ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

VI VĂN SƠN

PHÂN CỤM THÔ CỦA DỮ LIỆU TUẦN TỰ

Ngành:Hệ thống thông tin

Chuyênngành: Hệ thống thông tin

Mã số: 60480104

LUẬN VĂN THẠC SĨ NGÀNH CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC : PGS.TS Hoàng Xuân Huấn

HàNội, năm 2016

LỜI CẢM ƠN

Trước hết, tôi xin gửi lời biết ơn sâu sắc đến người thầy PGS. TS Hoàng Xuân Huấn đã dành rất nhiều thời gian và tâm huyết hướng dẫn nghiên cứu và giúp tôi hoàn thành tốt luận văn tốt nghiệp này. Thầy đã mở ra cho tôi những vấn đề khoa học rất lý thú, định hướng nghiên cứu các lĩnh vực hết sức thiết thực, đồng thời tạo điều kiện thuận lợi tốt nhất cho tôi học tập và nghiên cứu.

Tôi cũng xin được bày tỏ lòng biết ơn tới các thầy cô trường Đại học Công nghệ đã tham gia giảng dạy và chia sẻ những kinh nghiệm quý báu cho tập thể và cá nhân tôi nói riêng. Tôi xin cảm ơn tất cả các Anh, Chị và các bạn luôn chia sẻ, giúp đỡ, trao đổi, góp ý trong quá trình học tập.

Tôi xin gửi lời biết ơn tới bố mẹ, gia đình và người thân đã tạo mọi điều kiện tốt nhất để tôi cơ hội lựa chọn con đường đi của mình.

Một lần nữa, tôi xin chân thành cảm ơn!

Hà Nội, tháng 11 năm 2016.

Học viên

Vi Văn Sơn

LỜI CAM ĐOAN

Những kiến thức trình bày trong luận văn là do tôi tìm hiểu, nghiên cứu và trình bày lại theo cách hiểu. Trong quá trình làm luận văn tôi có tham khảo các tài liệu có liên quan và đã ghi rõ nguồn tài liệu tham khảo đó. Tôi xin cam đoan đây là công trình nghiên cứu của tôi và không sao chép của bất kỳ ai.

Hà Nội, tháng 11 năm 2016.

Học viên

Vi Văn Sơn

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, TỪ VIẾT TẮT

CSDL	Cơ sở dữ liệu
DBSCAN	Density – Based Spatial Clustering of Applications with Noise
FN	Furthest Neighbour
GIS	Geographic Information System
LLCS	Length of longest common subsequence
NN	Nearest Neighbour
PCDL	Phân cụm dữ liệu
RelSim	Relative similarity
S^3M	Similarity measure for sequences
SeqSim	Sequence similarity
SetSim	Set similarity
STING	STatistical Information Grid approach

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG

MỞ ĐẦU

Phân cụm dữ liệu là một kỹ thuật quan trọng trong công nghệ tri thức, nó được ứng dụng rộng rãi và đa dạng trong các ngành khoa học như sinh học, tâm lý học, y học, ngành marketing, thị giác máy tính, và điều kiển học v.v. Phân cụm dữ liệu tổ chức dữ liệu bằng cách nhóm các đối tượng có độ tương đồng cao vào một cụm, các đối tượng thuộc các cụm khác nhau có độ tương đồng thấp hơn so với các đối tượng trong cùng một cụm. Tùy theo đặc điểm cấu trúc của tập dữ liệu và mục đích sử dụng, có các phương pháp giải quyết khác nhau như: Phân cụm dựa vào hàm mục tiêu, phân cụm phân cấp, phân cụm dựa vào mật độ và phân cụm dựa vào lưới.

Thông thường, thông tin về thế giới xung quanh là không chính xác, không đầy đủ, không chắc chắn hoặc chồng chéo. Đó cũng là vấn đề gặp phải khi phân cụm dữ liệu. Phân cụm được chia làm hai loại phân cụm là phân cụm cứng và phân cụm mềm. Trong phân cụm cứng đối tượng được phân thành các cụm khác nhau, mỗi đối tượng thuộc về chính xác một cụm, ngược lại ở phân cụm mềm các đối tượng có thể thuộc về nhiều hơn một cụm và mỗi đối tượng có độ thuộc với cụm.

