động tập trung của IRSS ứng dụng lý thuyết tập thô trong khai phá dữ liệu là dãy các hội nghị quốc tế "Rough Sets and Knowledge Technology" diễn ra hàng năm là RSKT 2006 (Chongqing, China), RSKT 2007 (Toronto, Canada), RSKT 2008 (Chengdu, China), RSKT 2009 (Gold Coast, Australia), RSKT

2010 (Beijing, China), RSKT 2011 (Banff, Canada) và RSKT 2012 dự kiến sẽ được diễn ra tại Chengdu, China trong các ngày 17-20/8/2012.

Theo Marcin Szczuka [Szczu11], một số nội dung chính sau đây liên quan tới các phương pháp tập thô khai phá dữ liệu:

- Phương pháp tập thô dựa trên tính không phân biệt được giữa các đối tượng, thông qua việc so sánh giữa các phần tử. Một số chủ đề điển hình là tính phân biệt được, tính không phân biệt được, tính tương tự...
- Trong phát hiện tri thức, phương pháp tập thô có thể được áp dụng tới
 - (i) thực hiện các phương pháp tập thô trực tiếp để khai phá các mẫu hấp dẫn từ dữ liệu: rút gọn dữ liệu, luật quyết định tối thiểu, phân tách, học phân cấp...;
 - (ii) kết hợp và cải thiện hiệu quả các phương pháp hiện có trong khai phá dữ liệu như cây quyết định, luật kết hợp, phân cụm, kNN, các mạng thần kinh, mạng Bayesian...;
 - (iii) thiết kế dự án phát hiện tri thức thực sự trong một số lĩnh vực ứng dụng phức tạp như: máy tìm kiếm, phân tích dữ liệu y tế, tin sinh học,...

Mục con đầu tiên (9.2.1) trình bày một số kiến thức cơ bản về tập thô để làm nền tảng cho việc khảo sát các phương pháp tập thô trong khai phá dữ liệu. Chúng tôi chọn lựa các phương pháp tập thô rút gọn và rời rạc hóa thuộc tính trong hai mục con 9.2.2-9.2.3. Các phương pháp tập thô khai phá dữ liệu khác được trình bày trong nhiều công trình, trong đó có [SZ00, Szczu11].

9.2.1. Một số kiến thức cơ sở về lý thuyết tập thô

N. Pawlak hình thức hóa bảng dữ liệu bằng khái niệm *hệ thông tin* (information system) [Pawlak82].

9.2.1.1. Một số khái niệm cơ bản về tập thô

Định nghĩa 9.2.1 (Hệ thông tin: information system): Hệ thông tin là cặp $S = (U, A)$ trong đó U là một tập hữu hạn khác rỗng các *đối tượng*, A là một tập hữu hạn khác rỗng các *thuộc tính*; $\forall a \in A$: tồn tại tập giá trị V_a và ánh xạ $a: U \rightarrow V_a$. Gọi tập $V = \bigcup_{a \in A} V_a$ là tập giá trị của hệ thông tin.

Bảng 9.1. Một ví dụ về hệ thông tin

	Đau-đầu	Đau-cơ	Thân-nhiệt
u_1	Có	Có	Chuẩn
u_2	Có	Có	Cao
u_3	Có	Có	Rất-cao
u_4	Không	Có	Chuẩn
u_5	Có	Có	Cao
u_6	Không	Có	Rất-cao
u_7	Không	Không	Cao
u_8	Không	Có	Rất-cao

Hệ thông tin thường biểu diễn dưới dạng bảng hai chiều, mỗi hàng tương ứng với một đối tượng, mỗi cột tương ứng với một thuộc tính, giao của hàng i với cột j chính là giá trị $j(i)$ của ánh xạ j trên đối tượng i .

Ví dụ, cho hệ thông tin được mô tả ở bảng 9.1 gồm 6 đối tượng (mỗi đối tượng là một bệnh nhân cúm) và 3 thuộc tính: *Đau-đầu*, *Đau-cơ*, *Thân-nhiệt*.

Hai bệnh nhân khác nhau u_2 và u_5 nhận giá trị thuộc tính giống nhau: đây là trường hợp các đối tượng khác nhau song *không phân biệt được* nếu chỉ sử dụng thông tin từ ba thuộc tính Đau - đầu, Đau - cơ và Thân - nhiệt. Có thể nhận thấy sự mập

mờ từ việc không phân biệt được: nếu chỉ xem xét các thuộc tính trên đây thì hai bệnh nhân u_2 và u_5 là hoàn toàn giống nhau, tuy nhiên, bệnh nhân u_2 bị cúm còn bệnh nhân u_5 thì không bị cúm (xem bảng 9.2.2). Cặp bệnh nhân (u_6 và u_8) lại có điểm khác cặp trên đây khi mà chúng không chỉ giống nhau ở 3 triệu chứng mà còn giống nhau ở tình trạng cúm.

Do quan hệ tương đương được sử dụng ở hầu hết các ứng dụng lý thuyết tập thô trong khai phá dữ liệu, vì vậy chúng ta giới thiệu lại định nghĩa quan hệ tương đương.

Định nghĩa 9.2.2 (Equivalence relation) Cho U là tập các đối tượng, một quan hệ nhị phân $R \subseteq U \times U$ trên U được gọi là:

- *Phản xạ* nếu mọi đối tượng đều có quan hệ với chính nó xRx ,
- *Đối xứng* nếu xRy thì yRx ,
- *Bắc cầu* nếu xRy và yRz thì xRz .

Một quan hệ R thỏa mãn cả ba tính chất phản xạ, đối xứng và bắc cầu được gọi là một *quan hệ tương đương*. Quan hệ tương đương R chia (phân hoạch) tập tổng thể U thành các *lớp tương đương*. Lớp tương đương của phần tử $u \in U$, kí hiệu là $[u]$, chứa tất cả các đối tượng $v \in U$ mà uRv ; rõ ràng là $\forall u \neq v$: hoặc $[u] = [v]$ hoặc $[u] \cap [v] = \emptyset$.

Hệ thông tin $S = (U, A)$ sinh ra một quan hệ tương đương R_S (thường được ký hiệu là R_A để tương ứng với tập thuộc tính A) trên tập đối tượng U như sau:

$$\forall u, v \in U: u R_S v \Leftrightarrow \forall a \in A: a(u) = a(v)$$

Mở rộng, tương ứng với tập thuộc tính $B \subseteq A$, ta có quan hệ tương đương R_B như sau:

$$\forall u, v \in U: u R_B v \Leftrightarrow \forall a \in B: a(u) = a(v)$$

Từ quan hệ tương đương R_A (hay R_S), N. Pawlak đưa ra khái niệm tập sơ cấp và tập mô tả được.

Định nghĩa 9.2.3. (Tập sơ cấp: elementary set): Một lớp tương đương trên U theo quan hệ tương đương R_A được gọi là một tập sơ cấp.

Ta dùng ký hiệu E để chỉ tập tất cả các tập sơ cấp trên U . Trong lý thuyết về tính toán hạt (granular computing), tập sơ cấp được gọi là một hạt tương đương (equivalence granule) hoặc ngắn gọn hơn là một hạt. Hệ thông tin ở Bảng 9.2.1 có 6 tập sơ cấp (hạt) là $\{u_1\}$, $\{u_2, u_5\}$, $\{u_3\}$, $\{u_4\}$, $\{u_6, u_8\}$ và $\{u_7\}$.

Một ngôn ngữ L và một ánh xạ $\sigma: L \rightarrow 2^U$ được xây dựng để mô tả một lớp các tập con của tập U ; một tập con trong lớp này được gọi là tập mô tả được (described set) theo S [Pawlak82]. Tập mô tả được là tập kết quả của ánh xạ σ lên một biểu thức (term) trong ngôn ngữ L . N. Pawlak cũng chỉ ra tập X là tập mô tả được theo S khi và chỉ khi X là hợp của một số hữu hạn các tập sơ cấp.

Trong hệ thông tin ví dụ, tập $M = \{u_1, u_2, u_5, u_6\}$ là mô tả được vì là hợp của ba tập sơ cấp $\{u_1\}$, $\{u_2, u_5\}$, $\{u_7\}$. Biểu thức mô tả tập X là: $X = ((Đau-đầu = "Có") \wedge (Đau-cơ = "Có") \wedge (Thân-nhiệt = "Chuẩn")) \vee ((Đau-đầu = "Có") \wedge (Đau-cơ = "Có") \wedge (Thân-nhiệt = "Cao")) \vee ((Đau-đầu = "Có") \wedge (Đau-cơ = "Có") \wedge (Thân-nhiệt = "Rất-cao"))$. Dấu hiệu " \wedge " biểu thị phép toán "giao" và dấu hiệu " \vee " biểu thị phép toán "hợp". Tập $X = \{u_1, u_2, u_3\}$ không phải là một tập mô tả được vì nó không là hợp của một số tập sơ cấp hay không có biểu thức nào qua ánh xạ σ cho kết quả là X .

Như vậy, tập tất cả các tập con của U (2^U) được chia ra làm hai loại: tập mô tả được theo S và tập không mô tả được theo S . Ý tưởng thông qua khái niệm "mô tả được" để mô tả (xấp xỉ) các tập không mô tả được chính là xuất phát điểm của khái niệm tập thô. Khái niệm xấp xỉ trên và xấp xỉ dưới của một tập con $X \subseteq U$ được đưa ra vì mục đích này.

Định nghĩa 9.2.4 (Tập xấp xỉ / xấp xỉ tập: set approximations). Cho hệ thông tin $S = (U, A)$ và một tập $X \subseteq U$. Khi đó:

- Xấp xỉ dưới (lower approximation) của X (ký hiệu \underline{X}) là tập mô tả được có dạng $\underline{X} = \cup e: e \in E$ và $e \subseteq X$
- Xấp xỉ dưới của X là hợp của tất cả các tập sơ cấp được chứa trong X .

- Xấp xỉ trên (upper approximation) của X (ký hiệu \bar{X}) là tập mô tả được có dạng $\bar{X} = \cup e: e \in E \text{ và } e \cap X \neq \emptyset$.
- Xấp xỉ trên \bar{X} của X là hợp của tất cả các tập sơ cấp có giao khác trống với X .

Rõ ràng là với mọi tập X thì $\underline{X} \subseteq \bar{X}$. Tập hiệu $\bar{X} \setminus \underline{X}$ được gọi là tập biên (boundary set). Trong trường hợp X là tập mô tả được thì $\bar{X} = \underline{X} = X$ và tập biên rỗng. Khi X là tập không mô tả được, thì cặp tập mô tả được xấp xỉ (\bar{X}, \underline{X}) được sử dụng để "mô tả" (biểu diễn) X . Cặp mô tả được đó còn được gọi là tập thô tương ứng với X .

Với tập ví dụ $X = \{u_1, u_2, u_3\}$, ta có $\underline{X} = \{u_1, u_3\}$, $\bar{X} = \{u_1, u_2, u_3, u_5\}$.

Tương tự như khái niệm hàm đặc trưng trong tập thông thường, một hàm "thô" (rough function) f_X được xây dựng trên tập $X \subseteq U$ theo công thức:

$$\forall u \in U: f_X(u) = \frac{\| [u] \cap X \|}{\| [u] \|}, \text{ trong đó } [u] \text{ là lớp tương đương theo quan hệ } R_S \text{ chứa đối tượng } u.$$

Ta có, $\underline{X} = \{u \in U: f_X(u) = 1\}$ và $\bar{X} = \{u \in U: f_X(u) \geq 0\}$. Như vậy, theo một nghĩa nào đó, hàm thô đại diện cho cặp tập xấp xỉ (\bar{X}, \underline{X}) .

Với tập ví dụ $X = \{u_1, u_2, u_3\}$, ta có $f_X(u_1) = 1$, $f_X(u_2) = 1/2$, $f_X(u_3) = 1$, $f_X(u_5) = 1/2$, $f_X(u_4) = f_X(u_6) = f_X(u_7) = f_X(u_8) = 0$.

Định nghĩa 9.2.4 về tập xấp xỉ trong hệ thông tin $S = (U, A)$ được mở rộng tới các hệ thông tin thu hẹp S trên tập con thuộc tính của A . Giả sử B là tập con của A khi đó $S_B = (U, B)$ là hệ thông tin S sau khi bỏ đi các cột thuộc $A \setminus B$. Với một tập $X \subseteq U$, hệ thông tin S_B cũng cho cặp các tập xấp xỉ X , được ký hiệu là $(\bar{B}(X), \underline{B}(X))$. Như vậy, cặp (\bar{X}, \underline{X}) các tập xấp xỉ tập X trong hệ thông tin $S = (U, A)$ cũng được viết là $(\bar{A}(X), \underline{A}(X))$.

Một dạng đặc biệt của hệ thông tin là bảng quyết định, trong đó tập thuộc tính được chia thành hai nhóm: nhóm thuộc tính điều kiện và nhóm thuộc tính quyết định.

Bảng 9.2. Một ví dụ về bảng quyết định bệnh cúm

	<i>Đau-dầu</i>	<i>Đau-cơ</i>	<i>Thân-nhiệt</i>	<i>Cúm</i>
u_1	Có	Có	Chuẩn	Không
u_2	Có	Có	Cao	Có
u_3	Có	Có	Rất-cao	Có
u_4	Không	Có	Chuẩn	Không
u_5	Có	Có	Cao	Không
u_6	Không	Có	Rất-cao	Có
u_7	Không	Không	Cao	Không
u_8	Không	Có	Rất-cao	Có

Định nghĩa 9.2.5 (Bảng quyết định: decision table): Bảng quyết định là một hệ thông tin T có dạng $T = (U, C \cup D)$ trong đó C, D là tập các thuộc tính, $C \cap D = \emptyset$. Các thuộc tính thuộc C được gọi là *thuộc tính điều kiện* hay *điều kiện*, các thuộc tính thuộc D được gọi là *thuộc tính quyết định* hay *quyết định*.

Thuộc tính quyết định có thể có nhiều hơn hai giá trị, tuy nhiên, đơn giản là kiểu giá trị nhị phân, chẳng hạn $\{\text{Có}, \text{Không}\}$. Quá trình khám phá ra mối quan hệ giữa thuộc tính quyết định theo thuộc tính điều kiện trong bảng quyết định thuộc vào loại *học máy có giám sát*.

Bảng 9.2 cho một ví dụ về bảng quyết định. Hai bệnh nhân x_2 và x_5 đều có các giá trị giống nhau theo thuộc tính điều kiện (các triệu chứng), nhưng kết quả quyết định (tình trạng bệnh) đối với hai đối tượng này là khác nhau.

Trường hợp đặc biệt, bảng quyết định có dạng $T = (U, C \cup \{d\})$, trong đó $d \notin C$, và C là tập các *thuộc tính điều kiện* hay *điều kiện* còn d là *thuộc tính quyết định* hay *quyết định*.

9.2.1.2. Quan hệ không phân biệt được trong hệ thông tin

Một trong những cơ sở toán học của lý thuyết tập thô là quan hệ không phân biệt được trong hệ thông tin. Cho hệ thông tin $S = (U, A)$, quan hệ không phân biệt được được trình bày như dưới đây.

Định nghĩa 9.2.6 (Quan hệ không phân biệt được: Indiscernible relation): Với tập con bất kỳ $B \subseteq A$, tồn tại một quan hệ tương đương được gọi là quan hệ không phân biệt được theo B (kí hiệu là $\text{IND}_S(B)$) được xác định như sau:

$$\text{IND}_S(B) = \{(u, u') \in U^2 \mid \forall a \in B: a(u) = a(u')\}$$

Tính "không phân biệt được theo B " của $\text{IND}_S(B)$ được giải thích như sau: nếu như hai đối tượng u, u' mà $(u, u') \in \text{IND}_S(B)$ thì u và u' là không phân biệt được lẫn nhau bởi các thuộc tính trong B . Trong nhiều trường hợp khi hệ thông tin đã hoàn toàn xác định, dùng ký hiệu $\text{IND}(B)$ hay IND thay cho ký hiệu $\text{IND}_S(B)$.

Lớp tương đương của đối tượng u theo quan hệ không phân biệt được theo B được biểu diễn là $[u]_B$. Lớp tương đương $[x]_B$ (theo quan hệ $\text{IND}_S(B)$) cũng được gọi là hạt tương đương theo B (hoặc ngắn gọn là hạt theo B). Trong nhiều trường hợp, cặp $(u, |[u]_B|)$ được chọn làm đại diện cho hạt $|[u]_B|$.

9.2.2. Phương pháp tập thô rút gọn thuộc tính

Như đã được giới thiệu ở Chương 2, rút gọn dữ liệu là một bước thực hiện được tiến hành trong giai đoạn tiền xử lý dữ liệu. Rút gọn dữ liệu bao gồm rút gọn theo chiều dọc (rút gọn thuộc tính), rút gọn theo chiều ngang (rút gọn không gian đối tượng) và kết hợp cả hai. Như đã được giới thiệu, tính chất hạt của tập sơ cấp hoặc theo các lớp tương đương từ tập thuộc tính cho phép rút gọn dữ liệu theo chiều ngang. Mục này tập trung nghiên cứu phương pháp tập thô rút gọn thuộc tính.

Rút gọn dữ liệu không những làm giảm kích thước dữ liệu trong khai phá dữ liệu mà còn quan trọng hơn là cho một cách thức nâng cao chất lượng dữ liệu khi đưa ra một "đại diện" tốt hơn

cho dữ liệu miền ứng dụng. Rút gọn dữ liệu còn làm giảm bớt tình huống quá khớp dữ liệu (data overfit).

Rút gọn thuộc tính trong hệ thông tin

Rút gọn thuộc tính là công việc tìm ra một tập con các thuộc tính có thể thay thế tập tất cả các thuộc tính trong một miền ứng dụng. Rút gọn thuộc tính là một bài toán khai phá dữ liệu dựa trên tập thô quan trọng bậc nhất.

Định nghĩa 9.2.7 (Ma trận không phân biệt được: discernibility matrix): Cho hệ thông tin $S = (U, A)$. Ma trận $M = (m_{u,v})$ có kích thước $|U| \times |U|$ được gọi là ma trận không phân biệt được của S nếu các phần tử $m_{u,v}$ được xác định:

$$m_{u,v} = \{a \in A : u(a) \neq v(a)\}$$

Định nghĩa 9.2.8 (Tập rút gọn: reduct set): Cho hệ thông tin $S = (U, A)$. Tập thuộc tính $B \subseteq A$ được gọi là tập thuộc tính rút gọn của A nếu như thỏa mãn hai điều kiện:

- (i) Hai quan hệ tương đương R_A và R_B bằng nhau ($R_A = R_B$),
- (ii) Không tồn tại tập $B' \subseteq B$, $B' \neq B$ để $R_A = R_{B'}$.

Định nghĩa 9.2.9 (Tập rút gọn theo ma trận không phân biệt được: reduct set): Cho hệ thông tin $S = (U, A)$ và M là ma trận không phân biệt được của nó. Tập thuộc tính $B \subseteq A$ được gọi là tập thuộc tính rút gọn của A theo M nếu như thỏa mãn hai điều kiện:

- (i) $\forall u, v \in U : m_{u,v} \cap B \neq \emptyset$,
- (ii) $\forall a \in B, \exists u, v \in U : m_{u,v} \cap (B - \{a\}) = \emptyset$

Có thể thấy khái niệm tập rút gọn từ Định nghĩa 9.2.8 cho ý tưởng về rút gọn hệ thông tin theo chiều dọc, còn khái niệm hạt cho ý tưởng rút gọn hệ thông tin theo chiều ngang. Định nghĩa 9.2.9 cho ý tưởng về một giải pháp kiểm định một tập thuộc tính có là rút gọn hay không (tính tương đương ngữ nghĩa của hai khái niệm này đã được chứng minh).

Rút gọn thuộc tính trong bảng quyết định

Trong bảng quyết định, rút gọn thuộc tính được xem xét trên tập thuộc tính điều kiện. Bảng 9.2.3 nhận được từ bảng quyết định tại Bảng 9.2.2 sau khi bỏ đi hai phần tử u_5 , u_8 và đánh lại chỉ số các đối tượng được sử dụng làm ví dụ trong mục này.

Xem xét khái niệm "miền dương" của một tập thuộc tính điều kiện theo D (positive region of D); đây là một tập con các đối tượng. Trước hết, xét miền dương của tập thuộc tính điều kiện C:

$$POS_C(D) = \bigcup_{X \in U/D} CX \quad (9.1)$$

$\forall B \subseteq C$, ta cũng có miền dương của B:

$$POS_B(D) = \bigcup_{X \in U/D} BX \quad (9.2)$$

Khái niệm miền dương trong bảng quyết định như công thức (9.1), (9.2) được mở rộng vào hệ thông tin.

Bảng 9.3. Bảng quyết định bệnh cúm thu gọn

	Đau-dầu	Đau-cơ	Thân-nhiệt	Cúm
u_1	Có	Có	Chuẩn	Không
u_2	Có	Có	Cao	Có
u_3	Có	Có	Rất cao	Có
u_4	Không	Có	Chuẩn	Không
u_5	Không	Không	Cao	Không
u_6	Không	Có	Rất cao	Có

Định nghĩa 9.2.10 (Miền dương trong hệ thông tin): Cho hệ thông tin $S = (U, A)$, cho hai tập thuộc tính $P, Q \subseteq A$. Miền dương của tập P theo Q, ký hiệu là $POS_P(Q)$, là một tập con các đối tượng được xác định theo công thức:

$$POS_P(Q) = \bigcup_{X \in U/Q} P(X) \quad (9.3)$$

Định nghĩa 9.2.11 (Thuộc tính điều kiện có thể bỏ: dispensable attribute): Thuộc tính điều kiện $c \in C$ được gọi là thuộc tính có thể bỏ nếu như $POS_{C-\{c\}}(D) = POS_C(D)$ có nghĩa là miền dương theo D không bị thay đổi nếu bỏ đi thuộc tính c. Thuộc tính c không có tính chất nói trên được gọi là thuộc tính không bỏ được (indispensable attribute).