Lý thuyết tập thô (Rough Set Theory) do Zdzisaw Pawlak (1926-2006) đề xuất vào năm 1982 đã được ứng dụng ngày càng rộng rãi trong lĩnh vực khoa học máy tính. Lý thuyết tập thô được phát triển trên một nền tảng toán học vững chắc, cung cấp các công cụ hữu ích để giải quyết các bài toán phân tích dữ liệu, phát hiện luật, nhận dạng… Đặc biệt thích hợp với các bài toán phân tích trên khối lượng dữ liệu lớn, chứa đựng thông tin mơ hồ, không chắc chắn. Mục đích chính của phân tích dữ liệu dựa trên lý thuyết tập thô nhằm đưa ra các xấp xỉ để biểu diễn các đối tượng không thể được phân lớp một cách chắc chắn bằng tri thức có sẵn. Theo quan điểm của lý thuyết tập thô, mọi tập thô đều liên kết với 2 tập "rõ" là xấp xỉ dưới và xấp xỉ trên của nó. Xấp xỉ dưới bao gồm các đối tượng chắc chắn thuộc, còn xấp xỉ trên chứa tất cả các đối tượng có khả năng thuộc về tập đó. Các tập xấp xỉ là cơ sở để rút ra các kết luận(tri thức) từ cơ sở dữ liệu. Do đó trong luận văn này dựa trên lý thuyết tập thô cụ thể là xấp xỉ trên của tập thô và thuật toán phân cụm thô được đề xuất áp dụng phân cụm trên dữ liệu tuần tự.

Cấu trúc của luận văn của tôi được chia làm ba chương như sau:

Chương 1: Tổng quan về phân cụm dữ liệu. Giới thiệu về phân cụm dữ liệu và các phương pháp phân cụm.

Chương 2: Lý thuyết tập thô. Trình bày tổng quan về lý thuyết tập thô bao gồm hệ thông tin, bảng quyết định, tính không phân biệt được và xấp xỉ tập hợp.

Chương 3:Áp dụng thuật toán phân cụm thô vào bài toán phân cụm người dùng trên Web. Dựa trên lý thuyết tập thô và áp dụng thuật toán phân cụm thô phân cụm người dùng trên Web(chuyển hướng Web của người dùng).

CHƯƠNG I
TỔNG QUAN VỀ PHÂN CỤM DỮ LIỆU

1.1 Phân cụm dữ liệu là gì

Phân cụm dữ liệu là một kỹ thuật trong khai phá dữ liệu nhằm tìm kiếm, phát hiện các cụm, cácmẫu dữ liệu tự nhiên, tiềm ẩn, quan trọng trong tập dữ liệu lớn từ đó cung cấpthông tin, tri thức hữu ích cho việc ra quyết định.

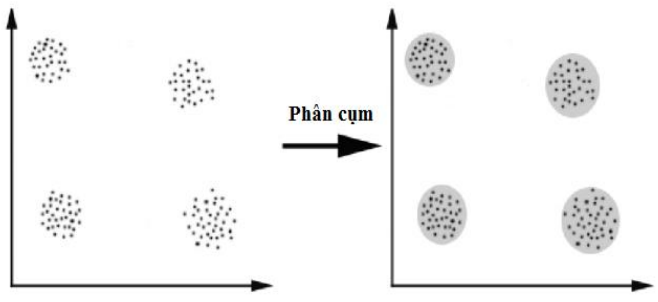
Phân cụm nhìn từ góc độ tự nhiên là một việc hết sức bình thường mà chúng ta vẫn làm và thực hiện hàng ngày. Ví dụ như phân loại học sinh trong lớp; phân loại đất đai; phân loại tài sản; phân loại sách trong thư viện;…

Cụm dữ liệu là tập hợp các đối tượng có những tính chất nào đó tương tự nhau ở một mức độ nào đó trong tập dữ liệu.

Ở một mức cơ bản nhất, người ta đã đưa ra định nghĩa phân cụm dữ liệu (PCDL) như sau:[3]

"Phân cụm dữ liệu là một kỹ thuật trong khai phá dữ liệu (Data mining), nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tự nhiên tiềm ẩn, quan tâm trong tập dữ liệu lớn, từ đó cung cấp thông tin, tri thức hữu ích cho ra quyết định."