Định nghĩa 9.2.12 (Bảng quyết định không phụ thuộc: independent decision table): Bảng quyết định được gọi là không phụ thuộc nếu mọi thuộc tính điều kiện là không bỏ được.

Định nghĩa 9.2.13 (Tập thuộc tính rút gọn điều kiện: reduct of condition attributes): Tập thuộc tính điều kiện $R \subseteq C$ được gọi là rút gọn của C nếu như $T' = (U, R, D)$ là không phụ thuộc và $POS_R(D) = POS_C(D)$.

Bảng quyết định ở Bảng 9.3 có hai tập thuộc tính rút gọn là {Đau-dầu, Thân-nhiệt} và {Đau-cơ, Thân-nhiệt}. Với tập thuộc tính rút gọn {Đau-cơ, Thân-nhiệt} thì các cặp đối tượng (u_1, u_4) và (u_3, u_6) được gom nhóm lại.

Gọi CORE là tập tất cả các tập thuộc tính rút gọn R trong bảng quyết định T. Khi đó "lõi" (core) của bảng quyết định T là tập thuộc tính $CORE = \cap R$ (mọi R là rút gọn trong T).

Trong ví dụ trên, $CORE = \{\text{Thân-nhiệt}\}$.

Định nghĩa 9.2.14: Cho hệ thông tin $S = (U, A)$, cho hai tập thuộc tính $P, Q \subseteq A$, chúng ta nói độ phụ thuộc của Q theo P (ký hiệu $\gamma_P(Q)$) là một giá trị ($0 \leq k \leq 1$) được tính theo công thức

$$k = \gamma_P(Q) = \frac{|POS_P(Q)|}{|U|} \quad (9.4)$$

Một thuật toán tìm một tập rút gọn

Tồn tại một vài phiên bản của bài toán rút gọn thuộc tính mà hai bài toán điển hình nhất là: (i) Tìm một tập thuộc tính rút gọn; (ii) Tìm tất cả các thuộc tính rút gọn và tìm tập thuộc tính lõi.

Trong [JS09], Richard Jensen và Qiang Shen giới thiệu thuật toán QUICKReduct rất đơn giản để tìm một tập rút gọn của tập thuộc tính điều kiện. Thuật toán được trình bày như dưới đây.

Thuật toán QuickReduct:

Input: Bảng quyết định $T = (U, C \cup D)$, C là tập thuộc tính điều kiện, D là tập thuộc tính quyết định..

Output: Tập R là một tập thuộc tính điều kiện rút gọn.

Phương pháp:

- (1) $R \leftarrow \emptyset$ // ban đầu R rỗng
- (2) do
- (3) $T \leftarrow R$
- (4) foreach $x \in (C - R)$ // thử thêm x
- (5) if $\gamma_{R \cup \{x\}}(D) > \gamma_T(D)$
- (6) $T \leftarrow R \cup \{x\}$
- (7) $R \leftarrow T$
- (8) until $\gamma_R(D) == \gamma_C(D)$
- (9) return R.

Giải thuật 9.1. Thuật toán tìm một tập rút gọn [JC09]

Hoạt động của thuật toán rất đơn giản: bổ sung dần các thuộc tính cần thiết cho đến khi không còn thuật toán cần thiết nữa (dòng 8).

Thuật toán này chưa có khả năng tìm tập thuộc tính lõi của bảng quyết định/hệ thông tin. Ma trận không phân biệt được cho phép xây dựng các thuật toán tìm tập thuộc tính lõi [SZ00].

9.2.3. Phương pháp tập thô rời rạc tập giá trị thuộc tính

Một bài toán chuyển dạng dữ liệu quan trọng đó là rời rạc giá trị thuộc tính (Chương 2). Trong trường hợp miền giá trị thuộc tính có lực lượng rất lớn ($|V_a| >> 1$) thì công việc nói trên là rất cần thiết không chỉ làm giảm độ phức tạp tính toán mà còn cho phép "làm tròn" dữ liệu.

Theo Andrzej Skowron và Ning Zhong [SZ00], bài toán rời rạc hóa thuộc tính trong bảng quyết định được phát biểu như sau:

Cho bảng quyết định $T = (U, A \cup \{d\})$: U là tập đối tượng, A là tập thuộc tính điều kiện, d là thuộc tính quyết định, $d \notin A$. Mỗi thuộc tính điều kiện $a \in A$ có miền giá trị thực $V_a = [v_a, w_a]$. Một phân hoạch P_a của V_a là dãy các điểm cắt $P_a = v_1 < v_2 < \dots < v_k$ trong V_a . Một họ các phân hoạch $\{P_a\}_{a \in A}$ được gọi là tập các nhát cắt trong T . Bài toán rời rạc hóa dữ liệu là đi tìm một tập tối thiểu các nhát cắt trong T . Hình 9.4 cho một ví dụ về một nhát cắt ban đầu trong bảng quyết định [SZ00].

Bảng 9.4. Ví dụ chuyển bảng quyết định qua tập nhát cắt [SZ00]

U	a	b	d
x_1	0.8	2	1
x_2	1	0.5	0
x_3	1.3	3	0
x_4	1.4	1	1
x_5	1.4	2	0
x_6	1.6	3	1
x_7	1.3	1	1

\Rightarrow

U	a^P	b^P	d
x_1	0	2	1
x_2	1	0	0
x_3	1	2	0
x_4	1	1	1
x_5	1	2	0
x_6	2	2	1
x_7	1	1	1

Andrzej Skowron và Ning Zhong mô tả thuật toán tìm tập nhát cắt tối thiểu trong bảng quyết định gồm các bước sau đây:

- *Bước 0:* Xác định tập nhát cắt ban đầu gồm tất cả các giá trị của mỗi thuộc tính có trong bảng quyết định. Với bảng quyết định ban đầu phía trái Bảng 9.4, bước này cho kết quả là

$$P_a = 0.8 < 1 < 1.3 < 1.4 < 1.6$$

$$P_b = 0.5 < 1 < 2 < 3$$

- *Bước 1:* Định nghĩa các biến logic (Boolean variabe) trên tập đối tượng U từ nhát cắt ban đầu nói trên. Theo ví dụ trên, ta có:

$$BV(U) = \{P_1^a, P_2^a, P_3^a, P_4^a, P_1^b, P_2^b, P_3^b\} \text{ trong đó}$$

P_1^a , tương ứng với khoảng $[0.8, 1)$ của a ,

P_2^a , tương ứng với khoảng $[1, 1.3)$ của a ,

P_3^a , tương ứng với khoảng [1.3, 1.4) của a,

P_4^a , tương ứng với khoảng [1.4, 1.6) của a,

P_1^b , tương ứng với khoảng [0.5, 1) của b,

P_2^b , tương ứng với khoảng [1, 2) của b,

P_3^b , tương ứng với khoảng [2, 3) của b.

Bước 2. Chuyển bảng quyết định T thành bảng quyết định mới T' theo tập các biến lôgic xây dựng được ở bước 1.

$T = (U, A \cup \{d\}) \rightarrow T' = (U^*, \{P_i^a\} \cup \{d\})$ theo các biến mệnh đề P_i^a tương ứng với các khoảng $[v_i^a, v_{i+1}^a)$ với mọi nhát cắt và mọi thuộc tính a. Bảng 9.5 cho kết quả thực hiện bước 2 từ ví dụ đã cho.

Bảng 9.5. Bảng quyết định trung gian [SZ00]

U^*	p_1^a	p_2^a	p_3^a	p_4^a	p_1^b	p_2^b	p_3^b
(x1,x2)	1	0	0	0	1	1	0
(x1,x3)	1	1	0	0	0	0	1
(x1,x5)	1	1	1	0	0	0	0
(x4,x2)	0	1	1	0	1	0	0
(x4,x3)	0	0	1	0	0	1	1
(x4,x5)	0	0	0	0	0	1	0
(x6,x2)	0	1	1	1	1	1	1
(x6,x3)	0	0	1	1	0	0	0
(x6,x5)	0	0	0	1	0	0	1
(x7,x2)	0	1	0	0	1	0	0
(x7,x3)	0	0	0	0	0	1	0
(x7,x5)	0	0	1	0	0	1	0

Xây dựng ma trận phân biệt Ψ tương ứng với bảng quyết định T' , trong đó Ψ_{ij} là tập các P_i^a phân biệt hai đối tượng thứ i và thứ j của tập đối tượng U, Ψ_{ij} được viết dưới dạng chuẩn tuyển của các biến mệnh đề. Ví dụ,

$\Psi_{12} = P_1^a \vee P_1^b \vee P_2^b$ được hiểu là để tách được đối tượng x_1 và đối tượng x_2 cần ba nhát cắt: (i) một nhát cắt giữa a(0.8) và a(1), (ii) một nhát cắt giữa b(0.5) và b(1), (iii) một nhát cắt giữa b(1) và b(2).

Bước 3. Tìm tập con tối thiểu các nhát cắt mà được tách mọi đối tượng trong U thành các lớp khác nhau thông qua phép lấy chuẩn hội của các mệnh đề chuẩn tuyển Ψ_{12} :

$$\Phi^U = \wedge \{\Psi_{12}: d(x_i) \neq d(x_j)\}$$

Chuyển Φ^U về dạng chuẩn tuyển và số hạng ít mệnh đề trong chuẩn tuyển này là tập nhát cắt cần tìm. Kết quả thực hiện với ví dụ trên là $\Phi^U = (p_2^a \wedge p_4^a \wedge p_2^b) \vee (p_2^a \wedge p_3^a \wedge p_2^b \wedge p_3^b) \vee (p_3^a \wedge p_1^b \wedge p_2^b \wedge p_3^b) \vee (p_1^a \wedge p_4^a \wedge p_1^b \wedge p_2^b)$. Từ đó có $\{p_2^a, p_4^a, p_2^b\}$ là tập nhát cắt cần tìm. Bảng 9.6 trình bày bảng quyết định kết quả cuối cùng cho thấy số giá trị các thuộc tính giảm đi so với bảng quyết định ở phía phải Bảng 9.4.

Bảng 9.6. Bảng quyết định kết quả rời rạc hóa thuộc tính [SZ00]

U	a	b	d
x1	0.8	2	1
x2	1	0.5	0
x3	1.3	3	0
x4	1.4	1	1
x5	1.4	2	0
x6	1.6	3	1
x7	1.3	1	1

$P = \{(a, 1.2), (a, 1.5), (b, 1.5)\}$

U	a^P	b^P	d
x1	0	1	1
x2	0	0	0
x3	1	1	0
x4	1	0	1
x5	1	1	0
x6	2	1	1
x7	1	0	1

9.3. PHƯƠNG PHÁP TẬP MỜ-THÔ TRONG KHAI PHÁ DỮ LIỆU

Năm 1990, D. Dubois và H. Prade [DB90] đưa ra khái niệm tập mờ thô (fuzzy rough set hoặc fuzzy-rough set) như là sự tổng quát hóa mờ của tập thô và khái niệm tập thô mờ (rough fuzzy set hoặc rough-fuzzy set) như là một trường hợp đặc biệt của tập mờ thô. Nhiều tác giả sau đó đã phát triển lý thuyết tập mờ thô cũng như áp dụng lý thuyết tập mờ thô vào các bài toán khai phá dữ liệu.

Theo Richard Jensen [Jen11], có hai khuynh hướng trong khai phá dữ liệu dựa trên tập thô mờ: (i) tiếp cận hướng tiên đề: khảo sát các đặc trưng toán học từ lý thuyết tập mờ-thô có khả năng ứng dụng trong khai phá dữ liệu; (ii) tiếp cận hướng kiến thiết: khái quát (mờ) hóa xấp xỉ trên và xấp xỉ dưới. Thực tiễn cho thấy, hướng tiếp cận thứ hai đang thể hiện được thành công hơn.

Khai phá dữ liệu dựa trên tập mờ thô nhận được sự quan tâm của một số nhóm nghiên cứu, trong đó nhóm nghiên cứu của Richard Jensen và cộng sự (Aberystwyth University, Anh) là một trong những nhóm nghiên cứu có nhiều công trình nghiên cứu; một số nội dung trình bày dưới đây chủ yếu là từ các kết quả nghiên cứu của nhóm này. Hơn nữa, nhóm nghiên cứu của Richard Jensen và cộng sự đã tích hợp các thuật toán khai phá dữ liệu dựa trên tập mờ-thô²⁶ vào bộ công cụ khai phá dữ liệu WEKA²⁷. Các thuật toán được trình bày dưới đây dựa trên các kết quả nghiên cứu nói trên.

FRQUICKREDUCT(\mathbb{C}, \mathbb{D}).

\mathbb{C} , the set of all conditional attributes;

\mathbb{D} , the set of decision attributes.

- (1) $R \leftarrow \{\}; \gamma'_{best} = 0; \gamma'_{prev} = 0$
- (2) **do**
- (3) $T \leftarrow R$
- (4) $\gamma'_{prev} = \gamma'_{best}$
- (5) **foreach** $x \in (\mathbb{C} - R)$
- (6) **if** $\gamma'_{R \cup \{x\}}(\mathbb{D}) > \gamma'_T(\mathbb{D})$
- (7) $T \leftarrow R \cup \{x\}$
- (8) $\gamma'_{best} = \gamma'_T(\mathbb{D})$
- (9) $R \leftarrow T$
- (10) **until** $\gamma'_{best} == \gamma'_{prev}$
- (11) **return** R

Giải thuật 9.2. Thuật toán tập mờ-thô lựa chọn thuộc tính [JS09]

9.3.1. Lựa chọn thuộc tính dựa trên tập mờ - thô

Richard Jensen và Qiang Shen [JS09] đề xuất một thuật toán rút gọn thuộc tính trong bảng quyết định với tập thuộc tính điều kiện C và tập thuộc tính quyết định D.

Đầu vào: Bảng quyết định

²⁶ <https://www.hse.ru/data/2011/06/25/1215300216/Weka.pdf>

²⁷ <http://users.aber.ac.uk/rkj/book/wekafull.jar>

Đầu ra: Tập thuộc tính điều kiện R thu gọn tập thuộc tính điều kiện C.

Thuật toán hoạt động theo tiếp cận từ dưới lên. Độ phụ thuộc mờ trong tính toán các hàm γ cho phép lựa chọn thuộc tính tiếp theo có khả năng tốt nhất (Hình 9.6).

9.3.2. Phân lớp k-NN dựa trên tập mờ - thô

Richard Jensen và Chris Cornelis [JC11] trình bày thuật toán phân lớp k-NN dựa trên lý thuyết tập mờ thô như sau:

FRNN($\mathbb{U}, \mathcal{C}, y$).

\mathbb{U} , the training data; \mathcal{C} , the set of decision classes;
 y , the object to be classified.

- (1) $N \leftarrow \text{getNearestNeighbors}(y, K)$
- (2) $\mu_1(y) \leftarrow 0, \mu_2(y) \leftarrow 0, \text{Class} \leftarrow \emptyset$
- (3) $\forall C \in \mathcal{C}$
- (4) if $((R \downarrow C)(y) \geq \mu_1(y) \ \&\& \ (R \uparrow C)(y) \geq \mu_2(y))$
- (5) $\text{Class} \leftarrow C$
- (6) $\mu_1(y) \leftarrow (R \downarrow C)(y), \mu_2(y) \leftarrow (R \uparrow C)(y)$
- (7) output Class

Trong thuật toán này, $R \uparrow C(y)$ và $R \downarrow C(y)$ được tính theo công thức $(R \downarrow C)(x) = \inf_{y \in X} \mu((R(x,y), A(y)))$ và $(R \uparrow C)(x) = \sup_{y \in X} \tau(R(x,y), A(y))$ với μ là một t-conorm, τ là một t-norm, $R(x,y)$ là một quan hệ thứ lõi mờ và $A(y)$ là giá trị tập mờ A tại điểm y.

CÂU HỎI VÀ BÀI TẬP

- 9.1. Hãy chỉ ra ít nhất 5 hàm t-norm và 5 hàm s-norm.
- 9.2. Phân tích ý nghĩa thực tiễn của định nghĩa tập mờ loại 2 và các loại cao hơn.
- 9.3. Trình bày thuật toán chuyển CSDL có giá trị định lượng sang CSDL giao dịch

- 9.4. Trình bày thuật toán khai phá luật kết hợp mờ.
- 9.5. Trình bày và phân tích nội dung thuật toán FCM
- 9.6. Cho hệ thông tin $S = (U, A)$ và $X \subseteq U$. Cặp $(\underline{X}, \overline{X})$ là hai tập mô tả được xấp xỉ X và f_X là hàm thô tương ứng với X . Hãy chứng minh: $\underline{X} = \{u \in U : f_X(u) = 1\}$ và $\overline{X} = \{u \in U : f_X(u) \geq 0\}$.
- 9.7. Cho hệ thông tin $S = (U, A)$ và $B \subseteq A$. Hãy chỉ ra các mối quan hệ có thể có giữa hạt theo B và hạt theo A .
- 9.8. Trình bày và phân tích tính đúng đắn của thuật toán tập thô rút gọn thuộc tính.
- 9.9. Trình bày và phân tích tính đúng đắn của thuật toán tập thô rời rạc hóa giá trị thuộc tính.
- 9.10. Trình bày và phân tích về ý nghĩa của thuật toán rút gọn thuộc tính sử dụng tập mờ-thô.

Chương 10.

MỘT SỐ BÀI HỌC VÀ KHUYNH HƯỚNG PHÁT TRIỂN TRONG KHAI PHÁ DỮ LIỆU

Hình thành vào cuối thập niên 1980, lĩnh vực Khai phá dữ liệu (Phát hiện tri thức trong CSDL) phát triển mạnh mẽ và đã trở thành một lĩnh vực nghiên cứu và triển khai quan trọng trong việc khẳng định vai trò chiến lược của CNTT. Ngày nay, sử dụng tri thức đã trở thành động lực chủ yếu cho tăng trưởng kinh tế và khai phá dữ liệu tham gia vào quá trình chuyển hóa tri thức YDYN thành tri thức YNYN trong phạm vi tổ chức cũng như trong phạm vi quốc gia. Sự phát triển mạnh mẽ của lĩnh vực này vừa là nguyên nhân vừa là kết quả của sự quan tâm ngày càng sâu sắc và rộng lớn của cộng đồng nghiên cứu. Thuật ngữ *data mining* và hai thuật ngữ liên quan *machine learning* và *statistics* được ghi nhận vào danh sách 100 thuật ngữ khoa học nổi bật nhất theo ResearcherID²⁸, trong đó thuật ngữ *machine learning* ở tốp 11 thuật ngữ nổi bật nhất còn hai thuật ngữ *data mining* và *statistics* ở tốp 39 thuật ngữ nổi bật nhất; ngoài ra, thuật ngữ *knowledge management* cũng có mặt trong danh sách 100 thuật ngữ nói trên.

Do được gắn kết hữu cơ với kinh tế tri thức, khai phá dữ liệu là một lĩnh vực khoa học – công nghệ rất rộng mở. Quá trình phát triển lĩnh vực này đã ghi nhận một số bài học điển hình, những thách thức và khuynh hướng phát triển trong giai đoạn tiếp theo. Đây là những chủ đề chủ yếu của chương này.

²⁸ www.researcherid.com

Hai mục đầu tiên trình bày một số bài học và một số lối điển hình nhất trong khai phá dữ liệu. Yêu cầu bảo vệ tính riêng tư trong khai phá dữ liệu được giới thiệu trong mục 10.3. Mục cuối cùng trình bày một số khuynh hướng phát triển của khai phá dữ liệu, trong đó các khuynh hướng phát triển nổi bật nhất như khai phá dữ liệu hướng miền ứng dụng (Domain Driven Data Mining: DDDM), khai phá dữ liệu phương tiện xã hội (Social Media Data Mining) và học máy không dừng (Never-Ending Language Learning) được giới thiệu chi tiết hơn.

10.1. MỘT SỐ BÀI HỌC TRONG KHAI PHÁ DỮ LIỆU

Khai phá dữ liệu là lĩnh vực KH-CN có miền ứng dụng rất rộng lớn, thu hút ngày càng đông đảo đội ngũ những người nghiên cứu và triển khai từ nhiều lĩnh vực rất khác nhau. Tuy nhiên, có một tỷ lệ không nhỏ các dự án khai phá dữ liệu thất bại mà một trong những nguyên nhân chính yếu nhất là đội thực hiện dự án chưa được trang bị đủ kiến thức và kỹ năng. Nhấn mạnh kiến thức và kỹ năng về các phương pháp khai phá dữ liệu là rất quan trọng song thấu hiểu các bài học trong khai phá dữ liệu cũng quan trọng không kém.

Mục này giới thiệu ba lớp bài học chính: bài học về thi hành ứng dụng khai phá dữ liệu, bài học về triển khai dự án khai phá dữ liệu và một số đặc trưng cần có của các chuyên viên khai phá dữ liệu.

10.1.1. Bài học về kỹ thuật

Trong [NEM09], Robert Nisbet và cộng sự đã dành một chương để trình bày về các bài học điển hình trong khai phá dữ liệu. Mục này tóm lược nội dung các bài học nói trên với một số bổ sung.

10.1.1.1. Bài toán khai phá dữ liệu phải được xác định tốt

Khai phá dữ liệu nhằm phát hiện tri thức mới, tiềm ẩn trong dữ liệu để phục vụ chiến lược phát triển tổ chức, song bài toán

khai phá dữ liệu không thể được xác định chung chung như "tìm ra bất kỳ mẫu hấp dẫn nào" có trong dữ liệu (ngoại trừ dự án tiền khả thi điều tra xem có nên hay không tiến hành bài toán khai phá dữ liệu tại doanh nghiệp). Bài toán khai phá dữ liệu cần được xác định rõ ràng với một vài (nên không quá con số 3) mục tiêu cụ thể. Trong quá trình tiến hóa mô hình khai phá dữ liệu, thành phần xác định bài toán khai phá dữ liệu trong mô hình này ngày càng được chú trọng hơn, có nghĩa là công việc xác định bài toán khai phá dữ liệu ngày càng trở nên quan trọng hơn. Mô hình lặp khai phá dữ liệu [CCGMS98] khuyến cáo đặt ra nhiều nhất ba mục tiêu kinh doanh cho một dự án khai phá dữ liệu doanh nghiệp. Gần đây, mô hình khai phá dữ liệu hướng miền ứng dụng [CYZZ10] bao gồm có 13 bước thì có tới 5 bước có nội dung thực hiện liên quan tới vấn đề xác định bài toán; ngoài ba bước thực hiện "hiểu vấn đề" (P1), "phân tích ràng buộc" (P2) và "định nghĩa các mục tiêu phân tích và xây dựng đặc trưng" (P3) thì các bước thi hành lặp (P7, P11) cũng bao gồm vấn đề xác định bài toán. Đồng thời, các mô hình khai phá dữ liệu được đề xuất gần đây cũng ngày càng nhấn mạnh khía cạnh tương tác với chuyên gia tri thức miền ứng dụng; đôi khi các nhân viên khai phá dữ liệu cần "thương lượng" với các chuyên gia khai phá dữ liệu.

Trong miền ứng dụng kinh doanh, phát hiện gian lận, nâng cao chất lượng dịch vụ khách hàng, giảm chi phí dịch vụ... là một số mục tiêu thường gặp của các bài toán khai phá dữ liệu. Hơn nữa, khai phá dữ liệu trong kinh doanh thường theo **mô hình đủ** (Sufficiency Paradigm) mà không phải theo **mô hình tối ưu** (Efficiency Paradigm). Theo mô hình đủ, các giải pháp khai phá dữ liệu tốt nhất được xác định theo cách chúng làm việc tốt ra sao cùng với các quá trình kinh doanh khác để tăng cường sự gắn kết trong toàn bộ chuỗi lợi nhuận mà không phải hoàn toàn theo tối đa hóa hiệu quả tài chính theo mô hình tối ưu. Sự gắn kết như vậy cho phép công ty chủ động và thích ứng với sự thay đổi từ tri thức mới, chứ không phản ứng và cản trở sự thay đổi.

Như vậy, tri thức và kỹ năng được sử dụng nhằm xác định và thi thành được "mô hình đủ tốt" (good-enough model) để khai phá dữ liệu trong thực tiễn có tầm quan trọng đặc biệt [NEM09].

10.1.1.2. Truy vấn thông thường hoặc công cụ xử lý phân tích trực tuyến không thể giải được bài toán được đặt ra

Khai phá dữ liệu không phải và không thể là một "mốt" hoặc một "niềm tin" công nghệ mà khai phá dữ liệu phải được đặt xứng tầm phát hiện tri thức kinh doanh mà không phải để trả lời cho các câu hỏi nghiệp vụ, hay như cách nói dân gian "dùng dao phay để cắt tiết gà" khi tiến hành khai phá dữ liệu. Một số nội dung phân biệt giữa bài toán khai phá dữ liệu với các bài toán truy vấn hoặc xử lý phân tích trực tuyến (Online Analysis Processing: OLAP) đã được đề cập tại Chương 1 và Chương 2. Trước khi tiến hành một dự án khai phá dữ liệu, chúng ta cần kiểm tra, thử nghiệm một cách rộng rãi xem các phương tiện truy vấn thông thường hoặc các công cụ OLAP có đạt được các mục tiêu được đặt ra hay không. Trong trường hợp các phương tiện và công cụ đã biết không thể đạt được mục tiêu hoặc đạt mục tiêu nhưng chi phí lao động quá nhiều, vượt quá một ngưỡng cho phép thì mới cần tiến hành dự án khai phá dữ liệu.

Yêu cầu phân biệt sự khác nhau bản chất khi xác định bài toán khai phá dữ liệu với bài toán truy vấn, thống kê, OLAP thông thường là có tính cốt lõi để đảm bảo sự thành công của dự án khai phá dữ liệu. Chỉ trong bối cảnh dự án khai phá dữ liệu được xem xét công phu thì dự án mới được đặt đúng tầm cao của nó, do đó mới huy động được đủ tài nguyên cần thiết để thực hiện quá trình khai phá dữ liệu.

Thực tiễn ở Việt Nam, nhiều trường hợp vi phạm bài học này: đặt bài toán khai phá dữ liệu dù chưa tiến hành khai phá công cụ OLAP. Một nguyên nhân chủ yếu dẫn tới sai sót như vậy là chưa đủ tri thức và kỹ năng làm chủ các phương tiện truy vấn thông thường hoặc các công cụ OLAP. Hạn chế này rất hay xảy ra với những người mới làm quen với lĩnh vực khai phá dữ liệu, chẳng

hạn, lầm tưởng một số kết quả thống kê thông thường với việc thực hiện bài toán khai phá dữ liệu. Thậm chí, nhiều trường hợp còn coi nhận định kiểu "sinh viên vùng đồng bằng học tốt hơn sinh viên vùng sâu vùng xa" như là kết quả thực hiện một bài toán khai phá dữ liệu.

10.1.1.3. Dữ liệu phải sẵn có cho khai phá dữ liệu

Như đã nói, chúng ta có cảm nghĩ rằng bài học này là rất tầm thường, tuy nhiên, trên thực tế, dữ liệu không phải luôn luôn có sẵn để khai phá dữ liệu. Tình huống đầu tiên dẫn tới tình trạng này là do dữ liệu được lưu trữ trên phạm vi toàn bộ doanh nghiệp (có thể phân tán trên phạm vi rất rộng và có thể được tổ chức lưu trữ dưới dạng các hệ thống di chúc), và dữ liệu được định dạng rất khác nhau. Để dữ liệu là sẵn sàng cho khai phá dữ liệu, cần dành công sức và thời gian thích hợp cho thu thập dữ liệu (Chương 3) mà điều này không phải luôn được con người sẵn sàng chấp nhận. Khi thực hiện bài toán khai phá dữ liệu, chúng ta thường có cảm giác nhảm chán với công việc thu thập dữ liệu mang tính thủ công, không có tính sáng tạo cho nên chúng ta thường tự thỏa mãn với lượng dữ liệu nào đó mà chúng ta cho là đã tương đối đầy đủ. Ví dụ, trong bài toán khai phá dữ liệu web liên quan tới lĩnh vực y tế và chăm sóc sức khỏe, do nhiều nguyên nhân, chúng ta thường bỏ qua việc nghiên cứu kỹ lưỡng cấu trúc mạng các trang web liên quan tới y tế và chăm sóc sức khỏe mà chỉ tập trung vào một số địa chỉ web mà chúng ta coi là điển hình để tải về nội dung các trang web. Chúng ta thường bỏ qua phương pháp xác định "danh sách địa chỉ nhân" của các thuật toán duyệt web (crawling).

Đôi khi, ở một số tổ chức, dữ liệu cho bài toán khai phá dữ liệu phải được tích hợp từ các bộ dữ liệu do nhiều bộ phận nắm giữ, tính cát cứ cục bộ tại một số bộ phận trong tổ chức đó cũng tạo khó khăn cho việc thu thập dữ liệu.

Tình huống thứ hai dẫn tới tình trạng dữ liệu chưa sẵn có trong trong một số trường hợp là do có một số ràng buộc pháp lý ngăn chặn việc truy cập dữ liệu nhạy cảm.

Khi quyết định thực hiện một dự án khai phá dữ liệu, cần tiến hành việc cam kết đối với công việc thu thập dữ liệu một cách có hệ thống để dữ liệu sẵn sàng cho khai phá dữ liệu và đảm bảo cam kết được thực hiện khi triển khai. Cần dành đủ công sức, thời gian và cơ chế cho thu thập dữ liệu đối với tình huống thứ nhất hoặc thực thi các giải pháp khai phá dữ liệu với tính riêng tư đối với tình huống thứ hai (mục 10.3 trình bày chi tiết hơn về khai phá dữ liệu với tính riêng tư).

10.1.1.4. Dữ liệu phải đủ, sạch và liên quan tới bài toán khai phá dữ liệu

Chương 2 trình bày quá trình tiến hóa về mô hình khai phá dữ liệu cho thấy tri thức miền ứng dụng ngày càng có vị trí quan trọng hơn trong quá trình phát hiện tri thức từ dữ liệu. Tri thức miền ứng dụng giúp làm tường minh bài toán khai phá dữ liệu, do đó, định rõ được tính đủ, tính sạch và tính liên quan của dữ liệu tới bài toán khai phá dữ liệu. Chương 3 trình bày nội dung bước tiền xử lý dữ liệu đảm bảo dữ liệu có chất lượng tốt cho quá trình khai phá.

Nền tảng dữ liệu cho bài toán khai phá dữ liệu là các CSDL tác nghiệp mà chúng được thiết kế với mục đích phục vụ hoạt động nghiệp vụ hàng ngày và hầu hết trong số đó thường không kèm theo mục đích ứng dụng khai phá dữ liệu cho nên việc đảm bảo tính liên quan của dữ liệu sẵn có cũng không là một công việc dễ dàng. Hiểu bài toán khai phá dữ liệu và hiểu dữ liệu đảm bảo tính liên quan của dữ liệu với bài toán khai phá dữ liệu.

Tính đủ của dữ liệu được đảm bảo bằng việc thu thập dữ liệu đủ đại diện cho miền ứng dụng. Tính sạch của dữ liệu được đảm bảo bằng quá trình hiểu dữ liệu, làm sạch dữ liệu, lựa chọn đặc trưng (như đã đề cập tại Chương 2). Tính sạch của dữ liệu có quan hệ với tính đầy đủ khi để đảm bảo rằng dữ liệu đủ mà không dư thừa vì dư thừa dữ liệu có thể gây ra nhiễu.

Để kiểm tra các tiêu chí này được đảm bảo, việc tiến hành một nghiên cứu thí điểm phân tích dữ liệu qua mẫu sẵn có là rất hữu ích. Nghiên cứu thí điểm cần phải làm rõ các vấn đề tồn tại

cho chất lượng dữ liệu, từ đó ước tính được thời gian và nỗ lực cho tiền xử lý dữ liệu.

Như đã đề cập tại Chương 2, công việc hiểu và chuẩn bị dữ liệu rất mất nhiều thời gian và công sức. Nhiều tác giả nhận định rằng Quy tắc 80:20 (Quy tắc/Luật Pareto) cũng hiện diện trong pha xây dựng mô hình khai phá dữ liệu, theo đó, khoảng 80% thời gian là dành cho việc chuẩn bị dữ liệu còn 20% còn lại là dành cho việc dạy và kiểm thử mô hình. Quy tắc này đòi hỏi tính kiên trì của các chuyên viên khai phá dữ liệu.

10.1.1.5. Các tri thức mới phải có tính hành động

Quá trình khai phá dữ liệu có thể tạo ra tri thức mới nhưng tri thức đó phải có tính hành động trong điều kiện của tổ chức và cho ra kết quả đáp ứng mục tiêu của tổ chức. Tính hành động (hay áp dụng được) của mẫu hay độ đo hấp dẫn đã được đề cập ở Chương 2. Tuy nhiên, việc thực thi các độ đo hấp dẫn mới chỉ cho phép nhận định rằng mẫu đó có thể là hấp dẫn mà không phải quyết định mẫu chắc chắn có tính hành động.

Do phụ thuộc vào điều kiện của tổ chức cho nên tính hành động của tri thức đối với các tổ chức khác nhau là khác nhau. Ví dụ, trong một công ty tiếp thị trực tiếp, có thể triển khai các kết quả khai phá dữ liệu theo một số cách:

- Thông qua giao diện dành riêng tới các phần mềm ứng dụng hiện có để tri thức mới truy cập được đối với người dùng ít kinh nghiệm.
- Tối ưu hóa các chiến dịch tiếp thị gửi đi. Với thư trực tiếp có thể đạt được một giảm giá 20-40%.
- Triển khai kết quả trong các kênh khác, ví dụ như trung tâm cuộc gọi. Kết quả khai phá dữ liệu trực tuyến có thể được dùng cho hộp thoại điều khiển. Nếu chúng ta kết hợp nội dung dữ liệu của một cuộc đối thoại với các dữ liệu phía sau từ hệ thống hoạt động, chúng ta có được một hệ thống tiếp thị rất mạnh mẽ.

10.1.2. Bài học về triển khai dự án

Nada Lavrac và cộng sự [LMFHL04] trình bày một số nhận định sau đây liên quan tới quá trình triển khai dự án khai phá dữ liệu tại công ty:

- Hầu hết các chuyên gia miền ứng dụng (doanh nhân, quản lý tiếp thị, đại diện bán hàng, quản lý đảm bảo chất lượng, nhân viên an ninh, v.v) là những người làm việc trong ngành công nghiệp chỉ quan tâm đến khai phá dữ liệu theo góc độ là chúng giúp họ làm tốt hơn công việc riêng của họ. Họ không quan tâm chi tiết kỹ thuật và càng không muốn quan tâm tới vấn đề tích hợp. Chính vì lý do này mà cần tạo ra một cơ chế kết hợp hiệu quả các chuyên gia khai phá dữ liệu với các chuyên gia miền ứng dụng.

- Ứng dụng khai phá dữ liệu thành công cần được tích hợp hoàn toàn với một ứng dụng tiếp thị, một công cụ quản lý quan hệ khách hàng (Customer relationship management: CRM), một môi trường quản lý dịch vụ, một hệ thống kiểm kê hoặc một công cụ quản lý triệu chứng và sức khỏe. Để hoàn thành việc tích hợp nói trên, lời giải cho bài toán khai phá dữ liệu thường không phải là lời giải tối ưu chỉ cho bài toán khai phá dữ liệu mà cần là lời giải đủ tốt song cho phép phù hợp với các bài toán cần tích hợp để tạo ra tác động trực tiếp vào mục tiêu phát triển của tổ chức.

- Đưa một thuật toán thành công trong phòng thí nghiệm, ngay cả với dữ liệu lấy từ thực tiễn cuộc sống, trở thành một ứng dụng khai phá dữ liệu có hiệu quả trong công nghiệp có thể lại phải qua một quá trình lâu dài. Các vấn đề như hiệu quả chi phí, quản lý, bảo trì, tích hợp phần mềm, tái công nghệ lao động và quá trình kinh doanh phải được tính toán theo suốt quá trình lâu dài đó.

- Tương tự như mọi dự án CNTT khác, toàn bộ dự án khai phá dữ liệu phải nhận được sự hỗ trợ của người quản lý hàng đầu của công ty, cần được thực hiện bởi các nhóm nhỏ với tích hợp nội bộ mạnh và một phong cách quản lý linh hoạt. Vấn đề chủ sở hữu cần xác định rõ người chịu trách nhiệm dự án khai phá dữ liệu. Phương án tốt là người chịu trách nhiệm dự án không phải là một

nà phân tích kỹ thuật hoặc chuyên gia tư vấn mà phải là một người có trách nhiệm kinh doanh trực tiếp, ví dụ như, một người thuộc môi trường bán hàng hoặc tiếp thị. Điều này mang lại lợi ích cho tích hợp bên ngoài đối với giải pháp khai phá dữ liệu.

- Dự án khai phá dữ liệu góp phần tăng cường tri thức tổ chức vì vậy việc thực hiện các dự án thí điểm với đường cong học dốc (steep learning curve) có tầm quan trọng sống còn. Người sử dụng hệ thống khai phá dữ liệu trở nên thành thạo chỉ với thời gian và nỗ lực ở mức tối thiểu. Hoàn vốn đầu tư dương nên được hoàn thành trong vòng từ 6 đến 12 tháng. Kết quả ứng dụng khai phá dữ liệu không chỉ là vấn đề kỹ thuật phức tạp liên quan đến các nhân viên kỹ thuật mà chủ yếu là tác động đến một nhóm rộng lớn con người trong tổ chức, vì vậy, dự án cần được quản lý một cách chặt chẽ.

Sarabjot S. Anand [AGHLRW07] đề cập tới vai trò và thi hành các chuẩn (công nghiệp) khi nhúng một công nghệ khai phá dữ liệu đứng riêng rẽ vào công nghệ tích hợp được truy cập và sử dụng rộng rãi trong môi trường kinh doanh của doanh nghiệp (nói riêng) và trong môi trường thực hiện sứ mạng của tổ chức (nói chung). Các chuẩn được xây dựng nhằm làm cho quá trình tích hợp này trong suốt và minh bạch. Chuẩn CRISP-DM (The CRoss-Industry Standard Process for Data Mining, như đã đề cập tại Chương 2) với bốn chiều ngũ cảnh miền ứng dụng (Application Domain), kiểu bài toán khai phá dữ liệu (Data Mining Problem Type), khía cạnh kỹ thuật (Technical Aspect) và các công cụ và kỹ thuật (Tools and Techniques) được coi là một chuẩn công nghiệp phổ dụng khi đưa các dự án khai phá dữ liệu vào ứng dụng thực tiễn.

10.1.3. Đặc trưng của chuyên viên khai phá dữ liệu

Khai phá dữ liệu là một loại hoạt động có độ phức tạp cao, tốn thời gian và công sức và thường đòi hỏi một quá trình lâu dài. Chương 1 đã đề cập tới một số đặc trưng của nhà khoa học dữ liệu mà về bản chất họ cũng chính là chuyên viên khai phá dữ liệu. Để “thi hành sáng tạo hoạt động khảo sát và phân tích; tăng cường tư

vấn, hợp tác, và phối hợp năng lực của những người khác để tiến hành nghiên cứu và giáo dục bằng các bộ dữ liệu số; đi tiên phong trong việc phát triển sáng tạo trong lĩnh vực công nghệ cơ sở dữ liệu và khoa học thông tin, bao gồm phương pháp trực quan hóa dữ liệu và phát hiện tri thức để áp dụng vào các lĩnh vực khoa học và giáo dục liên quan đến các bộ dữ liệu; thi hành một cách tốt nhất cả theo khía cạnh thực tiễn lẫn khía cạnh công nghệ; đóng vai trò cố vấn để khởi tạo hoặc chuyển đổi dữ liệu cho các nhà điều tra, sinh viên và những người khác có quan tâm tới khoa học dữ liệu; thiết kế và thi hành các chương trình giáo dục và tiếp cận cộng đồng làm cho lợi ích của các bộ dữ liệu và thông tin khoa học kỹ thuật số tới các nghiên cứu viên, giảng viên, sinh viên và công chúng trong một phạm vi rộng nhất có thể được", họ cần có đặc trưng riêng để nhận ra được các tri thức hữu ích, cần thiết từ "núi dữ liệu đồ sộ". Giám đốc thông tin (CIO) được coi như một chuyên viên khai phá dữ liệu cao cấp mà đặc trưng của loại chuyên viên cao cấp này đã được giới thiệu tại Chương 2. Những đặc trưng được đề cập dưới đây liên quan tới chuyên viên khai phá điển hình.

Theo các chuyên gia hàng đầu về khai phá dữ liệu, ngoài những đặc trưng của chuyên viên CNTT nói chung, chuyên viên khai phá dữ liệu cần có các đặc trưng sau đây [NM09]:

- *Tính kiên trì:* Cần kiên trì "tấn công" một vấn đề khai phá dữ liệu một cách liên tục và từ các góc độ khác nhau. Cần thực hiện việc tự động hóa các bước cần thiết, đặc biệt khi thực hiện các bài kiểm tra lấy mẫu lại. Cần huy động hoạt động kiểm tra, đánh giá ngoài (bao gồm đánh giá chéo) khi triển khai công việc cũng như trong việc đánh giá hiệu quả của mô hình. Phản biện khách quan, phát hiện sai sót của mô hình, nghiên cứu các tình huống phá vỡ mô hình là các giải pháp cần thiết khi xây dựng mô hình khai phá dữ liệu.

- *Thái độ làm việc:* Thứ nhất, công việc khai phá dữ liệu đòi hỏi tinh thần lạc quan, tin tưởng vào kết quả phát hiện tri thức khi tiến hành một quá trình nhiều khăn khó khăn như khai phá dữ liệu. Thứ hai, cần giữ một thái độ đúng mức về kết quả khai phá dữ liệu.

• *Làm việc nhóm:* Phải hợp tác chặt chẽ với các chuyên gia kinh doanh và thống kê để có được tiến độ tốt nhất cho dự án. Cần đảm bảo chắc chắn rằng mỗi đối tác đều có thể phát triển nghề nghiệp thông qua sự thành công của dự án. Chỉ có một nhóm cộng tác hiệu quả của các chuyên gia khai phá dữ liệu, kinh doanh, thống kê mới tạo ra được nhận thức như vậy. Không phải tất cả mọi người đã muốn dự án thành công ngay từ đầu. Đôi khi, các chuyên gia kinh doanh e ngại về các bí mật công việc, lo lắng về các mối nguy hiểm tiềm năng khi dự án khai phá dữ liệu đào sâu vào miền hoạt động của mình.

• *Tính khiêm tốn:* Học hỏi từ những người khác (đặc biệt là các chuyên gia miên ứng dụng) để san lấp các lỗ hổng về tri thức miên ứng dụng cũng như quy trình tổ chức của đơn vị triển khai dự án khai phá dữ liệu. Tính khiêm tốn giúp chuyên viên khai phá dữ liệu hiểu về miên ứng dụng (nói riêng hiểu dữ liệu) toàn diện hơn do thu thập được thông tin toàn diện từ lớp rộng lớn những người có liên quan. Cần có tinh thần thứ lỗi tốt khi gặp hiện tượng phát biểu sai của khách hàng và những người liên quan. Tính khiêm tốn còn được thể hiện trong việc không coi những công nghệ mà mình đã nắm bắt được là đặc hiệu vạn năng mà cần phải biết lựa chọn từ nhiều công nghệ thay thế nhau để lựa chọn ra được một công nghệ phù hợp với từng bài toán khai phá dữ liệu cụ thể.