Quá trình PCDL là quá trình phân chia một tập dữ liệu ban đầu thành các cụm dữ liệu sao các phần tử trong cùng một cụm thì "tương tự" nhau và các phần tử trong các cụm khác nhau thì "kém tương tự" nhau. Số các cụm dữ liệu được phân ở đây có thể được xác định trước theo kinh nghiệm hoặc có thể được tự động xác định theo phương pháp phân cụm.



Hình 1.1 Mô phỏng vấn đề phân cụm dữ liệu.

Trong học máy, PCDL được xem là vấn đề học không có giám sát (unsupervised learning), vì nó phải giải quyết vấn đề tìm một cấu trúc trong tập hợp dữ liệu chưa biết trước các thông tin về cụm, các thông tin về tập huấn luyện hay thông tin nhãn của các lớp. Trong nhiều trường hợp, nếu phân lớp được xem là vấn đề học có giám sát thì PCDL là một bước trong phân lớp dữ liệu, nó sẽ khởi tạo các lớp cho phân lớp bằng cách xác định các nhãn cho các nhóm dữ liệu.[3,2]

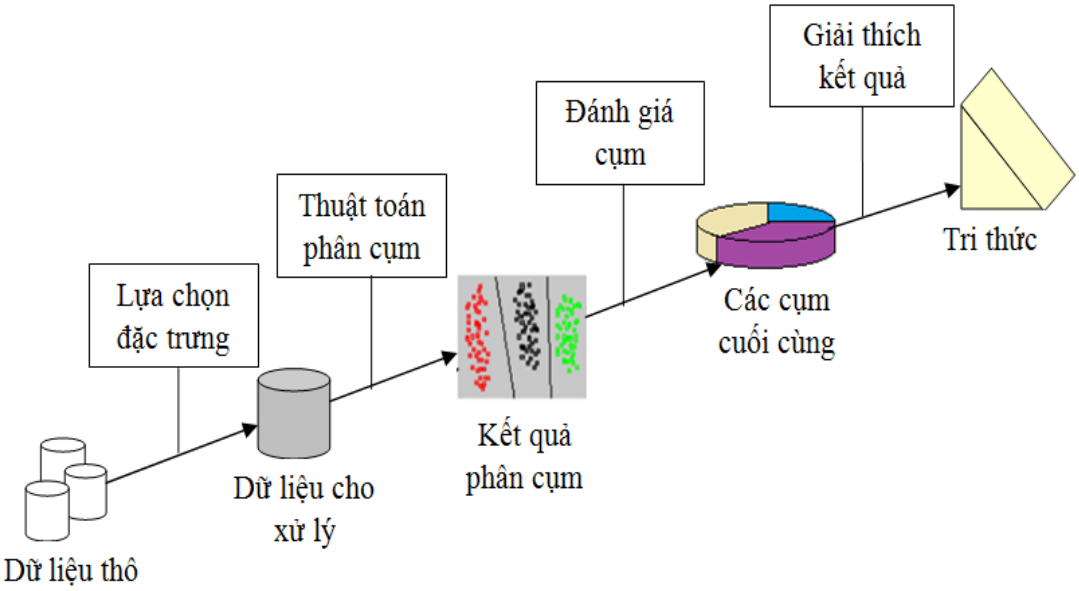
Với một tập dữ liệu, quá trình phân cụm có thể cho ra nhiều kết quả khác nhau tùy thuộc vào tiêu chí cụ thể được sử dụng để phân cụm. Các bước cơ bản của quá trình phân cụm được thể hiện trong hình 1.1 và được tóm tắt như sau:

Lựa chọn đặc trưng (Feature selection): các đặc trưng phải được lựa chọn một cách hợp lý để có thể "mã hóa" nhiều thông tin nhất liên quan đến nhiệm vụ mà chúng ta quan tâm. Mục tiêu chính là giảm thiểu dư thừa thông tin giữa các đặc trưng. Do đó, tiền xử lý dữ liệu là một nhiệm vụ quan trọng trước khi tiến hành các bước sau.

Lựa chọn thuật toán phân cụm (clustering algorithm selection): cần lựa chọn một sơ đồ thuật toán riêng biệt nhằm làm sáng tỏ cấu trúc của tập dữ liệu.