10.2. MỘT SỐ LỖI THƯỜNG GẶP TRONG KHAI PHÁ DỮ LIỆU

Mục 10.1 đã giới thiệu một số bài học kinh nghiệm trong khai phá dữ liệu được đúc kết từ kết quả thành công hay thất bại khi triển khai các ứng dụng khai phá dữ liệu. Đồng thời và tương ứng với các bài học đó, các lỗi điển hình trong khai phá dữ liệu cũng được phát hiện. Chẳng hạn, bài học về dữ liệu phải đầy đủ, sạch sẽ và liên quan tới bài toán khai phá dữ liệu sẽ được tương ứng với lỗi thiếu dữ liệu. Tuy nhiên, việc trình bày tường minh các lỗi thường gặp nhất trong khai phá dữ liệu cũng là một nội dung hết sức cần thiết để nhắc nhở những người mới bắt đầu tham gia triển khai dự án khai phá dữ liệu. Công bố các kết quả không

mong đợi [CD10], nhận diện và công bố các lỗi thường gặp trong khai phá dữ liệu là những hoạt động có tầm quan trọng trong cộng đồng những người nghiên cứu và triển khai trong lĩnh vực này.

Danh sách các lỗi thường gặp trong hoạt động khai phá dữ liệu được giới thiệu dưới đây do Robert Nisbet và cộng sự [NEM09] nhận diện. Theo các tác giả, đầu tiên là một danh sách 10 lỗi điển hình nhất được xác định, và sau đó lỗi "thiếu dữ liệu" tưởng như "ai cũng biết" được bổ sung và được đánh chỉ số 0 ("không"). Nội dung mục này được tổng hợp từ tài liệu [NEM09] và một số tài liệu liên quan khác, trong đó có chuyên mục "Các kết quả không mong đợi" (Unexpected results) của Tạp chí ACM SIGKDD Explorations newsletter số 2, tập 12 năm 2010.

Thiếu dữ liệu

"Dữ liệu" được đề cập ở đây là tập ví dụ được chọn làm đại diện cho miền dữ liệu của bài toán khai phá dữ liệu. Tính đại diện của tập ví dụ đòi hỏi việc hình thành tập ví dụ đáp ứng yêu cầu tập ví dụ "duy trì" cấu trúc của miền dữ liệu mà cấu trúc cơ bản nhất là phân bố xác suất của dữ liệu. Robert Nisbet và cộng sự [NEM09] cho một ví dụ về tình huống tri thức tiềm ẩn được ví như "cái kim trong đống cỏ khô" dữ liệu trong bài toán phát hiện gian lận tín dụng ngân hàng. Mô hình dữ liệu được hình thành từ tập ví dụ mẫu cho phép không bỏ sót các mẫu tiềm ẩn đó.

Lỗi thiếu dữ liệu xuất phát từ một số nguyên nhân. Thứ nhất, một ví dụ thường được tạo ra bằng phương pháp thủ công với những thao tác dễ gây nhảm chán cho người thực hiện. Thứ hai, "hiểu dữ liệu" (như trình bày ở Chương 3) là một công việc nghiên cứu và triển khai công phu, trong đó đáng chú ý là công việc kiểm nghiệm giả thiết mô hình dữ liệu. Không hiểu tốt dữ liệu dẫn đến tình huống nhận được một tập ví dụ với kích thước lớn song vẫn trong tình trạng thiếu dữ liệu do chọn nhầm (thừa) ví dụ vừa tốn công sức vừa có thể làm sai lệch mô hình dữ liệu.

Học bán giám sát là một định hướng giải pháp tốt để khắc phục lỗi thiếu dữ liệu, tuy nhiên, nó không phải là giải pháp vạn

năng áp dụng được cho mọi trường hợp [Gold10, Zhu08]. Trong những trường hợp áp dụng được giải pháp học bán giám sát, "hiểu dữ liệu" càng có vai trò đặc biệt quan trọng.

Cần phân biệt khái niệm tập dữ liệu trong tình huống đánh giá một thuật toán khai phá dữ liệu với tình huống triển khai một dự án khai phá dữ liệu. Trong tình huống đầu tiên, các bộ dữ liệu "chuẩn" của công đồng nghiên cứu được công bố trên Internet là một lựa chọn tốt. Tình huống thứ hai công phu hơn, vừa phải sử dụng các bộ dữ liệu chuẩn vừa phải hiểu dữ liệu miền ứng dụng để hình thành tập ví dụ mẫu.

Quá chú trọng vào việc học

Nhấn mạnh công việc tinh chỉnh mô hình theo dữ liệu để nhận được một mô hình tốt theo tập ví dụ là một biểu hiện của tình huống quá chú trọng vào việc học. Việc làm như vậy thường dẫn đến tình huống "quá khớp" (overfitting) giữa mô hình và dữ liệu bởi vì dù bước hiểu dữ liệu có được tiến hành công phu đến mấy thì tập ví dụ cũng không thể đại diện đầy đủ cho dữ liệu miền ứng dụng. Khi chú trọng làm khít mô hình với dữ liệu học, chúng ta có thiên hướng nhấn mạnh đặc trưng riêng của tập ví dụ hơn là các đặc trưng chung của dữ liệu miền ứng dụng mà tập ví dụ đại diện.

Dự trữ ví dụ để đánh giá sau mô hình là một giải pháp định hướng cho phép khắc phục lỗi quá chú trọng vào việc học. Tuy nhiên, ví dụ học là tài nguyên quá cho xây dựng mô hình cho nên không phải lúc nào cũng dành được ví dụ dự trữ. Trong trường hợp đó, việc lấy mẫu bổ sung (resampling) cần được tiến hành.

Trong nhiều trường hợp, kỹ thuật đánh giá chéo (cross-folds valuation) cũng được coi là một giải pháp khắc phục lỗi mô hình "quá khít" với ví dụ học. Khi áp dụng kỹ thuật đánh giá chéo, tính ngẫu nhiên của việc phân chia tập ví dụ có ý nghĩa rất quan trọng.

Dựa vào chỉ một kỹ thuật

Trong toán học, kết quả nghiên cứu là bản chất còn kỹ thuật thi hành để đi tới kết quả chỉ là thứ yếu. Theo cách nói của Gian-

Carlo Rota²⁹, mỗi nhà toán học (thậm chí cả nhà toán học vĩ đại người Đức David Hilbert) chỉ có một vài mẹo vặt (nguyên văn tiếng Anh: "Every mathematician has only a few tricks"). Trong khai phá dữ liệu, thì có điều khác biệt là chúng ta không chỉ dựa vào những kỹ thuật khai phá dữ liệu quen biết để thực hiện các bài toán khai phá dữ liệu khác nhau. Sự khác biệt này có xuất phát điểm từ sự khác biệt của nguồn gốc tri thức "mới". Trong toán học, tri thức mới có được dựa trên suy luận lôgic, biện luận, chứng minh của nhà toán học theo những kỹ thuật riêng vì vậy nhà toán học thường ưa chuộng các kỹ thuật sẵn có của mình. Trong khi đó, trong khai phá dữ liệu, tri thức mới được tiềm ẩn trong dữ liệu, không phụ thuộc vào ý kiến chủ quan của người khai phá dữ liệu, vì vậy, không thể dựa vào các kỹ thuật nào đó quen thuộc của họ.

Việc áp dụng một số kỹ thuật khác nhau để giải bài toán khai phá dữ liệu cho phép chúng ta đưa ra được nhiều phương án nhằm mục đích đánh giá chúng và lựa chọn phương án tốt nhất trong số các phương án đã được thi hành. Tiến hành công việc như vậy có thể gây ra sự tốn kém nhất định, đặc biệt trong thực nghiệm, song là rất cần thiết.

Tích hợp các kỹ thuật khác nhau là một tiếp cận được xem xét khi giải quyết các bài toán khai phá dữ liệu. Mỗi kỹ thuật khai phá dữ liệu phù hợp tốt với một loại mô hình dữ liệu tương ứng, tuy nhiên, giả thiết về mô hình dữ liệu miền ứng dụng không phải là chính xác hoặc hoàn toàn chính xác. Khi tích hợp nhiều kỹ thuật khai phá dữ liệu với nhau thì cách kỹ thuật này bổ sung cho nhau những hạn chế về giả thiết mô hình dữ liệu của mỗi mô hình. Robert Nisbet và cộng sự [NEM09] đã chỉ dẫn cụ thể về lợi thế của tích hợp mô hình trong nhiều ứng dụng khai phá dữ liệu.

Christophe Giraud Carrier và Margaret H. Dunham [CD10] nhấn mạnh rằng không phải mọi kỹ thuật khai phá dữ liệu được coi là tốt thì đều áp dụng được cho mọi tình huống. Các tác giả tổng hợp ba trường hợp về các kỹ thuật hiệu quả rất phổ biến song trong một số trường hợp các kỹ thuật này lại cho kết quả rất hạn chế.

²⁹ <http://alumni.media.mit.edu/~cahn/life/gian-carlo-rota-10-lessons.html>

(i) mô hình ngữ nghĩa ẩn Latent Semantic Indexing LSI không bao gồm được các mối quan hệ giữa từ, chủ đề ẩn và tài liệu trong các bộ dữ liệu TREC,

(ii) kỹ thuật đánh giá chéo theo độ đo AUC cho hiệu quả thấp trong thường hợp sử dụng cơ chế stack và mẫu có ít ví dụ dương,

(iii) hiệu năng của các bộ phân lớp không tăng khi thông tin/ví dụ mẫu được bổ sung vào tập ví dụ mẫu. Một nguyên nhân liên quan tới điểm phù hợp trên là cấu tạo của kiến trúc lớp không phản ánh tương ứng với phân bố của các thể hiện.

Những khuyến cáo trên đây giúp mọi người tránh lỗi chỉ sử dụng các kỹ thuật khai phá dữ liệu quen thuộc.

Đặt sai câu hỏi

Lỗi đặt sai câu hỏi xuất hiện ở hai cấp độ xác định mục tiêu và xác định mô hình mục tiêu.

Thứ nhất, đặt câu hỏi sai có nguyên nhân từ xác định sai mục tiêu khai phá dữ liệu. Như vậy, lỗi này liên quan mật thiết tới bài học cần đặt đúng bài toán khai phá dữ liệu. Mục tiêu của bài toán khai phá dữ liệu gắn kết với mục tiêu kinh doanh, việc chuyển đổi từ mục tiêu kinh doanh thành mục tiêu khai phá dữ liệu là rất khó khăn, phức tạp, vì vậy sự cộng tác của các chuyên gia nhiều lĩnh vực là hết sức quan trọng.

Một vài nguyên nhân điển hình gây ra tình huống đặt sai câu hỏi (xác định sai mục tiêu) khai phá dữ liệu là do công sức làm việc để hiểu bài toán và dữ liệu chưa được bỏ ra đúng mức độ, do những người liên quan đã áp đặt việc thừa kế quá mức bài toán khai phá dữ liệu sẵn có mà được coi là cùng loại hoặc do đã ưu tiên quá mức kinh nghiệm của các chuyên viên khai phá dữ liệu.

Thứ hai, mục tiêu khai phá dữ liệu được đặt đúng song xác định mô hình mục tiêu có thể không đúng. Câu hỏi đặt ra cho ứng dụng khai phá dữ liệu là một bộ phận trong cách thức xác định mô hình mục tiêu. Phân tích đa chiều mạnh cho phép xác định mô hình mục tiêu tốt.

Chỉ “nghe” từ dữ liệu

Tiếp cận khai phá dữ liệu “tìm những mẫu (tri thức) mới, hữu dụng, có giá trị, tiềm ẩn trong dữ liệu” không đồng nhất với quan niệm rằng dữ liệu sẵn có là tất cả các nguồn tài nguyên có thể có phục vụ quá trình khai phá dữ liệu. Bài toán khai phá dữ liệu cần những nguồn tài nguyên bổ sung khác.

Một mặt, dữ liệu chúng ta thu thập được có thể chưa bao gồm hết các đặc trưng dữ liệu miền ứng dụng cho bài toán khai phá dữ liệu. Trong thực tiễn, dữ liệu thu thập được từ các hệ thống quan sát mà các hệ thống quan sát đó không phải lúc nào cũng cho phép hình thành thông tin toàn diện mô tả dữ liệu. Có thể nói một số “đặc trưng” (thuộc tính) của dữ liệu bị bỏ sót trong quá trình thu thập dữ liệu. Tri thức miền ứng dụng cho phép giảm thiểu tình huống bỏ sót như vậy.

Mặt khác, mặc dù chuyên gia khai phá dữ liệu có khả năng “nghe được các câu chuyện do dữ liệu kể” nhưng để nghe được “câu chuyện từ dữ liệu” thì họ cần phải được cung cấp thêm tri thức miền ứng dụng (do các chuyên gia miền ứng dụng cung cấp), nội dung và ý nghĩa mục tiêu của bài toán khai phá dữ liệu (do người quản lý cao cấp cung cấp).

Như trình bày trong Chương 1, hệ thống khai phá dữ liệu chứa một cơ sở tri thức như một thành phần tách ra khỏi tài nguyên dữ liệu đầu vào cho bài toán khai phá dữ liệu. Hơn nữa, cơ sở tri thức này là không đầy đủ và các yếu tố trong cơ sở tri thức này có thể được bổ sung, thay đổi, hay loại bỏ.

Chấp nhận rò rỉ từ tương lai

Tiêu đề của mục nhỏ này ám chỉ rằng có sự nhập nhằng giữa đầu vào và đầu ra của bài toán khai phá dữ liệu, hay nói khác đi, tồn tại một sự giao thoa nào đó của tập biến đầu vào với tập biến đầu ra. Sự giao thoa như vậy có nguyên nhân từ việc hiểu dữ liệu trong giai đoạn tiền xử lý dữ liệu chưa chính xác. Robert Nisbet và cộng sự [NEM09] đưa ra một số ví dụ của loại lỗi này, trong đó có ví dụ đi tìm luật liên quan tới sự phá sản của các công ty từ việc nghiên cứu dữ liệu của các công ty đang tồn tại.

Một ví dụ tầm thường của lỗi này là sự giao thoa giữa tập ví dụ học với tập ví dụ đánh giá mô hình. Một số người mới làm quen với khai phá dữ liệu khi tiến hành đánh giá mô hình lại cho phép ví dụ học đóng vai trò của dữ liệu kiểm thử.

Giảm bớt ví dụ "làm phiền"

Trong quá trình hiểu dữ liệu, có thể chúng ta phát hiện ra một vài ví dụ khác biệt hoàn toàn với đặc trưng chung của tập ví dụ còn lại. Những ví dụ khác biệt này tồn tại trong thực tiễn song có vẻ như nó gây khó khăn rất lớn khi xây dựng mô hình. Trong trường hợp đó, dễ xảy ra nhận định rằng nếu bỏ đi các ví dụ này, quá trình xây dựng mô hình vừa đơn giản và mô hình xây dựng được có vẻ rất phù hợp với tập dữ liệu còn lại. Từ nhận định này dẫn tới việc loại bỏ các ví dụ khác biệt như đã nói và lỗi giảm bớt ví dụ làm phiền xuất hiện.

Christophe Giraud Carrier và Margaret H. Dunham [CD10] khuyến cáo về việc cần phải tránh những lỗi khi lựa chọn và sử dụng dữ liệu đầu vào.

Đáp ứng mọi yêu cầu

Lỗi này có nguyên nhân từ nhận thức chưa toàn diện về khai phá dữ liệu, chưa hình dung hết quá trình khó khăn và phức tạp của khai phá dữ liệu. Trong một số trường hợp, quyết định nóng vội mong muốn ứng dụng một công nghệ tiên tiến cũng là một nguyên nhân dẫn tới tình trạng sự chuẩn bị tri thức và kỹ năng chưa theo kịp với mong muốn đó. Các nguyên nhân nói trên dẫn tới tình trạng xác định chưa đúng phạm vi kết quả của khai phá dữ liệu.

Yêu cầu đối với một ứng dụng hay phạm vi kết quả của một ứng dụng khai phá dữ liệu cần được xác định trong một giới hạn mục tiêu phù hợp và được khuyến cáo là không nên vượt quá ba mục tiêu cho một ứng dụng.

Quá tập trung vào việc đi tìm mô hình tốt

Có thể coi lỗi này là đồng dạng với lỗi dựa vào chỉ một kỹ thuật khai phá dữ liệu. Tìm được mô hình tốt, phù hợp với ngữ

cánh của bài toán khai phá dữ liệu là định hướng chủ đạo của quá trình khai phá dữ liệu. Nếu quá tập trung vào việc đi tìm một mô hình tốt, một mặt, sẽ xảy ra hiện tượng coi nhẹ các thành phần quan trọng khác của quá trình khai phá dữ liệu, mặt khác, dễ bị lạc vào "mê cung" khi tìm kiếm mô hình.

Như đã biết, tiếp cận lời giải cho bài toán khai phá dữ liệu là "lời giải đủ tốt" mà không phải là "lời giải tốt ưu", việc tìm mô hình tốt cần được thi hành với mức độ tập trung phù hợp song cũng cần dành thời gian và công sức cho các công việc khác trong toàn bộ quá trình phát hiện tri thức từ dữ liệu, đặc biệt là cần đầu tư thích đáng cho công việc tiền xử lý dữ liệu, biểu diễn dữ liệu, giải thích và trực quan hóa kết quả.

Bảng 10.1. So sánh một số kỹ thuật phát hiện và trích chọn danh sách trên Web [Weni10]

Algorithm	# Extracted	Recall
Google Sets	8	7.48%
WebTables	41	38.32%
MDR	34	31.78%
Visual Ext.	59	55.14%

Như đã biết, mỗi một mô hình khai phá dữ liệu đòi hỏi miền ứng dụng bài toán khai phá dữ liệu cần đáp ứng yêu cầu giả thiết của mô hình. Dù rằng, trong hầu hết trường hợp ứng dụng khai phá dữ liệu, nếu không bác bỏ được giả thiết về một mô hình trên miền ứng dụng của bài toán thì cần chấp nhận mô hình đó, song "mô hình tốt" sẽ đòi hỏi nhiều giả thiết hơn vì vậy khả năng xuất hiện phản ví dụ để bác bỏ mô hình sẽ cao hơn.

Tim Weninger và cộng sự [Weni10] khảo sát các kỹ thuật phát hiện và trích chọn danh sách chung trên web. Các tác giả kỳ vọng rằng các kỹ thuật làm tinh vi sẽ cho một hiệu năng cao phát hiện và trích chọn danh sách chung. Tuy nhiên, kết quả thử nghiệm chỉ ra rằng kỹ thuật trực quan (theo tiếp cận đơn giản "ngây thơ") lại cho kết quả trung bình (độ hồi tưởng đạt 55%) cao

hơn nhiều so với các kỹ thuật tinh vi hơn như Google Sets, WebTables và WWT MDR (Bảng 10.1).

Mẫu tình cờ

Phát hiện tri thức từ dữ liệu được ví như "tìm kim trong đống cỏ" cho nên đặt ra yêu cầu là số lượng mẫu để học mô hình cần khoảng 10% số lượng dữ liệu có thể trong miền ứng dụng. Để đạt được tỷ lệ này hoặc chúng ta phải giảm kích thước không gian dữ liệu miền ứng dụng bằng cách bỏ đi cách mẫu thông dụng nhất (under sample) hoặc bổ sung các mẫu mới. Trong cả hai trường hợp, loại bỏ mẫu (định hướng giảm kích thước không gian dữ liệu) hoặc bổ sung mẫu mới, lỗi mẫu tình cờ xảy ra; việc loại bỏ hay bổ sung mẫu không như mong muốn.

Về lý thuyết, tập ví dụ mẫu (ví dụ học và ví dụ kiểm thử) là đại diện cho tập dữ liệu miền ứng dụng, được chọn một cách "ngẫu nhiên" từ dữ liệu miền ứng dụng. Trên thực tế, yêu cầu này rất khó thực hiện một cách tuyệt đối chính xác. Xác định tốt phân bố dữ liệu theo các đặc trưng cho phép việc xây dựng ví dụ mẫu một cách ngẫu nhiên theo phân bố đặc trưng đã được xác định cho phép giảm thiểu lỗi tình cờ. Trong trường hợp khó khăn đảm bảo tính ngẫu nhiên trong xây dựng ví dụ mẫu thì nên áp dụng một thứ tự ngẫu nhiên cho các mẫu. Thứ tự này có thể được dùng trong việc lựa chọn tập ví dụ học và tập ví dụ kiểm thử.

Theo Tim Weninger và cộng sự [Weni10], trong trường hợp của bài toán phát hiện và trích chọn danh sách, các phương pháp tinh vi được xem xét có xu hướng thiên vị trong việc lấy mẫu; sự thiên vị như vậy có thể là nguyên nhân làm cho các kỹ thuật tinh vi đó không đạt kết quả như kỳ vọng.

Ngoại suy

Lỗi ngoại suy có xuất phát điểm từ việc lạm dụng kinh nghiệm từ các dự án khai phá dữ liệu đã thực hiện. Gặp một trường hợp mà được coi là "tương tự", những kinh nghiệm đã có thường dẫn đến ngoại suy các tình huống bài toán liên quan. Một loại mẫu ngoại suy điển hình là ngoại suy theo số chiêu không

gian dữ liệu miền ứng dụng: từ kinh nghiệm trong quá khứ đối với cỡ chiềú nhỏ, ngoại suy tình huống "tương tự" đối với cỡ chiềú lớn. Liên quan tới ngoại suy theo cỡ không gian dữ liệu, Robert Nisbet và cộng sự [NEM09] nêu các nhận định sau đây của Friedman:

- Cỡ của tập ví dụ mẫu tăng cấp số nhân theo số chiềú của không gian dữ liệu,
- Lân cận của một bộ phận nhỏ dữ liệu có thể là rất lớn,
- Hầu hết các điểm là gần một cạnh của không gian mẫu hơn điểm gần nhất với nó,
- Hầu hết các điểm là khác biệt (bất thường) theo phép chiếu riêng của nó.

Những nhận định trên đây cho thấy về độ phức tạp của không gian dữ liệu miền ứng dụng là những thách thức không nhỏ khi sử dụng tiếp cận ngoại suy.

Một giải pháp tốt để có thể phát huy tốt kinh nghiệm trong quá khứ và tránh được các lỗi ngoại suy là các chuyên viên khai phá dữ liệu cần thường xuyên giao tiếp và trao đổi với nhau và với khách hàng về tình huống bài toán, nhằm bổ sung được các giả thuyết khách quan về không gian dữ liệu miền ứng dụng.

10.3. CÔNG CỤ KHAI PHÁ DỮ LIỆU

Sự phát triển về số lượng công cụ khai phá dữ liệu và doanh số của công cụ khai phá dữ liệu trong kinh doanh thông minh (business intelligence) là một minh chứng nổi bật cho ý nghĩa và tầm quan trọng của khai phá dữ liệu. Theo Ralf Mikut và Markus Reischl [MR11], thị trường toàn thế giới về kinh doanh thông minh (phần mềm và lệ phí bảo trì) đạt 7,8 tỷ đô la Mỹ vào năm 2008, trong đó có 1,5 tỷ đô la Mỹ cho các phân tích cao cấp bao gồm khai phá dữ liệu và thống kê; khu vực kinh doanh này đã tăng 12,1% so với năm 2007. Các công cụ có thị phần lớn là SAS Enterprise Miner (33,2%), IBM SPSS Modeler (14,3%), MicroSoft SQL Server Analysis Services (1,7%), Teradata Database (1,5%),

TIBCO Spotfire (1,4%). Đồng thời, nhiều công cụ phần mềm mở (miễn phí) cũng trở nên rất phổ biến, chẳng hạn như Waikato Environment for Knowledge Analysis (WEKA). Ralf Mikut và Markus Reischl [MR11] đã cung cấp một nghiên cứu tổng quan về công cụ khai phá dữ liệu và nội dung cơ bản của nghiên cứu trên được trình bày trong mục này.

10.3.1. Tiêu chí phân loại các công cụ khai phá dữ liệu

Công cụ khai phá dữ liệu được phân loại dựa theo một số tiêu chí gồm nhóm người dùng, kiểu dữ liệu, bài toán và phương pháp khai phá dữ liệu, phương án nhập dữ liệu và đưa ra kết quả, mô hình giấy phép.

Theo nhóm người dùng, công cụ khai phá dữ liệu được phân loại thành bốn nhóm là ứng dụng kinh doanh, ứng dụng nghiên cứu, phát triển thuật toán, và dạy - học. Nhóm người dùng ứng dụng kinh doanh sử dụng công cụ khai phá dữ liệu để giải quyết các bài toán áp dụng kinh doanh thương mại hóa như quản lý quan hệ khách hàng, phát hiện gian lận... Họ chủ yếu quan tâm tới các công cụ đã được thương mại hóa cung cấp hỗ trợ các CSDL lớn và tích hợp với dòng kinh doanh của doanh nghiệp. Các công cụ nổi bật nhất thuộc nhóm này là ADAPA (Zementis), CART, IBM SPSS Modeler, IBM SPSS Statistics, KXEN, MATLAB, Oracle Data Mining (ODM), SAP Netweaver Business Warehouse (BW), SAS Enterprise Miner, SQL Server Analysis Services, STATISTICA, TIBCO Spotfire. Nhóm người dùng ứng dụng nghiên cứu áp dụng công cụ khai phá dữ liệu đã được chứng minh theo phương pháp luận, các giao diện (giao diện đồ họa, giao diện khuôn dạng dữ liệu hoặc CSDL miễn ứng dụng) vào hoạt động nghiên cứu (ví dụ, công nghệ và khoa học đời sống). Nhóm người dùng phát triển thuật toán đòi hỏi các công cụ khai phá dữ liệu chứa nhiều thuật toán hiện thời để phát triển thuật toán khai phá dữ liệu mới theo hai phương diện tích hợp thuật toán mới với các công cụ và so sánh nó với thuật toán đã có. Nhóm người dùng dạy - học cần các công cụ khai phá dữ liệu trực quan, giao diện người

dùng tiện dụng và không tốn kém. Hơn nữa, nó cần công cụ có khả năng cho phép tích hợp phương pháp tự phát triển tại các trường đại học.

Có một thuộc tính cơ bản của kiểu dữ liệu là số chiều (dimension) của kiểu dữ liệu đó. Các công cụ khai phá dữ liệu làm việc với các kiểu dữ liệu là bảng đặc trưng (feature tables) hai chiều, văn bản (texts) hai chiều, chuỗi thời gian (time series) ba chiều, dãy (sequences) ba chiều, ảnh (images) bốn chiều, đồ thị (graphics) bốn chiều, ảnh ba chiều (3D graphics) năm chiều, video năm chiều, 3D video sáu chiều.

Các công cụ phần mềm bao phủ toàn bộ các bài toán khai phá dữ liệu như học giám sát (phân lớp, phân lớp mờ, hồi quy), học không giám sát (phân cụm, phân đoạn), và học bán giám sát. Công cụ khai phá dữ liệu cũng giải quyết các bài toán đi kèm các bài toán trên đây như làm sạch dữ liệu, lọc dữ liệu, trích xuất đặc trưng, chuyển dạng dữ liệu, đánh giá và lựa chọn đặc trưng, tính toán tính tương tự và phát hiện các phần tử tương tự, xác nhận mô hình, hợp nhất mô hình (hợp nhất với tri thức chuyên gia), tối ưu hóa mô hình.

Hầu hết các phương pháp học máy thống kê cổ điển và các phương pháp học máy mới hơn đều có sẵn công cụ phần mềm thi hành. Độ thường xuyên xuất hiện của các phương pháp này trong các công cụ khai phá dữ liệu là một tiêu chí so sánh chúng. Xuất hiện thường xuyên (có trong hầu hết các công cụ khai phá dữ liệu) là các phương pháp phân lớp dựa trên hàm mật độ xác suất ước tính (như Bayes), phân tích tương quan, lựa chọn đặc trưng theo thống kê, và tính toán (test) tương quan. Xuất hiện trong nhiều công cụ khai phá dữ liệu là các phương pháp cây quyết định, phân cụm, hồi quy, làm sạch dữ liệu, lọc dữ liệu, trích xuất đặc trưng, phân tích thành phần chính (PCA: principal component analysis), phân tích nhân tử (factor analysis), đánh giá và lựa chọn đặc trưng tiên tiến, tính toán độ tương tự, mạng nơron, đánh giá chéo

mô hình, tính toán (test) tương quan thống kê. Xuất hiện trong một vài công cụ khai phá dữ liệu là các phương pháp phân lớp mờ (fuzzy classification), học luật kết hợp và khai phá tập mục thường xuyên, phân tích thành phần độc lập (independent component analysis), bootstrapping, độ đo phức (complexity measures), hợp nhất mô hình, máy hỗ trợ vector (SVM), k láng giềng gần nhất (k-NN), mạng Bayes (Bayesian networks), và học các luật rõ (crisp rules). Xuất hiện trong một vài công cụ khai phá dữ liệu là các phương pháp rừng ngẫu nhiên (random forests), học hệ thống mờ, tập thô, tối ưu hóa thuật toán bằng thuật toán tiến hóa.

Về tương tác người dùng, công cụ khai phá dữ liệu được phân thành ba loại (theo mức độ tiện dụng từ thấp lên cao cho người dùng) là tương tác dòng lệnh thuần túy sử dụng một ngôn ngữ lập trình, tương tác đồ họa với cấu trúc thực đơn, tương tác đồ họa người dùng thực sự.

Mô hình xuất ra kết quả (đưa ra) và đưa vào nhập dữ liệu trong các công cụ khai phá dữ liệu có vai trò rất quan trọng. Các mô hình nhập – xuất ở đây thường tuân theo một số dạng chuẩn để làm thuận tiện hơn trong việc kết nối thông tin giữa công cụ này với các hệ thống phần mềm khác.

Công cụ khai phá dữ liệu có thể chạy trên nền hệ thống độc lập hoặc hệ thống khách/chủ. Các công cụ khai phá dữ liệu đang đi theo xu hướng chạy trên nền web và hỗ trợ chạy trên nền tính toán đám mây.

Theo mô hình giấy phép, các công cụ khai phá dữ liệu được chia thành hai nhóm chính: Sản phẩm thương mại và phần mềm nguồn mở (tự do). Công cụ khai phá dữ liệu thương mại là sự lựa chọn của nhóm người dùng áp dụng khai phá dữ liệu trong kinh doanh do các công cụ này có lợi thế về tính ổn định cao, về khả năng tích hợp với các công cụ kho dữ liệu, về bảo trì hệ thống và về hướng dẫn, đào tạo. Các nhóm người dùng khác sử dụng công cụ

khai phá dữ liệu nguồn mở (tự do) với mức độ giấy phép khác nhau. Phần mềm nguồn mở có lợi thế về sửa lỗi nhanh hơn, về tính dễ dàng phát triển, về sự tồn tại cộng đồng cùng phát triển nguồn mở. Nên lưu ý rằng phần mềm tự do hay nguồn mở không đồng nhất với tính miễn phí. Mô hình giấy phép công cụ phần mềm nguồn mở khai phá dữ liệu là mô hình giấy phép GNU General Public License của Free Software Foundation. Một số công cụ khai phá dữ liệu theo mô hình trộn như MatLab khi sử dụng phần mềm nguồn mở cho các công cụ thương mại.

10.3.2. Các kiểu công cụ khai phá dữ liệu

Dựa theo các tiêu chí phân loại nói trên, công cụ khai phá dữ liệu được phân thành hệ thống khai phá dữ liệu (Data mining suites: DMS), gói thông minh kinh doanh (Business intelligence packages: BI), gói toán học (Mathematical packages: MAT), gói tích hợp (INT), công cụ dành riêng (extensions: EXT), thư viện khai phá dữ liệu (Data mining libraries: LIB), công cụ chuyên dụng (Specialties: SPEC), công cụ nghiên cứu (research: RES), giải pháp (Solutions: SOL).

- Hệ thống khai phá dữ liệu (DMS) thi hành nhiều phương pháp giải quyết các bài toán khai phá dữ liệu, được định hướng tới miền ứng dụng rộng rãi song vẫn có các tiện ích để tạo phương án ứng dụng cụ thể. Phần lớn DMS là phần mềm thương mại, khá đắt tiền và cũng có một vài DMS nguồn mở như RapidMiner. Các DMS điển hình là IBM SPSS Modeler, SAS Enterprise Miner, Alice d'Isoft, DataEngine, DataDetective, GhostMiner, Knowledge Studio, KXEN, thành phần khai phá dữ liệu trong NAG, Partek Discovery Suite, STATISTICA, và TIBCO Spotfire.

- Gói thông minh kinh doanh (BI) chứa các hàm khai phá dữ liệu cơ bản (đặc biệt là các phương pháp thống kê) ứng dụng trong kinh doanh. Hầu hết gói BI là thương mại (IBM Cognos 8 BI, Oracle DataMining, SAPNetweaver Business Warehouse,

Teradata Database, IBM DB2 Data Warehouse, và PolyVista) nhưng cũng có gói nguồn mở (Pentaho).

- Gói toán học (MAT) cung cấp một tập lớn và mở rộng được các thuật toán và chương trình con trực quan hóa. Hiện có các gói MAT thương mại (MATLAB và R-PLUS) hoặc nguồn mở (R, Kepler).

- Gói tích hợp (INT) được mở rộng từ nhiều thuật toán khai phá dữ liệu nguồn mở. Gói tích hợp hoặc là chạy độc lập (chủ yếu được viết trên Java: KNIME, phiên bản giao diện đồ họa của WEKA, KEEL, và TANAGRA) hoặc là gói được mở rộng từ gói toán học MAT (như Gait-CAD, PRTools cho MATLAB, và RWEKA cho R).

Bảng 10.2. Quan hệ kiểu công cụ – nhóm người dùng
(+” đặc biệt hữu dụng, 0: ít hữu dụng, -: không hữu dụng) [MR11].

Types	Data Mining Suites	Business Intelligence Packages	Mathematical Packages	Integration Packages	Extensions	Data Mining Libraries	Specialties	Research Prototypes	Solutions
Number of Recent Tools	45	16	5	8	10	20	55	17	19
Business applications	+	+	-	0	0	-	0	-	0
Applied research	+	+	-	+	0	0	0	0	+
Algorithm development	-	-	+	+	-	+	0	-	-
Education	+	-	0	0	-	-	-	-	0

- Công cụ dành riêng (EXT) là tiện ích nhỏ thi hành một thuật toán khai phá dữ liệu cho các công cụ khác: Forecaster XL và XLMiner cho Excel, Toolbox Matlab Neural Networks cho Matlab. Có cả hai dạng EXT thương mại và nguồn mở.

- Thư viện khai phá dữ liệu (LIB) là một gói hàm thực hiện các phương pháp khai phá dữ liệu. Các hàm này có thể được nhúng trong các công cụ phần mềm khác bằng cách sử dụng một giao diện lập trình ứng dụng.

- Công cụ chuyên dụng (SPEC) là tương tự như DMS, nhưng chỉ thực hiện một học phương pháp đặc biệt (chẳng hạn, học phương pháp mạng nơ ron nhân tạo). SPEC cũng bao gồm nhiều kỹ thuật trực quan.

Bảng 10.3.a Các công cụ khai phá dữ liệu thương mại điển hình [MR11].

Công cụ	Kiểu	Chỉ dẫn trang web
ADAPA(Zementis)	DMS	www.zementis.com
Alice(dIsoft)	DMS	www.alice-soft.com
Bayesia Lab	SPEC	www.bayesia.com
C5.0	SPEC	www.rulequest.com
CART	SPEC	www.salford-systems.com
Data Applied	DMS	data-app lied.com
Data Detective	DMS	www.sentient.nl/?dden
DataEngine	DMS	www.dataengine.de
DataScope	DMS	www.cygron.hu
DB2 Data Warehouse	BI	www.ibm.com/software/data/infoSphere/warehouse
DeltaMaster	BI	www.bicsantz.com/deltamaster
Forecaster XL	EXT	www.alyuda.com
GhostMiner	DMS	www.fqc.pl/business intelligence/products/ghostminer
IBM Cognos BI	BI	www.ibm.com/software/data/cognos/data-mining-tools.html
IBM SPSS Modeler	DMS	www.spss.com/software/modeling/modeler
IBM SPSS Statistics	MAT	www.spss.com/software/statistics
iModel	DMS	www.biocompsystems.com/products/imodel
InfoSphere Warehouse	BI	www.ibm.com/software/data/infoSphere/warehouse
JMP	DMS	www.jmpdiscovery.com
KnowledgeMiner	SPEC	www.knowledgeminer.net
KnowledgeStudio	DMS	www.angoss.com
KXEN	DMS	www.kxen.com
Magnum Opus	SPEC	www.giwebb.com
MATLAB	MAT	www.mathworks.com
MATLAB Neural Network Toolbox	EXT	www.mathworks.com
Model Builder	DMS	www.fico.com
ModelMAX	SOL	www.asacorp.com/products/mmrossover.jsp

Bảng 10.3.b Các công cụ khai phá dữ liệu thương mại điển hình (tiếp) [MR11].

Công cụ	Kiểu	Chia sẻ trang web
Molegro Data Modeler	SOL	www.molegro.com
NAG Data Mining Components	LIB	www.nag.co.uk/numeric/LDR/LDRdescription.asp
NeuralWorks Predict	SPEC	www.neuralware.com/products.jsp
Neurofusion	LIB	www.alyuda.com
Neuroshell	SPEC	www.neuroshell.com
Oracle Data Mining (ODM)	DMS	www.oracle.com/technology/products/bi/odm/index.html
Partek Discovery Suite	DMS	www.partek.com/software
Partek Genomics Suite	SOL	www.partek.com/software
PolyAnalyst	DMS	www.megaputer.com/polyanalyst.php
PolyVista	BI	www.polyvista.com
Random Forests	SPEC	www.salford-systems.com
RapAnalyst	SPEC	www.raptorinternational.com/rapanalyst.html
R-PLUS	MAT	www.experience-rplus.com
SAP Netweaver Business Warehouse (BW)	BI	www.sap.com/platforms/netweaver/components/businesswarehouse
SAS Enterprise Miner	DMS	www.sas.com/products/miner
See5	SPEC	www.railquest.com
SPAD Data Mining	DMS	eng.spadsoft.com
SQL Server Analysis Services	DMS	www.microsoft.com/sql
STATISTICA	DMS	www.statsoft.com/products/data-mining-solutions/G259
SuperQuery	DMS	www.azany.com
Teradata Database	BI	www.teradata.com
Think Enterprise Data Miner (EDM)	DMS	www.thinkanalytics.com
TIBCO Spotfire	DMS	spotfire.tibco.com
Unica PredictiveInsight	DMS	www.unica.com
WizRule und WizWhy	SPEC	www.wizsoft.com
X Affinity	SPEC	www.exclusivecore.com

- Công cụ nghiên cứu (RES) thực hiện một (hoặc rất ít) thuật toán mới và sáng tạo, vì vậy, chúng thường chưa ổn định. Hầu hết RES là mã nguồn mở. Trong RES, hỗ trợ đồ họa, vào-ra dữ liệu và tự động hóa ít được quan tâm.

Bảng 10.4. Các công cụ khai phá dữ liệu nguồn mở điển hình [MR11].

Công cụ	Kiểu	Chỉ dẫn trang web
ADaM*	LIB	datamining.itsc.uah.edu/adam
CellProfilerAnalyst	SOL	www.cellprofiler.org/index.htm
D2K*	DMS	alg.nesa.uiuc.edu
Gait-CAD	INT	sourceforge.net/projects/gait-cad
GATE	SOL	gate.ac.uk/download
GIFT	RES	www.gnu.org/software/gift
Gnome Data Mine Tools	DMS	www.togaware.com/datamining/gdatamine
Himalaya	RES	himalaya-tools.sourceforge.net
ImageJ	SOL	rsbweb.nih.gov/ij
ITK	SOL	www.itk.org
JAVA Data Mining Package	LIB	sourceforge.net/projects/jdmp
JavaNNS	SPEC	www.ra.cs.uni-tuebingen.de/software/JavaNNS/welcome.html
KEEL	INT	www.keel.es
Kepler	MAT	kepler-project.org
KNIME	INT	www.knime.org
LibSVM	LIB	www.csie.ntu.edu.tw/~cjlin/libsvm
MEGA	SOL	www.megasoftware.net/m_distance.html
MLC++	LIB	www.sgi.com/tech/mlc
Orange	LIB	www.ailab.si/orange
Pegasus	RES	www.cs.cmu.edu/~pegasus
Pentaho	BI	sourceforge.net/projects/pentaho
Proximity	SPEC	kdl.cs.umass.edu/proximity/index.html
PRTools	EXT	www.prtools.org
R	MAT	www.r-project.org
RapidMiner	DMS	www.rapidminer.com
Rattle	INT	rattle.togaware.com
ROOT	LIB	root.cern.ch/root
ROSETTA	SPEC	www.leb.uu.se/tools/rosetta/index.php
Rseslibs	RES	logic.mimuw.edu.pl/rses
Rule Discovery System*	SPEC	www.compumine.com
RWEKA	INT	cran.r-project.org/web/packages/RWeka/index.html
TANAGRA	INT	eric.univ-lyon2.fr/~ricco/tanagra/en/tanagra.html
Waffles	LIB	waffles.sourceforge.net
WEKA	DMS, LIB	sourceforge.net/projects/weka
XELOPES Library*	LIB	www.prudsys.de/en/technology/xelopes
XLMiner*	EXT	www.resample.com/xlminer

- Giải pháp (SOL) là một nhóm công cụ tùy chỉnh và hỗ trợ rất tốt cho một miền ứng dụng hẹp chẳng hạn như khai phá văn bản (GATE), xử lý hình ảnh (ITK, ImageJ), phát hiện ma túy (Molegro Data Modeler), phân tích hình ảnh trong kính hiển vi (CellProfilerAnalyst), hoặc khai phá dữ liệu hồ sơ biểu hiện gen (Partek Genomics Suite, MEGA). Hiện có rất nhiều SOL thương mại và nguồn mở.

Bảng 10.2 chỉ ra mối quan hệ giữa các kiểu công cụ khai phá dữ liệu với các nhóm người dùng. Hệ thống khai phá dữ liệu tỏ ra

hữu dụng cho ba lớp người dùng ứng dụng kinh doanh, ứng dụng nghiên cứu và dạy-học.

Bảng 10.3 (a,b) liệt kê các công cụ khai phá dữ liệu thương mại điển hình. Hai bảng này cung cấp tên công cụ, kiểu công cụ và chỉ dẫn trang web của công cụ khai phá dữ liệu.

Bảng 10.4 cung cấp một danh sách các công cụ khai phá dữ liệu mã nguồn mở với bốn công cụ phổ dụng nhất là ITK, KMINE, Orange, và WEKA. Tài liệu mô tả và hướng dẫn sử dụng công cụ là có sẵn tại trang web của mỗi công cụ.

10.3.3. Tập ví dụ đánh giá công cụ nghiên cứu

So sánh một thuật toán khai phá dữ liệu mới với các thuật toán cùng giải một bài toán cần phải được tiến hành trên tập dữ liệu miền ứng dụng hoặc một tập dữ liệu "đại diện" cho tập dữ liệu miền ứng dụng. Việc thu thập và gán nhãn dữ liệu là một công việc tốn nhiều công sức, hơn nữa, việc chứng tỏ tập dữ liệu xây dựng được đảm bảo tính "đại diện" cho dữ liệu miền ứng dụng lại là một bài toán khó. Thừa kế và phát triển các bộ dữ liệu được cộng đồng nghiên cứu thừa nhận là một tiếp cận tốt để có được các bộ dữ liệu mẫu cho quá trình xây dựng và đánh giá mô hình của thuật toán mới được đề xuất.

Với mỗi lớp bài toán, cộng đồng nghiên cứu thừa nhận có một số CSDL liệu mẫu được sử dụng để hỗ trợ việc đánh giá thuật toán mới. Kho chứa dữ liệu của nhóm học máy tại University of California, Irvine (UC Irvine Machine Learning Repository) là một ví dụ điển hình.

UC Irvine Machine Learning Repository được thừa nhận rộng rãi như một tập các CSDL mẫu dùng để đánh giá thuật toán học máy³⁰. Bảng 10.5 chỉ dẫn danh mục một số tập dữ liệu mẫu UCI (cột trái) và những tập dữ liệu được truy cập nhiều nhất (cột phải).

³⁰ <http://archive.ics.uci.edu/ml/>

Bảng 10.5. Một số tập dữ liệu mẫu trong kho chứa UCI

Newest Data Sets:		Most Popular Data Sets (hits since 2007):	
2012-07-04:		350364:	
2012-06-22:		248890:	
2012-06-09:		218320:	
2012-05-21:		178823:	
2012-04-25:		164309:	
2012-04-10:		138998:	
2012-02-14:		124793:	
2011-11-28:		104044:	
2011-11-28:		91119:	
2011-11-06:		89580:	
2011-10-18:		88411:	
2011-08-13:		89082:	

10.4. KHUYNH HƯỚNG PHÁT TRIỂN CỦA KHAI PHÁ DỮ LIỆU

Theo Ralf Mikut và Markus Reischl [MR11], thuật ngữ "data mining" lần đầu tiên xuất hiện vào năm 1983 trong bài báo của M. C. Lovell (M. C. Lovell (1983). Data Mining, *The Review of Economics and Statistics* 65:1-12) và thực sự được phát triển từ cuối những năm 1980. Trải qua khoảng 30 năm quá trình phát triển, khai phá dữ liệu không những trở thành một lĩnh vực khoa học-công nghệ rất rộng lớn mà vẫn luôn là nội dung nghiên cứu thời sự và đang được phát triển rất mạnh mẽ.

Hiệp hội các nhà khoa học về phát hiện tri thức và khai phá dữ liệu (The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining, viết tắt là SIGKDD) được thành lập và hoạt động. Ban điều hành của SIGKDD gồm một số nhà khoa học hàng đầu thế giới về lĩnh vực này do Piatetsky-Shapiro³¹ chủ trì. Từ năm 1995, hoạt động điển

³¹ <http://www.kdnuggets.com/gps.html>

hình nhất của SIGKDD là tổ chức Hội nghị khoa học quốc tế thường niên ACM SIGKDD Conference on Knowledge Discovery and Data Mining.

Là một thành phần năng động của khoa học máy tính cho nên khuynh hướng phát triển của khai phá dữ liệu có mối liên hệ mật thiết với khuynh hướng phát triển của khoa học máy tính.

10.4.1. Khuynh hướng phát triển của khoa học máy tính

Trong [Hop11], John E. Hopcroft trình bày về khuynh hướng phát triển của khoa học máy tính, bao gồm sự chuyển đổi các chủ đề của khoa học máy tính theo thời gian. Các chủ đề khoa học máy tính nổi bật đã chuyển đổi từ các chủ đề như Ngôn ngữ lập trình, Chương trình dịch, Hệ điều hành, Thuật toán, Cơ sở dữ liệu... tới các chủ đề như Theo dõi dòng tư tưởng trong tài liệu khoa học, Theo dõi quá trình tiến hóa của các cộng đồng trong các mạng xã hội, Trích xuất thông tin từ các nguồn dữ liệu phi cấu trúc, Xử lý các bộ dữ liệu và dòng dữ liệu đồ sộ, Trích xuất các tín hiệu từ tiếng ồn, Xử lý dữ liệu nhiều chiều và giảm kích thước...

Khuynh hướng chuyển đổi nói trên đối với các chủ đề nổi bật của khoa học máy tính cũng không nằm ngoài xu thế tăng trưởng với tốc độ cao khối lượng dữ liệu, đặc biệt là thành phần do người dùng tạo ra (UGC) như đã giới thiệu ở Chương 1. Trong nghiên cứu nói trên, J. E. Hopcroft giới thiệu một số nội dung lý thuyết cần được quan tâm để làm nền tảng khoa học giải quyết các bài toán thi hành xã hội điện tử như sau:

- Lý thuyết, mô hình và giải pháp tìm kiếm. Thứ nhất, câu hỏi tìm kiếm đã có sự thay đổi về chất từ câu hỏi mang tính cụ thể, thống kê sang câu hỏi mang tính tư vấn và đòi hỏi sự phân tích phức hợp như "Với tôi, mua ô tô loại nào là thích hợp?", "Hãy xây dựng một lịch sử có chú giải về lý thuyết đồ thị", "Tôi nên vào trường đại học nào?", "Các lĩnh vực của khoa học máy tính đã phát triển như thế nào?..." Thứ hai, không gian tìm kiếm là rộng lớn và câu hỏi được đặt ra mọi lúc, mọi nơi.

- Mạng và cảm biến. Trong một môi trường có tính sẵn sàng theo không gian và thời gian, hoạt động có tính ngẫu nhiên, giao tiếp với môi trường thông qua các cảm biến và kết nối mạng các mức thành phần (mức cảm biến, mức mạng các mạng con, mức các thành phần lớn và cực lớn...) cần được mô hình hóa với các giải pháp tích hợp hiệu quả.

- Xử lý dữ liệu nhiều chiều đồ sộ và chứa nhiều nhiễu. Tính đồ sộ của dữ liệu nằm trong xu thế bùng nổ thông tin như đã biết. Dữ liệu cần có nhiều chiều để biểu diễn sát thực hơn về thực tại. Tính ngẫu nhiên cùng với tính phức tạp của hệ thống dẫn đến việc dữ liệu có thể có chứa nhiều nhiễu.

- Mô hình và giải pháp tích hợp hệ thống và tài nguyên dữ liệu. Dù sử dụng phương pháp xây dựng hệ thống nào (chức năng, đối tượng, khác, và kết hợp) thì cách tiếp cận dựa trên thành phần đã trở thành cách tiếp cận chung, rất hữu hiệu đặc biệt là đối với các hệ thống lớn.

Một trong những mô hình toán học điển hình nhất liên quan tới các nội dung lý thuyết nêu trên là đồ thị lớn. Một ví dụ đơn giản là đồ thị Web được đề cập trong các máy tìm kiếm hiện nay đã có số đỉnh lên tới hàng tỷ nút. Tính sẵn sàng, mọi lúc, mọi nơi đòi hỏi mô hình hệ thống được thiết lập dưới dạng đồ thị sẽ có số nút rất lớn. Hơn nữa, các đồ thị lớn này cần là các đồ thị ngẫu nhiên. Lời giải cho các đồ thị lớn hiện nhận được sự quan tâm đặc biệt.

10.4.2. Khuynh hướng phát triển của khai phá dữ liệu

[Http://www.sigkdd.org/index.php](http://www.sigkdd.org/index.php) và <http://www.kdnuggets.com/> là hai trang web cung cấp nhiều kết quả nghiên cứu và triển khai cập nhật nhất về lĩnh vực phát hiện tri thức từ dữ liệu, là nguồn dữ liệu tiềm ẩn các thông tin hữu ích về khuynh hướng phát triển của lĩnh vực này mà chúng ta có thể “phát hiện” ra.

Theo Jiawei Han và cộng sự [HKL12], xu hướng phát triển nghiên cứu và triển khai điển hình về khai phá dữ liệu bao gồm:

- Phát triển một lý thuyết thống nhất về khai phá dữ liệu. Như đã được trình bày, lĩnh vực khai phá dữ liệu được ứng dụng

rộng rãi, nhận được sự quan tâm của đông đảo các nhà khoa học thuộc các lĩnh vực nghiên cứu rất đa dạng, vì vậy trình độ phát triển hiện thời của mỗi một nghiên cứu về khai phá dữ liệu lại mang tính quá đặc thù. Rất nhiều kỹ thuật được thiết kế cho các bài toán riêng lẻ, chẳng hạn như phân lớp hoặc phân cụm, mà không có một cơ sở lý thuyết thống nhất.

- Mở rộng miền ứng dụng khai phá dữ liệu cả về bề rộng và chiều sâu (không gian-thời gian, đối tượng di chuyển và hệ thống mạng vật lý, dữ liệu đa phương tiện khai phá, văn bản và web; dữ liệu sinh học và y sinh; hình ảnh và âm thanh; mạng xã hội và mạng thông tin). Phát triển các ứng dụng khai phá dữ liệu được mở rộng tới thương mại điện tử, tiếp thị điện tử và trở thành trào lưu trong dịch vụ bán lẻ, đồng thời, được tăng cường sử dụng trong nhiều lĩnh vực khác như phân tích tài chính, viễn thông, sinh dược phẩm và các ngành khoa học. Xu thế trình độ kinh tế tri thức của xã hội ngày càng được tăng cường là tiền đề cho việc mở rộng miền ứng dụng của khai phá dữ liệu.

- Phát triển các phương pháp khai phá dữ liệu có tính khả cõi và tương tác, phát triển các phương pháp thăm dò. Sự tăng trưởng khôi lượng các dữ liệu có rất nhiều chiều và dòng dữ liệu tốc độ cao. Phù hợp với sự bùng nổ thông tin và nhu cầu phát triển ứng dụng khai phá dữ liệu, việc đề xuất các thuật toán khai phá dữ liệu có chức năng tự tương tác và tương tác lẫn nhau đã có tính bản chất. Trong một số ứng dụng, chẳng hạn trong khai phá text hoặc phân tích an toàn hệ thần kinh, số chiều của dữ liệu lên tới từ hàng trăm triệu tới hàng tỷ đặc trưng. Trong một số ứng dụng khác, chẳng hạn trong các bài toán nghiên cứu về thiên văn hoặc về mạng máy tính, dòng dữ liệu là rất lớn (có thể lên tới hàng trăm TB tại thời điểm hiện nay). Công nghệ khai phá dữ liệu hiện tại vẫn quá chậm để chủ động được đối với các dữ liệu lớn như vậy. Mặt khác, khai phá dữ liệu dựa trên ràng buộc là một định hướng quan trọng nâng cao năng lực tổng thể của quá trình khai phá dữ liệu có sự tăng cường tương tác với người sử dụng.

- Phát triển các mô hình và phương pháp tích hợp khai phá dữ liệu vào các hệ thống CSDL, hệ thống kho dữ liệu, hệ thống tìm kiếm, hệ thống tính toán đám mây. Các hệ thống này đã trở thành trào lưu của các hệ thống xử lý thông tin. Chẳng hạn, bài toán tích hợp Web với kho dữ liệu bao gồm nhiều nội dung của khai phá nội dung Web để xây dựng được kho dữ liệu với nguồn dữ liệu giàu có của Web. Vấn đề quan trọng khi tích hợp khai phá dữ liệu ở đây phải đảm bảo rằng các phục vụ khai phá dữ liệu được coi là các thành phần phân tích dữ liệu bản chất của hệ thống cần phải được tích hợp một cách trơn tru với môi trường xử lý thông tin.

- Chuẩn hóa quá trình phát hiện tri thức, chuẩn hóa các ngôn ngữ khai phá dữ liệu cùng với các phương tiện chuẩn hóa khác làm thuận tiện hơn việc phát triển có tính hệ thống các giải pháp khai phá dữ liệu tính liên thao tác của các hệ thống và chức năng khai phá dữ liệu phức hợp [AGHHL07]. Một số kết quả ở mức sản phẩm công nghệ điển hình theo hướng này có OLE DB (*Object Linking and Embedding, Database*) dùng cho khai phá dữ liệu của MicroSoft, PMML (*Predictive Model Markup Language*) của Data Mining Group (DMG) và CRISP-DM (*CRoss Industry Standard Process for Data Mining*) của nhóm phát triển CRISP-DM (<http://www.crisp-dm.org/>).

- Khai phá dữ liệu động, không cân bằng và nhạy cảm về chi phí. Mô hình khai phá dữ liệu cần gắn kết với thời gian vì dữ liệu là không tĩnh và thay đổi theo thời gian. Theo cách thông thường, mô hình được học cần phù hợp theo thời gian, khi có dữ liệu hiện thời cần học tiếp mô hình cho các khai phá tiếp theo, có nghĩa là mô hình cũng có tính xu hướng. Một khuynh hướng của khai phá dữ liệu là mô hình được xây dựng bao hàm được tính xu hướng càng nhiều càng tốt. Tương tự về khai phá dữ liệu đối với dữ liệu không cân bằng, nhạy cảm về chi phí.

- Khai phá dữ liệu trong một khung cảnh mạng, trong đó có các mạng xã hội trực tuyến hoặc các mạng máy tính (khai phá dữ liệu tốc độ cao đối với dòng dữ liệu tốc độ cao). Liên quan mật thiết tới khai phá dữ liệu trong khung cảnh mạng là các bài toán khai

phá dữ liệu phân tán và khai phá dữ liệu đa tác tử cũng như khai phá dữ liệu liên quan tới các quá trình, luồng dữ liệu thời gian thực.

- Tăng cường tính trực quan hóa trong khai phá dữ liệu là giải pháp hiệu quả nhằm làm cho quá trình phát hiện tri thức từ tập dữ liệu đồ sộ được thi hành bằng các bộ công cụ trực quan hóa và dễ dàng tích hợp được với các thành phần khai phá dữ liệu.

- Bảo vệ tính riêng tư và an ninh thông tin.

Thông báo của các hội nghị KDD thế giới gần đây (KDD-2010, Washington DC, July 25-28; <http://www.kdd.org/kdd/2010/>, KDD-2011, San Diego CA, August 21-24, 2011; <http://www.kdd.org/kdd/2011>), và KDD-2012, Beijing-China, August 12-16, 2012; <http://www.kdd.org/kdd2012/> đã phản ánh cụ thể hơn cho các khuynh hướng nghiên cứu và triển khai nói trên:

- *Về nghiên cứu:* Mô tả việc nghiên cứu sáng tạo trên mọi khía cạnh của phát hiện tri thức và khai phá dữ liệu theo các chủ đề về phương pháp phân lớp và hồi quy, học bán giám sát, phân cụm, lựa chọn đặc trưng, các mạng xã hội, khai phá dữ liệu đồ thị, phân tích dữ liệu thời gian và không gian, tính mở rộng, sự riêng tư, trực quan hóa, phân tích văn bản, khai phá Web, hệ thống tư vấn, v.v. Mảng nghiên cứu cần nhấn mạnh cơ sở lý thuyết cho các tiếp cận mới lạ về mô hình và phương pháp thuật toán cho bài toán khai phá dữ liệu cụ thể trong khoa học, kinh doanh, y tế, và các ứng dụng kỹ thuật v.v.

- *Về triển khai:* Mô tả việc triển khai các giải pháp KDD có liên quan tới việc thiết lập công nghiệp hoặc chính quyền. Nhấn mạnh việc thúc đẩy sự hiểu biết thực tiễn, áp dụng, hoặc các vấn đề thực tế liên quan đến việc sử dụng các công nghệ KDD trong công nghiệp, chính quyền và làm nổi bật các thách thức nghiên cứu mới phát sinh từ nỗ lực để tạo ra các ứng dụng KDD thực tế. Miền ứng dụng bao gồm thương mại điện tử, y tế và dược phẩm, quốc phòng, chính sách công, kỹ nghệ, sản xuất, viễn thông, và chính phủ v.v.

Sự phong phú về khuynh hướng phát triển của khai phá dữ liệu là minh chứng rõ ràng cho sự phát triển mạnh mẽ của lĩnh vực này. Khai phá dữ liệu phương tiện xã hội (data mining in social media) và học máy không dừng (non-ending learning) là những chủ đề nghiên cứu nổi bật trong thời gian gần đây.

10.4.2.1. Khai phá dữ liệu phương tiện xã hội

Nội dung do người dùng tạo ra (UGC) đã trở thành bộ phận chiếm trọng số lớn tăng trưởng khối lượng dữ liệu (Chương 1) là nền tảng cho sự phát triển nhanh chóng của khai phá dữ liệu phương tiện xã hội (social media).

Bảng 10.6. Phân loại mạng xã hội theo hiện diện xã hội/phong phú phương tiện truyền thông (social presence/media richness) và tự trình bày/ tự tiết lộ (self-presentation / self-disclosure) [HK10]

		Social presence/ Media richness		
		Low	Medium	High
		High	Blogs Social networking sites (e.g., Facebook)	Virtual social worlds (e.g., Second Life)
Self-presentation/ Self-disclosure	High	Low	Collaborative projects (e.g., Wikipedia)	Content communities (e.g., YouTube)
				Virtual game worlds (e.g., World of Warcraft)

Theo Andreas M. Kaplan và Michael Haenlein [KH10], thuật ngữ "phương tiện xã hội" được hiểu là "một nhóm các ứng dụng dựa trên Internet được xây dựng trên nền tảng tư tưởng và công nghệ của Web 2.0 cho phép tạo và trao đổi nội dung do người dùng tạo ra". Theo các tác giả, thời đại của phương tiện xã hội được bắt đầu từ việc ra đời của trang web "Open Diary"³² (Nhật ký mở) của Bruce và Susan Abelson (vào tháng 5/2012, Open Diary có trên 381 nghìn nhật ký mở). Hai chiều đặc trưng cơ bản phân biệt các loại phương tiện xã hội là hiện diện xã hội/phong phú phương tiện truyền thông (social presence/media richness) và tự trình bày/tự tiết lộ (self-presentation / self-

³² <http://www.opendiary.com/>

disclosure). Các tác giả giải thích chi tiết về nội dung ngữ nghĩa của hai chiều đặc trưng này. Bảng phân loại các phương tiện xã hội theo hai chiều đặc trưng nói trên đã được đưa ra. Theo Jure Leskovec [Lesk11], phương tiện xã hội được thiết kế để phổ biến thông qua tương tác xã hội. Phương tiện xã hội được thi hành bằng các mạng xã hội trực tuyến đã tạo nên nguồn dữ liệu về đời sống xã hội loài người.

Chúng ta dùng thuật ngữ khai phá dữ liệu phương tiện xã hội để chỉ các nghiên cứu và triển khai khai phá dữ liệu từ phương tiện xã hội và từ mạng xã hội trực tuyến, nội dung do người dùng tạo ra... do mối liên quan chặt chẽ của chúng với phương tiện xã hội. Dữ liệu phương tiện xã hội trải trên một miền rộng lớn các lĩnh vực trong đời sống xã hội, đặc biệt chúng phản ánh tính "hiện thời" của đời sống cho nên khai phá dữ liệu phương tiện xã hội còn là nội dung chủ yếu của "phân tích cuộc sống" (living analytics³³). Có thể nói khai phá dữ liệu phương tiện xã hội hội tụ những nội dung thời sự nhất về mạng xã hội, về khai phá dữ liệu, về tiếp thị và kinh doanh, về hành vi con người...

Rất nhiều công trình nghiên cứu về khai phá dữ liệu phương tiện xã hội đã và sẽ được công bố. David Easley và Jon Kleinberg [EK10], Jiawei Han và cộng sự [HSYY10], Jure Leskovec [Lesk11], David Easley và Jon Kleinberg [EK10] cung cấp các khía cạnh khác nhau của một khung nhìn tổng thể về khai phá dữ liệu phương tiện xã hội bao gồm các khái niệm và nội dung về phương tiện xã hội, ý nghĩa kinh tế và xã hội của nghiên cứu phương tiện xã hội.

Hai kiểu đối tượng nghiên cứu chính trong khai phá dữ liệu phương tiện xã hội là nội dung phương tiện xã hội và cấu trúc phương tiện xã hội (mạng xã hội). Khai phá dữ liệu nội dung phương tiện xã hội để chỉ hoạt động khai phá dữ liệu nội dung văn bản mà người dùng tạo ra trên phương tiện xã hội. Thành phần này tạo thành một miền ứng dụng rất rộng lớn. Khai phá dữ liệu

³³ <http://www.larc.smu.edu.sg/>

cấu trúc phương tiện xã hội để chỉ hoạt động khai phá dữ liệu về cấu trúc mạng xã hội tương ứng với phương tiện xã hội. Hơn nữa, khai phá dữ liệu cũng được tiến hành dựa trên sự kết hợp nội dung và cấu trúc trong phương tiện xã hội. Phương tiện xã hội là một cách thức mà người dùng bất kỳ trong xã hội đều có thể chia sẻ và đóng góp nội dung, bày tỏ quan điểm và kết nối với những người khác, vì vậy phương tiện xã hội mang hơi thở của cuộc sống đời thường đang diễn ra với tính động cao.

Khai phá dữ liệu nội dung phương tiện xã hội để cập tới toàn bộ nội dung của hai lớp bài toán khai phá dữ liệu mô tả và dự báo; nó huy động một phạm vi toàn diện các thuật toán khai phá dữ liệu [HSYY10, Lesk11]. Khai phá dữ liệu nội dung phương tiện xã hội có một phạm vi ứng dụng rất rộng lớn trong quản lý danh tiếng (reputation management), tiếp thị phương tiện xã hội (Social media marketing), phản ứng công dân (citizen response), phân tích hành vi con người (Human behavior analysis), phóng viên công dân thời gian thực (Real time citizen journalist) và rất nhiều ứng dụng khác.

Chẳng hạn, Craig Macdonald và cộng sự [MSOS10] cho một phân tích về các nghiên cứu khai phá dữ liệu blogs trong khuôn khổ TREC giai đoạn 2006-2009 đối với ba bài toán: phát hiện quan điểm (opinion-finding) đối với một đối tượng đã cho (Người sử dụng blogs nghĩ gì về đối tượng X đã cho?), chưng cất blog (blog distillation) để tìm ra các blog quan tâm tới đối tượng X (Tìm blog quan tâm chính, định kỳ tới X?) và phát hiện tin nổi bật (top news) từ blogs (tìm các tin có giá trị gần đây nhất?). Hàng chục công trình nghiên cứu tham gia TREC-Blogs Track cung cấp một phổ rộng lớn các giải pháp khai phá dữ liệu để giải quyết ba bài toán nói trên. Theo các tác giả, bài toán phát hiện quan điểm (bài toán đầu tiên) từ blogs không có nhiều khác biệt so với bài toán khai phá quan điểm nói chung.

Xem xét một nghiên cứu khác về khai phá dữ liệu microblogs. Đặc thù về độ dài thông điệp ngắn, về cấu trúc liên kết thành viên và thông điệp tạo ra một số yếu tố bổ sung cho khai

phá dữ liệu nội dung từ microblogs. Tính cập nhật thông tin nhanh trên microblogs là tiền đề cho các giải pháp dự báo ngắn hạn trên microblogs. Trong [MCB11], Huina Mao và cộng sự cung cấp một khảo sát công phu về các chỉ số tâm trạng của nhà đầu tư chứng khoán gồm có tỷ lệ phần trăm tăng giá DSI (DSI bullish percentage: DSI), chỉ số thông minh của nhà đầu tư (Investor Intelligence: II), đánh giá nhà đầu tư Twitter (Twitter Investor Sentiment: TIS), lượng thuật ngữ tìm kiếm tài chính Tweet (Tweet volumes of financial search terms: TV-FST), đánh giá tin tức tiêu cực (Negative News Sentiment: NNS), và lượng tìm kiếm Google của các thuật ngữ tài chính (Google search volumes of financial search terms: GIS). Qua thực nghiệm theo thời gian một tuần, các tác giả phát hiện rằng GIS có độ liên quan đáng kể với các chỉ số tài chính phân biệt (different financial indexes: DJIA) và như vậy GIS có thể thay thế các chỉ số dự báo tài chính. Tuy nhiên, đối với chỉ số thông minh nhà đầu tư (II) thì không có được vai trò đó. Độ chính xác của dự báo có thể cải thiện khi làm giàu đặc trưng. Thực nghiệm theo thời gian ngày cho thấy TIS và TV-FST cho phép dự báo tốt đáng kể theo thống kê về hoàn vốn thị trường hàng ngày trong khi DSI thì không cho phép. NNS cũng cho kết quả theo chiều hướng tương tự như TSI và TV-FST nhưng kém hơn về độ liên quan.

Khai phá dữ liệu cấu trúc phương tiện xã hội để cập tới mẫu và tính động của cấu trúc phương tiện xã hội. Mẫu cấu trúc của một phương tiện xã hội phù hợp với tính chất chung của mạng xã hội và đặc tính riêng của phương tiện xã hội đó.

Tính chất chung của mạng xã hội gồm tính chất thế giới nhỏ (small world), liên kết mạnh – yếu (strong – weak tie), phân bố luật lũy thừa (power law distribution, cấu trúc cộng đồng (community)). Tính chất thế giới nhỏ chỉ ra rằng độ dài đường đi liên kết hai đỉnh bất kỳ trong mạng xã hội không vượt quá một số nguyên dương nhỏ. Tính chất này được Stanley Milgram phát hiện từ thực nghiệm vào năm 1969. Tính chất liên kết mạnh – yếu chỉ ra rằng liên kết giữa hai nút trong mạng xã hội không giống nhau và được chia thành hai lớp liên kết mạnh và liên kết yếu. Về mặt

xã hội, liên kết mạnh thể hiện mối quan hệ người thân, còn liên kết yếu thể hiện mối quan hệ mới tiếp xúc. Trong nhiều trường hợp, liên kết mạnh - yếu còn được chuyển đổi thành liên kết dương - âm để chỉ mối liên kết đồng thuận hoặc trái ngược nhau. Phân bố luật lũy thừa (power law distribution): số nút có k liên kết tới bằng khoảng $1/k^2$ với số $k > 2$, cấu trúc cộng đồng (community): tập tất cả các nút có thể được phân chia thành một số nhóm các nút có tính chất chung.

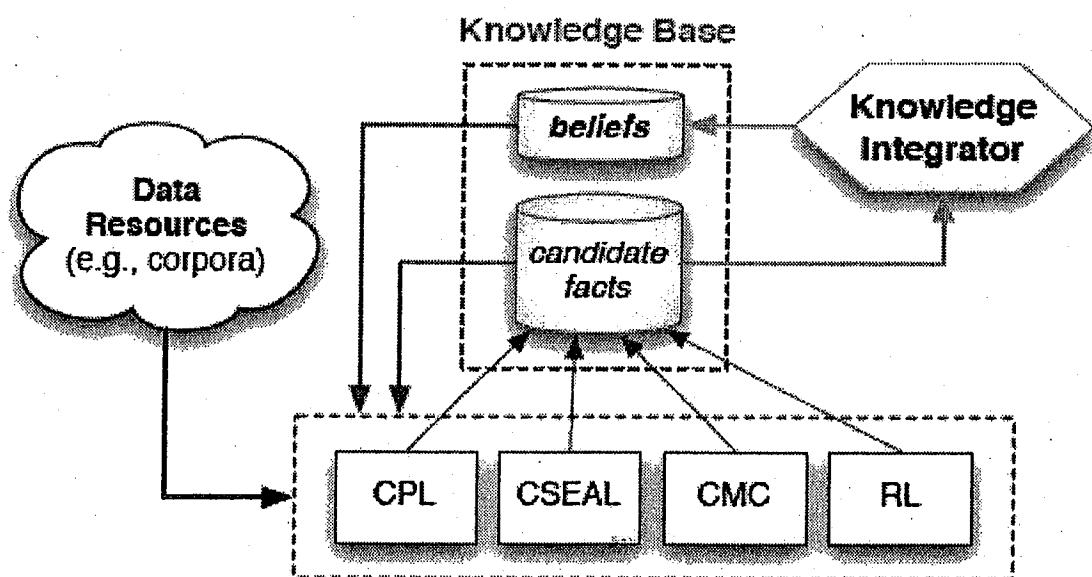
Dự báo liên kết là một bài toán quan trọng trong khai phá dữ liệu cấu trúc phương tiện xã hội. L. Liu và T. Zhou [LZ10] cung cấp một tổng quan về dự báo liên kết trong mạng xã hội. Cho đồ thị vô hướng mạng xã hội $G = (V, E)$ trong đó V là tập n đỉnh, E là tập cung đã có trong tập U gồm tất cả $n^*(n-1)/2$ các cung có thể có. Bài toán đặt ra là tìm ra các cung có thể có trong tương lai từ tập $U \setminus E$. Các tác giả hệ thống hóa các thuật toán giải quyết bài toán dự báo liên kết, bao gồm các thuật toán dựa trên độ tương tự, các thuật toán dựa theo cực đại khả năng, các thuật toán dựa trên mô hình xác suất. J. Leskovec và cộng sự [LHK10] đề xuất phương pháp học máy hồi quy để dự báo liên kết âm – dương trong mạng xã hội. Số lượng mẫu quan hệ giữa các liên kết được thu gọn dựa trên lý thuyết cân bằng (balance theory) và lý thuyết trạng thái (status theory) [EK10], vì vậy, mô hình học máy dự báo hồi quy thi hành hiệu quả hơn.

10.4.2.2. Học máy không dừng

Kỳ vọng về một hệ thống học máy làm được như con người "học suốt đời, trở nên học tốt hơn qua thời gian" xuất hiện từ những năm 1980, tuy nhiên, các kết quả nghiên cứu về học máy không dừng (never-ending learning) chưa được như kỳ vọng. Thời gian gần đây, một vài nhóm nghiên cứu, trong đó có nhóm nghiên cứu tại Carnegie Mellon University (Tom M. Mitchell và cộng sự) là một trong những nhóm đi tiên phong về chủ đề và đã công bố một số kết quả nghiên cứu đáng chú ý.

Lý tưởng hóa, học máy không dừng là học cách học để đánh giá, chọn lựa... mô hình giải quyết bài toán, mà không phải về học

trực tiếp mô hình giải quyết bài toán. Và như vậy có sự khác nhau về bản chất giữa học máy không dừng với học máy cải tiến mô hình dựa trên việc bổ sung dữ liệu hoặc tri thức miền ứng dụng (học tăng cường: reinforcement learning, học tích cực: active learning). Tuy nhiên, trên thực tế, tiếp cận học không dừng chưa đạt được mức lý tưởng mà ở mức là bước tiến mới của học tăng cường, học tích cực [MCCC10].



Hình 10.1. Kiến trúc một hệ thống học không dừng [CBKSH10]

Andrew Carlson và cộng sự [CBKSH10] cung cấp luận điểm chung về việc xây dựng các hệ thống học máy ngôn ngữ không dừng (Never-Ending Language Learner: NELL):

- Chỉ sử dụng các thành phần tạo ra lỗi không tương quan. Hệ thống bao gồm các thành phần con như vậy sẽ cho tỷ lệ lỗi thấp.
- Nhiều kiểu học các tri thức có liên quan nhau. Với các bộ học như vậy cho phép tạo các nguồn phức, độc lập nhau để tạo ra cùng một kiểu tri thức chân lý,
- Dùng các phương pháp học máy bán giám sát ghép cặp để hạn chế ràng buộc giữa các từ vị được học. Tạo thư mục phân cấp

(taxonomy) các lớp và các quan hệ để xác định được quan hệ cha-con, quan hệ loại trừ nhau giữa các lớp (quan hệ) để thuận tiện trong việc loại trừ ràng buộc giữa các vị từ học được.

- Phân biệt được đối tượng chân lý (belief) tin cậy cao trong cơ sở tri thức với các ứng viên tin cậy thấp. Giữ lại giải thích nguồn cho mỗi chân lý.

- Sử dụng một trình diễn cơ sở tri thức thống nhất để nắm bắt được các sự kiện ứng viên và chân lý được nâng cấp của mọi kiểu; dùng cơ chế chỉ dẫn và học phù hợp mà có thể thao tác được trên trình diễn dùng chung đó.

Dựa trên luận điểm chung đó, các tác giả đề xuất một mô hình thi hành NELL với 4 hệ thống thành phần (Hình 10.4):

- Bộ học mẫu ghép cặp (Coupled Pattern Learner: CPL): Một bộ trích xuất văn bản tiến hành học và sử dụng mẫu ngữ cảnh kiểu "mayor of X" và "X plays for Y" để trích xuất các thể hiện của các lớp và các quan hệ. CPL sử dụng thống kê đồng - xuất hiện cụm danh từ và mẫu ngữ cảnh (cả hai được xác định khi dùng dãy thẻ POS) để học trích xuất mẫu cho mỗi vị từ quan tâm và sau đó sử dụng các mẫu này để tìm các thể hiện bổ sung của mỗi vị từ.

- Coupled SEAL (CSEAL: Coupled Set Expander for Any Language): Một bộ trích xuất bán cấu trúc đặt truy vấn Internet với tập chân lý cho mỗi lớp hoặc quan hệ, và sau đó khai phá các danh sách và các bảng để trích xuất ra các thể hiện cho các vị từ tương ứng. CSEAL sử dụng các quan hệ loại trừ lẫn nhau để cung cấp các phản ví dụ, được dùng để lọc ra danh sách và các bảng quá chung chung.

- Các bộ phân lớp hình thái ghép cặp (Coupled Morphological Classifier: CMC): Một tập các mô hình phân lớp hồi quy logistic nhị phân L_2 (một mô hình cho một lớp) tiến hành phân lớp các cụm danh từ dựa vào các đặc trưng hình thái khác nhau (từ, viết hoa, phụ tố, các POS...). Chân lý từ cơ sở tri thức được dùng làm ví dụ học, nhưng mỗi CMC cần thực hiện lặp để có ít nhất 100 thể hiện bổ sung.

- Bộ học luật (Rule Learner: RL): Một bộ học luật theo thuật toán học quan hệ cấp 1 tương tự như thuật toán FOIL học luật Horn xác suất để nhận được các thể hiện mới của các quan hệ từ các thể hiện quan hệ có trong cơ sở tri thức.

Các tác giả đã tiến hành chạy thực nghiệm NELL và sau 67 ngày thi hành được 66 vòng lặp. Kết quả nhận được 242,453 chân lý mới tính theo mọi vị từ, 95% trong đó là thể hiện của lớp và 5% là thể hiện của quan hệ. NELL cho thấy sự tiến bộ đáng kể của quá trình hiện thực hóa các hệ thống học máy không dừng.

CÂU HỎI VÀ BÀI TẬP

- 10.1.** Hãy nhận diện trường hợp cần thiết phải triển khai dự án khai phá dữ liệu.
- 10.2.** Tính chất của dữ liệu cho bài toán khai phá dữ liệu.
- 10.3.** Tính chất của tri thức kết quả của quá trình khai phá dữ liệu.
- 10.4.** Đặc trưng của chuyên viên khai phá dữ liệu.
- 10.5.** Khai phá phương tiện xã hội.
- 10.6.** Khái niệm học không dừng và tiếp cận thi hành hệ thống học không dừng.

TÀI LIỆU THAM KHẢO

- [AGHHL07] Sarabjot Singh Anand, Marko Grobelnik, Frank Herrmann, Mark Hornick and Christoph Lingensfelder, et al. *Knowledge discovery standards*, Artificial Intelligence Review, 27 (1): 21-56, 2007.
- [ARA1] A.Rajaraman, J. D.Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2011.
- [AS00] Rakesh Agrawal, Ramakrishnan Srikant. *Privacy-Preserving Data Mining*, SIGMOD Conference 2000: 439-450, 2000.
- [BCGJ11] Francesco Bonchi, Carlos Castillo, Aristides Gionis, Alejandro Jaimes. *Social Network Analysis and Mining for Business Applications*, ACM TIST 2(3): 22, 2011.
- [BEF84] James C. Bezdek, Robert Ehrlich, William Full. *FCM: The fuzzy c-means clustering algorithm*, Computers & Geosciences, 10 (2–3, 1984): 191–203, 1984.
- [BLI1] B.Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, 2nd Edition, Springer.
- [Blum98] A. Blum and T. Mitchell. *Combining labeled and unlabeled data with co-training*. In COLT: Proceedings of the Workshop on Computational Learning Theory, pages 92-100, 1998.
- [BNGC00] Jeff Bowes, Eric Neufeld, Jim E. Greer, John Cooke. *A Comparison of Association Rule Discovery and Bayesian Network Causal Inference Algorithms to Discover Relationships in Discrete Data*, Canadian Conference on AI 2000: 326-336, 2000.

- [Branson02] S. Branson and A. Greenberg, *Clustering Web Search Results Using Suffix Tree Methods*, Final project report, 2002.
- [Brynjolfsson93] Brynjolfsson, Erik. *The productivity paradox of information technology*. Communications of the ACM36 (12): 66 –77, 1993.
- [BS02] Julian Birkinshaw and Tony Sheehan. *Managing the Knowledge Life Cycle*, Sloan Management Review, 44 (3): 75-83, 2002.
- [Carr03] Nicholas G. Carr. *IT does'n matter! HBR at Large*, 41-49, May 2003.
- [Carr05] Nicholas G. Carr. *The end of corporate computing*, MIT Sloan Management Review, Spring 2005: 67-73.
- [CBKSH10] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., Tom M. Mitchell. *Toward an Architecture for Never-Ending Language Learning*, AAAI 2010: 1306-1313, 2010.
- [CCG98] Kenneth Collier, Bernard Carey, Ellen Grusy, Curt Marjaniemi, Donald Sautter. *A Perspective on Data Mining, Technical Report*, Northern Arizona University, 1998.
- [CCKKR00] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer and Rÿdiger Wirth. *CRISP-DM 1.0: Step-by-step data mining guide*, The CRISP-DM consortium, August 2000.
- [CD05] Derek H. C. Chen and Carl J. Dahlman. *The Knowledge Economy, the KAM Methodology and World Bank Operations*, The World Bank, October 19, 2005.
- [CD10] Christophe Giraud Carrier, Margaret H. Dunham. *On the Importance of Sharing Negative Results*, ACM SIGKDD Explorations newsletter, 12(2): 3-4, 2010.
- [Chen07] Xiujuan Chen. *Computational Intelligence Based Classifier Fusion Models For Biomedical Classification Applications*, PhD Thesis, Georgia Stage University, USA, 2007.

- [Christopher08] C. D. Manning and P. Raghavan and H. Schütze, *An Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [CKV04] Chris Clifton, Murat Kantarcioglu and Jaideep Vaidya. *Defining Privacy for Data Mining, Next Generation Data Mining*, AAAI/MIT Press 2004.
- [Cui] X. Cui, T. E. Potok and Paul Palathingal, *Document Clustering using Particle Swarm Optimization*, IEEE Swarm Intelligence Symposium, The Westin, 2005.
- [Cutting93] Cutting, D. R., D. R. Karger, and J. O. Pedersen. *Constant interaction-timescatter/gather browsing of very large document collections*. In SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, 126-134. ACM Press, 1993.
- [CYZZ10] Longbing Cao, Philip S. Yu, Chengqi Zhang, Yanchang Zhao. *Domain Driven Data Mining*, Springer, 2010.
- [Dempster77] A. P. Dempster, N. M. Laird, & D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society, Series B, 39 (1), 1-38, 1977.
- [Deng10] Mina Deng. *Privacy Preserving Content Protection*, PhD Thesis, Katholieke Universiteit Leuven, 2010.
- [DHP06] D. Dubois, E. Hullermeier, H. Prade. *A systematic approach to the assessment of fuzzy association rules*, Data Mining and Knowledge Discovery, 13(2): 1–26, 2006.
- [DMSV03] Miguel Delgado, Nicolás Marín, Daniel Sánchez, and María-Amparo Vila. *Fuzzy Association Rules: General Model and Applications*, IEEE Transactions On Fuzzy Systems, 11 (2): 214-225, April 2003.
- [DP90] D. Dubois and H. Prade. *Rough fuzzy sets and fuzzy rough sets*, International Journal of General Systems, 17:191-209, 1990.

- [EK10] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*, Cambridge University Press, 2010.
- [Elroy00] Mark W. McElroy. *The New Knowledge Management, Knowledge And Innovation*, Journal of the KMCI, 1(1): 43-67, October 15, 2000.
- [Elroy02] Mark W. McElroy. *Corporate Epistemology And The New Knowledge Management, Managing The Complex*: IV Conference, 2002.
- [EM03] L. Egghe, C. Michel. *Construction of weak and strong similarity measures for ordered sets of documents using fuzzy set techniques*. Information Processing and Management No 39, 771– 807, 2003.
- [FPS96] Fayyad, Piatetsky-Shapiro, Smyth. *From Data Mining to Knowledge Discovery: An Overview*. In Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press/ The MIT Press, Menlo Park, CA, 1-34, 1996.
- [Fried97] Jerome H. Friedman. *Data Mining and Statistics: What's the Connection?* Technical report, Department of Statistics and Stanford Linear Accelerator Center, Stanford Linear Accelerator Center, Stanford University, 1997.
- [Garry05] Ken McGarry. *A Survey of Interestingness Measures for Knowledge Discovery*, The Knowledge Engineering Review, 20(1): 39-61, March 2005.
- [GH06] Liqiang Geng and Howard J. Hamilton. *Interestingness Measures for Data Mining: A Survey*, ACM Computing Surveys, 38 (3), Article 9, 2006.
- [Gold10] Andrew Brian Goldberg. *New directions in semi-supervised learning*, PhD. Thesis, University of Wisconsin-Madison, 2010.

- [Goldman00] S. Goldman and Y. Zhou, *Enhancing Supervised Learning with Unlabeled Data*. Proceedings of ICML, pp. 327-334, 2000.
- [GP10] Martin McGrane, Simon K. Poon. *Interaction as an Interestingness Measure*, ICDM Workshops 2010: 726-731, 2010.
- [GR11] John Gantz and David Reinsel. *Extracting Value from Chaos*, A Report Sponsored by EMC Corporation, June 2011.
- [Grube09] C. Grube. *Measuring the Immeasurable* (Part I: Knowledge as a valuable resource , Part III: Patent valuation), Springer, 2009.
- [Guses10] Fahriye Seda Gurses. *Multilateral Privacy Requirements Analysis in Online Social Network Services*, PhD Thesis, Katholieke Universiteit Leuven, 2010.
- [GZ11] Xijing Ge and Jianming Zhu. *Privacy Preserving Data Mining* (New Fundamental Technologies in Data Mining: Chapter 29), INTECH, 2011.
- [Han06] J. Han and M. Kamber, *Data Mining-Concepts and Techniques*, Morgan Kaufmann, 2006.
- [Haw04] Brian L. Hawkins. *A Framework for the CIO Position*, Educause Review, 39(6) : 94–103, November/December 2004.
- [HF09] Yang Hang, Simon Fong. *A Framework of Business Intelligence-Driven Data Mining for E-business*, NCM 2009: 1964-1970, 2009.
- [HG09] Jiawei Han and Jing Gao. *Research Challenges for Data Mining in Science and Engineering* (Chapter 1 in “Next Generation of Data Mining”, Hillol Kargupta, Jiawei Han, Philip S. Yu, Rajeev Motwani, Vipin Kumar, editors), Chapman & Hall, 2009.
- [HGEK07] Xuan-Hiep Huynh, Fabrice Guillet, Julien Blanchard, Pascale Kuntz, Henri Briand, and Regis Gras. *A graph-based*

clustering approach to evaluate interestingness measures: a tool and a comparative study, Quality Measures in Data Mining, Fabrice Guillet, Howard J. Hamilton (Ed.), 25-50, 2007.

[Hiro06] Takeuchi Hirotaka. *The New Dynamism of the Knowledge-Creating Company*, In Japan Moving Toward a More Advanced Knowledge Economy: ^Advanced Knowledge: Creating Companies, by Takeuchi, Hirotaka and Tsutomu Shibata. Washington, D.C.: World Bank Institute (WBI), 2006.

[HK0106] J. Han and M. Kamber. *Data Mining-Concepts and Techniques*, Morgan Kaufmann, 2006.

[HKK97] Eui-Hong (Sam) Han, George Karypis, and Vipin Kumar. *Scalable Parallel Data Mining for Association Rules*. Department of Computer Science, University of Minnesota, 4-192 EECS Building, 200 Union St. SE, Minneapolis, MN 55455, USA, 1997.

[Hop10] John Hopcroft. *Computer Science Theory to support Research in the Information Age*, Seminar Report, University of Southern California, April 6, 2010.

[Hop11] John Hopcroft. *Computing and the Future*, Microsoft Latin American Faculty Summit, Catagena, May 18, 2011.

[HP03] Enrique Herrera-Viedma, Eduardo Peis. *Evaluating the informative quality of documents in SGML format from judgements by means of fuzzy linguistic techniques based on computing with words*. Inf. Process. Manage, 39(2): 233-249, 2003.

[Hsu02] C.W. Hsu and C.-J. Lin, *A comparison of methods for multi-class support vector machines*, IEEE transactions on Neural Networks, vol. 13, pp. 415-425, 2002.

[HSYY10] Jiawei Han, Yizhou Sun, Xifeng Yan, Philip S. Yu. *Mining Knowledge from Databases: An Information Network Analysis Approach*, ACM SIGMOD Conference Tutorial, 2010.

- [HTF09] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning*, Data Mining, Inference, and Prediction (Second Edition), Springer, 2009.
- [Hul11] Eyke Hullermeier. *Fuzzy sets in machine learning and data mining*, Appl. Soft Comput. 11(2): 1493-1505, 2011.
- [Hunter10] Gordon Hunter. *The Chief Information Officer: A Review of the Role*, Journal of Information, Information Technology, and Organizations, 5: 125-143, 2010.
- [Hyll08] Eyke Hullermeier. *Fuzzy Methods for Data Mining and Machine Learning: State of the Art and Prospects*, Fuzzy Sets and Their Extensions: Representation, Aggregation and Models 2008: 357-375.
- [HZ10] Mojdeh Jalali Heravi, Osmar R. Zaúane. *A study on interestingness measures for associative classifiers*, SAC 2010: 1039-1046, 2010.
- [IDC10] *IDC Digital Universe Study*, sponsored by EMC, May 2010.
- [Inm02] W. H. Inmon. *Building the Data Warehouse* (Third Edition), Wiley Computer Publishing, 2002.
- [JC10] Richard Jensen, Chris Cornelis. *Fuzzy-rough instance selection*, FUZZ-IEEE 2010: 1-7, 2010.
- [JC11] Richard Jensen, Chris Cornelis. *Fuzzy-Rough Nearest Neighbour Classification*, Transactions on Rough Sets XIII, (J.F. Peters et al., Eds.): 56-72., 2011.
- [Jen05] Richard Jensen. *Combining rough and fuzzy sets for feature selection*, PhD Thesis, University of Edinburgh, 2005.
- [Jen11] Richard Jensen. *Fuzzy-rough data mining (A tutorial)*, Thirteenth International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC-2011), Higher School of Economics, Moscow, Russia, June 25 - June 27, 2011.

- [JIA1] H.Jiawei, P. Jian, Y.Yiwen, *Mining frequent patterns without candidate generation*, 2003.
- [JIA2] H. Jiawei, M.Kamber, and P.Jian, *Data Mining: Concepts and Techniques*, 3 edition, Morgan Kaufmann, 2011.
- [JS09] Richard Jensen, Qiang Shen. *New Approaches to Fuzzy-Rough Feature Selection*. IEEE T. Fuzzy Systems 17(4): 824-838, 2009.
- [KFW98] Chan Man Kuok, Ada Wai-Chee Fu, Man Hon Wong. *Mining Fuzzy Association Rules in Databases*, SIGMOD Record 27(1): 41-46, 1998.
- [KH10] Andreas M Kaplan, Michael Haenlein. *Users of the world, unite! The challenges and opportunities of Social Media*, Business horizons, 53:59-68, 2010.
- [Kim03] Won Kim. *Data Mining Is NOT Against Civil Liberties*, ACM Special Interest Group on Knowledge Discovery and Data Mining, www.acm.org/sigkdd/, June 30, 2003.
- [KV01] Boris Kovalerchuk and Evgenii Vityaev. *Data Mining in Finance: Advances in Relational and Hybrid Methods*. Kluwer Academic Publishers, Boston, Dordrecht - London, 2001.
- [Lang95] K. Lang, Newsweeder: *Learning to filter netnews*. Proceedings of the Twelfth International Conference (ICML '95), pp. 331-339, 1995.
- [Leary95] Daniel O'Leary. *Some Privacy Issues in Knowledge Discovery*: OECD Personal Privacy Guidelines, Experts Annual Index, 10(2): 48-52, 1995.
- [Lesk08] Jure Leskovec. *Dynamics of large networks*, PhD Thesis, Carnegie Mellon University, 2008.
- [Lesk11] Jure Leskovec. *Social Media Analytics*, Tutorial at the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Part 1: Information flow, Part2: Rich Interactions), 2011.

- [LH97] Lee J. H. and Hyung L. K. *An Extension of Association Rules using Fuzzy Sets*, Seventh IFSA World Congress: 399-402, Prague, 1997.
- [LHK10] J. Leskovec, D. Huttenlocher, J. Kleinberg. *Predicting Positive and Negative Links in Online Social Networks*, WWW, ACM Press, New York, 2010.
- [Li07] Jiye Li. *Rough Set Based Rule Evaluations and Their Applications*. PhD. Thesis, University of Waterloo, Ontario, Canada, 2007.
- [Line07] Jeffrey P. Lineman. *The Corporate CIO Model and the Higher Education CIO*, EQ, 30 (1): 4-5, 2007.
- [LMFHL04] Nada Lavrac, Hiroshi Motoda, Tom Fawcett, Robert Holte, Pat Langley, Pieter W. Adriaans. *Introduction: Lessons Learned from Data Mining Applications and Collaborative Problem Solving*, Machine Learning 57(1-2): 13-34, 2004.
- [LZ10] L. Lu and T. Zhou. *Link prediction in complex networks: A survey*, Physica A, 390:1150–1170, 2010.
- [LZLCD12] Jiye Liang, Xingwang Zhao, Deyu Li, Fuyuan Cao, Chuangyin Dang. *Determining the number of clusters using information entropy for mixed data*, Pattern Recognition 45(6): 2251-2265, 2012.
- [MBCCC10] Tom M. Mitchell, Justin Betteridge, Jamie Callan, Andy Carlson, William Cohen, Estevam, Hruschka, Bryan Kisiel, Mahaveer Jain, Jayant Krishnamurthy, Edith Law, Thahir Mohamed, Mehdi Samadi, Burr Settles, Richard Wang, Derry Wijaya. *Never Ending Learning*, ICML 2010 (Invited Talk), Haifa, Israel, June 21-24, 2010.
- [MCB11] Huina Mao, Scott Counts, Johan Bollen. *Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data*, CoRR abs/1112.1051, 2011.

- [Milgram06] J. Milgram, M. Cheriet, R. Sabourin, *One Against One or One Against All: Which One is Better for Handwriting Recognition with SVMs?*, Tenth International Workshop on Frontiers in Handwriting Recognition, 2006.
- [Mitch06] Tom M. Mitchell. *The Discipline of Machine Learning*, CMU-ML-06-108, July 2006.
- [Mitchell97] T. M. Mitchell, *Machine Learning*. McGraw-Hill International Edit, 1997.
- [MKG04] Nigel Melville, Kenneth L. Kraemer, Vijay Gurbaxani. *Review: Information Technology and Organizational Performance: An Integrative Model of IT Business Value*, MIS Quarterly, 28 (2): 283-322, 2004.
- [Moore65] Gordon E. Moore. *Cramming more components onto integrated circuits*, Electronics, 38 (8), April 19, 1965.
- [MR11] Ralf Mikut, Markus Reischl. *Data mining tools*, Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery 1(5): 431-443, 2011.
- [MSOS10] Craig Macdonald, Rödrygo L.T. Santos, Iadh Ounis, Ian Soboroff. *Blog Track Research at TREC*, SIGIR Forum 44(1): 58-75, 2010.
- [Nauck00] Detlef Nauck. *Data Analysis with Neuro-Fuzzy Methods*, Dr. of Science Thesis, der Otto-von-Guericke-Universit at Magdeburg, 2000.
- [NEM09] Robert Nisbet, John Elder, and Gary Miner. *Handbook of Statistical Analysis and Data Mining*, Elsevier, 2009.
- [Nguyen08] N. T. Thanh, N. L. Minh and A. Shimazu, *Using Semi-supervised Learning for Question Classification*, Journal of Natural Language Processing, 3(1):112-130, 2008.
- [Nigam00] K. Nigam and R. Ghani. *Analyzing the effectiveness and applicability of co-training*. In Proceedings of Ninth International Conference on Information and Knowledge Management, pages 86-93, 2000.

- [NS08] Hung Son Nguyen, Andrzej Skowron. *Rough Set Approach to KDD*, <http://sist.swjtu.edu.cn/imc/itw06/rskt2008/Skowron.pdf>, 2008.
- [NSF05] *National Science Foundation Long-Lived Digital Data Collections Enabling Research and Education in the 21st Century*, Reporting of National Science Foundation, National Science Board, <http://www.nsf.gov/pubs/2005/nsb0540/>
- [OESD96] OECD. *The knowledge-based economic, organisation for economic co-operation and development*, 1996.
- [Ohrn99] Aleksander Ohrn. *Discernibility and Rough Sets in Medicine: Tools and Applications*, PhD. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1999.
- [Pan10] Ding Pan. *An Integrative Framework for Continuous Knowledge Discovery*, Journal of Convergence Information Technology (JCIT), 5 (3): 46-53, May 2010.
- [Pawlak82] Pawlak Z. *Rough set*, International Journal of Computer and Information Sciences, 11 (5): 341-356, 1982.
- [Pawlak85] Pawlak Z. *Rough set and Decision Tables*, ICS PAS Report, 540, 3-1984, Warsawa, Poland, 1985.
- [PCVM10] Luis Portela, Roberto Carvalho, João Varajão, and Luis Magalhães. *A Review of Chief Information Officer' Main Skills*, M.D. Lytras et al. (Eds.): WSKS 2010, Part II, CCIS 112: 387–392, Springer-Verlag Berlin Heidelberg, 2010.
- [Pia06] Gregory Piatetsky-Shapiro. *Data Mining Course (Power Point Version)*. <http://www.kdnuggets.com/index.html>, 2006.
- [QLPD10] Yuhua Qian, Jiye Liang, Witold Pedrycz, Chuangyin Dang. *Positive approximation: An accelerator for attribute reduction in rough set theory*, Artificial Intelligence 174 (2010): 597–618, 2010.

- [RB10] Pascal Ravesteyn and Ronald Batenburg. *Cultural Differences in Implementing Business Process Management Systems*, AMCIS 2010 Proceedings Americas Conference on Information Systems: Paper 340, 2010.
- [RK02] A.M. Radzikowska, E.E. Kerre. *A comparative study of fuzzy rough sets*, Fuzzy Sets and Systems, 126 (2): 137-155, 2002.
- [RU11] Anand Rajaraman, Jeffrey D. Ullman. *Mining of Massive Datasets*, <http://i.stanford.edu/~ullman/mmds/book.pdf>, 2011.
- [SB08]. Swan, A and Brown, S. *The skills, role and career structure of data scientists and curators: An assessment of current practice and future needs*, A report to JISC, <http://eprints.ecs.soton.ac.uk/16675/>, 2008.
- [Schapire99] R. E. Schapire and Y. Singer, *Improved Boosting Algorithms Using Confidence-rated Predictions*, Machine Learning, 37(3):297-336, 1999.
- [SG10] Sulabh Sharma, Jairo A. Gutiérrez: *An evaluation framework for viable business models for m-commerce in the information technology sector*. Electronic Markets 20(1): 33-52, 2010.
- [Shap95] Gregory Piatetsky-Shapiro. *Guidelines for Eating of the Tree of Knowledge, or Knowledge Discovery in Databases vs. Personal Privacy*, Experts Annual Index, 10(2): 46-47, 1995.
- [Simon08] Morten Simonsson. *Predicting It Governance Performance: A Method For Model-Based Decision Making*, PhD Thesis, KTH-Royal Institute Of Technology, Stockholm, Sweden, April 2008.
- [Solow87] Robert M. Solow. *We'd Better Watch Out*, The New York Time: Book Review, page 36, July 12, 1987.
- [Spoh06] Jim Spohrer. *A Next Frontier in Education, Employment, Innovation, and Economic Growth*, IBM Corporation, 2006.

- [STH06] Son Doan, Quang Thuy Ha, and Susumu Horiguchi. *A General Fuzzy-based Framework for Text Representation and its Application to Text Categorization*, Lecture Notes on Artificial Intelligence (LNAI), 4423: 611-620, 2006.
- [Strass07] Paul A. Strassmann, *Measuring and Communicating I.T. Value*, <http://www.strassmann.com/talks/one-talk.php?talk=123>, 2007.
- [SZ00] Andrzej Skowron, Ning Zhong. *Rough Sets in KDD*, Tutorial Notes, PAKDD 2000.
- [Szczu11] Marcin Szczuka. *The use of Rough Set methods in KDD*, A Tutorial in Thirteenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing (RSFDGrC-2011), Higher School of Economics, Moscow, Russia, 2011.
- [TSK05] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. *Introduction to Data Mining*, Addison Wesley, 2005.
- [Vaidya04] Jaideep Shrikant Vaidya. *Privacy Preserving Data Mining over Vertically Partitioned Data*, PhD Thesis, Purdue University, 2004.
- [VBFPS04] Verykios V. S., Bertino E., Fovino I. N., Provenza L. P., Saygin Y., Theodoridis Y. *State-of-the-art in privacy preserving data mining*, ACM SIGMOD Record, 33 (1):50-57, 2004.
- [VCKP08] Vaidya, J., Clifton, C., Kantarcioglu, M., and Patterson, A. S. *Privacy-preserving decision trees over vertically partitioned data*. ACM Trans. Knowl. Discov. Data. 2, 3, Article 14, 2008.
- [Vincent03] V. Ng and C. Cardie, *Bootstrapping Coreference Classifiers with Multiple Machine Learning Algorithms*. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (EMNLP), Sapporo, Japan. 2003.

- [WB06] The World Bank (2006) Korea as a Knowledge Economy: *Evolutionary Process and Lessons Learned*, © 2006 The International Bank for Reconstruction and Development/The World Bank.
- [WB98] Christopher Westphal and Teresa Blaxto. *Data Mining Solutions Methods and Tools for Solving Real-World Problems*, John Wiley & Sons, Inc., 1998.
- [WFBHM10] Tim Weninger, Fabio Fumarola, Rick Barber, Jiawei Han, Donato Malerba. *Unexpected Results in Automatic List Extraction on the Web*, ACM SIGKDD Explorations newsletter, 12(2): 26-30, 2010.
- [WKQ08] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu , Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg. *Top 10 algorithms in data mining*, Knowl Inf Syst (2008) 14:1–37, 2008.
- [WW08] Wang, H. and S. Wang. *A knowledge management approach to data mining process for business intelligence*, Industrial Management & Data Systems, 108(5): 622-634, 2008.
- [Yao03] Yao Y.Y. *Information-theoretic measures for knowledge discovery and data mining*, Entropy Measures, Maximum Entropy and Emerging Applications, Karmeshu (Ed.), Springer, Berlin, 115-136, 2003.
- [Yarowsky95] D. Yarowsky. *Unsupervised Word Sense Disambiguation Rivaling Supervised Methods*. In Proceedings of the 33rd Annual Meeting of the ACL, 1995.
- [Yasien07] Ahmed HajYasien. *Preserving Privacy in Association Rule Mining*, PhD Thesis, Griffith University (Australia), 2007.
- [Your11]. E. Yourdon (2011), *CIOs at Work*, Springer, 2011.
- [YZ10] Yiyu Yao, Bing Zhou. *Naive Bayesian Rough Sets*. RSKT 2010: 719-726, 2010.

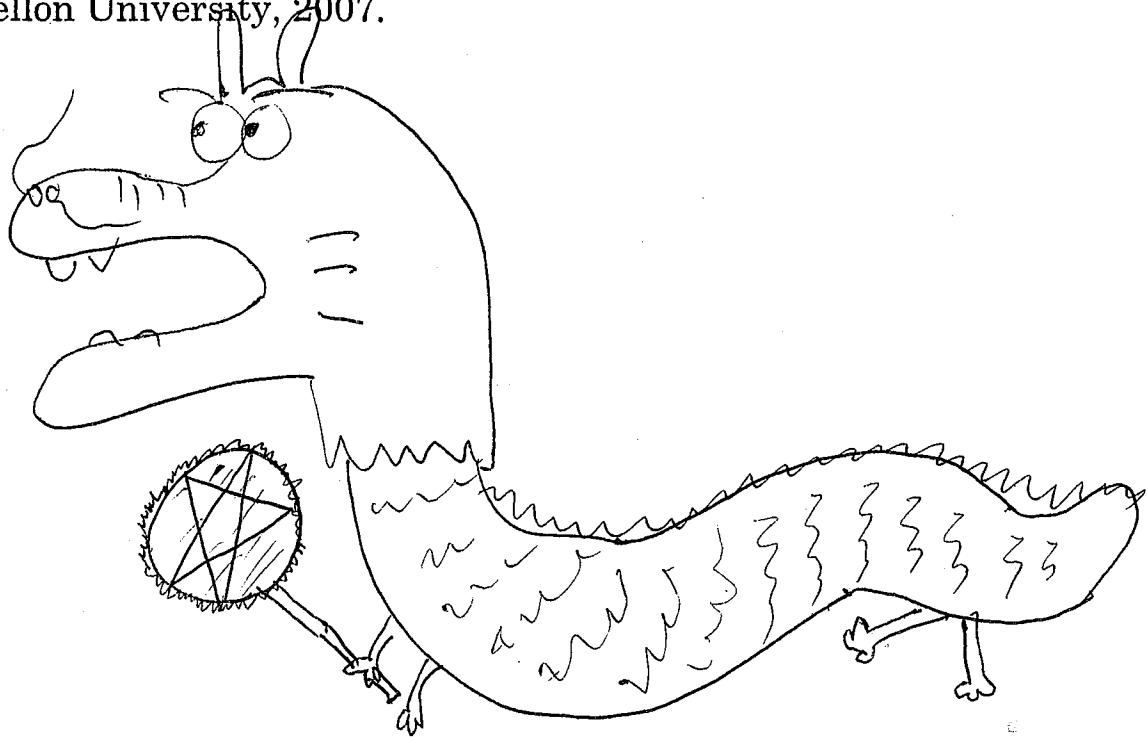
- [Zadeh65] Zadeh L.A. *Fuzzy sets*, Information and Control, 8: 338-353, Academic Press, New York, 1965.
- [Zadeh75] Zadeh L.A. *The concept of a linguistic variable and its application to approximate reasoning* (Parts I, II, and III), Information Sciences, 8:199-249; 8:301-357; 9: 43-80, 1975.
- [Zadeh78] Zadeh L.A. *Fuzzy sets as a basis for a theory of possibility*, Fuzzy Sets and Systems, 1: 3-28, 1978.
- [Zdarkov07] Z. Markov and D. T. Larose, *Data mining the web, uncovering patterns in Web content, structure and usage*, John Wiley & Sons, 2007.
- [ZHL98] Osmar R. Zaiane, Mohammad El-Hajj, and Paul Lu. *Fast Parallel Association Rule Mining Without Candidacy Generation*. University of Alberta, Edmonton, Alberta, Canada, 1998.
- [Zhou03] Zhi-Hua Zhou. *Three perspectives of data mining*, Artif. Intell. 143(1): 139-146, 2003.
- [Zhou05] Z. H. Zhou and M. Li, *Tri-Training: Exploiting Unlabeled Data Using Three Classifiers*, IEEE Transactions on Knowledge and Data Engineering, Vol. 17, No. 11, pp. 1529-1541, 2005.
- [Zhu05] X. Zhu. *Semi-supervised learning with graphs*. PhD Thesis, Carnegie Mellon University, CMU-LTI-05-192, 2005.
- [Zhu08] Xiaojin Zhu. *Semi-supervised learning literature survey*, Technical Report 1530, University of Wisconsin at Madison, 2008.
- [Zia94] Wojciech P. Ziarko (Ed., 1994). *Rough Sets, Fuzzy Sets and Knowledge Discovery*. Proceedings of the International Workshop on Rough Sets and Knowledge Discovery (RSKD'93), Banff, Alberta, Canada, Springer-Verlag, 12-15, 1993.

[ZPO01] Mohammed J. Zaki, Srinivasan Parthasarathy, and Mitsunori Ogihara. *Parallel Data Mining for Association Rules on Shared-Memory Systems*. In *Knowledge and Information Systems*, Vol. 3, Number 1, pages 1-29, 2001.

[ZYC09] Sheng Zhong, Zhiqiang Yang, Tingting Chen. *k-Anonymous data collection*, *Information Sciences (ISCI)*, 179(17): 2948-2963, 2009.

[ZZNS09] Yuejin Zhang, Lingling Zhang, Guangli Nie, Yong Shi. *A Survey of Interestingness Measures for Association Rules*, 2009 International Conference on Business Intelligence and Financial Engineering: 460-463, 2009.

Zhu[07] X. Zhu. *Semi-Supervised Learning Literature Survey*, Mellon University, 2007.



NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA HÀ NỘI

16 Hàng Chuối – Hai Bà Trưng Hà Nội

Giám đốc – Tổng biên tập: (04) 39715011; Hành chính: (04) 39714899;

Fax: (04) 39714899, Kinh doanh: (04) 39729437

Chịu trách nhiệm xuất bản:

Giám đốc – Tổng biên tập : TS. Phạm Thị Trâm

Biên tập : NGUYỄN THỊ THỦY

Ché bản : NGUYỄN THANH NHÀN

Trình bày bìa : NGUYỄN NGỌC ANH

GIÁO TRÌNH KHAI PHÁ DỮ LIỆU

Mã số: 1L-38ĐH2016, ISBN: 978-604-62-0955-3

In: 500 cuốn, khổ 16x24 cm tại Công ty TNHH In - TM và DV Nguyễn Lâm.

Địa chỉ: 325 đường Giải Phóng, Q. Thanh Xuân, Hà Nội

Số ĐKXB: 4535-2016/CXBIPH/06-354/ĐHQGHN, ngày 12/12/2016

Quyết định xuất bản số: 1369LK-TN/QĐ-NXB ĐHQGHN, ngày 20/12//2016

In xong và nộp lưu chiểu năm 2016