Đánh giá kết quả phân cụm (validation of results): Khi đã có kết quả phân cụm thì ta phải kiểm tra tính đúng đắn của nó. Với cùng một tập dữ liệu, những cách tiếp cận khác nhau thường dẫn tới các kết quả phân cụm khác nhau và ngay cả cùng một thuật toán với các tham số đầu vào khác nhau cũng cho ra các kết quả khác nhau. Vì vậy, các tiêu chuẩn và tiêu chí để đánh giá kết quả phân cụm là rất quan trọng. Nó cung cấp cho người dùng mức độ tin cậy của các kết quả mà thuật toán phân cụm thực hiện.

Giải thích kết quả (interpretation of results): Mục tiêu cuối cùng của việc phân cụm là cung cấp cho người sử dụng những hiểu biết ý nghĩa từ dữ liệu gốc. Các chuyên gia phải giải thích những phân vùng dữ liệu thu được. Trong nhiều trường hợp, các chuyên gia trong các lĩnh vực ứng dụng phải tích hợp các kết quả phân cụm với các bằng chứng thực nghiệm khác và phân tích để rút ra những kết luận đúng.

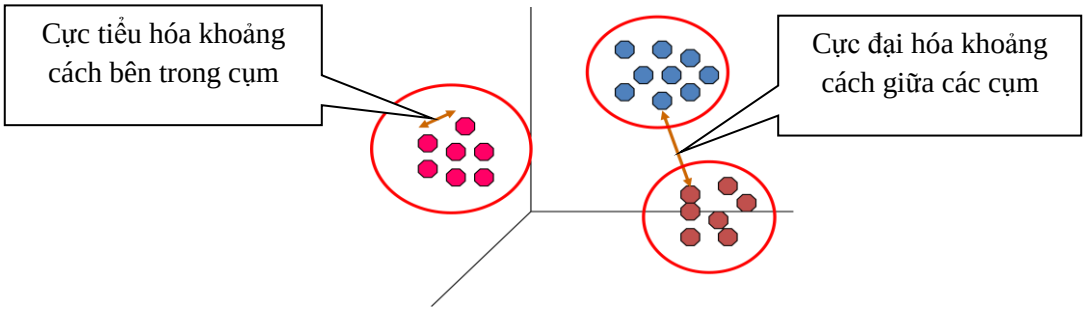


Hình 1.2 Các bước của quá trình phân cụm dữ liệu.

1.2 Thế nào là phân cụm tốt

Một phương pháp phân cụm tốt sẽ sinh ra các cụm có chất lượng cao [3], trong đó:

- Mức độ tương tự giữa các đối tượng trong cùng một cụm là cao.
- Mức độ tương tự giữa các đối tượng nằm trong các cụm khác nhau là thấp.



Hình 1.3 Tiêu chuẩn phân cụm.

Chất lượng của kết quả phân cụm phụ thuộc vào cả độ đo tương tự được sử dụng và cách thức thực hiện.

Chất lượng của phương pháp phân cụm cũng được đánh giá bởi khả năng phát hiện các mẫu tiềm ẩn.

Các yêu cầu của phân cụm trong khai phá dữ liệu:[3,2]

Việc xây dựng và lựa chọn một thuật toán phân cụm là bước then chốt cho việc giải quyết vấn đề phân cụm, sự lựa chọn này phụ thuộc vào đặc tính dữ liệu cần phân cụm, mục đích của ứng dụng thực tế hoặc xác định độ ưu tiên giữa chất lượng của các cụm hay tốc độ thực hiện thuật toán,...

Hầu hết các nghiên cứu và phát triển thuật toán PCDL đều nhằm thỏa mãn các yêu cầu cơ bản sau:

- *Có khả mở rộng* :

  Một số thuật toán có thể ứng dụng tốt cho tập dữ liệu nhỏ (khoảng 200 bản ghi dữ liệu) nhưng không hiệu quả khi áp dụng cho tập dữ liệu lớn(khoảng 1 triệu bản ghi).
- *Thích nghi với các kiểu dữ liệu khác nhau*:

  Thuật toán có thể áp dụng hiệu quả cho việc phân cụm các tập dữ liệu với nhiều kiểu dữ liệu khác nhau như dữ liệu kiểu số, kiểu nhị phân, dữ liệu định danh, hạng mục,…và thích nghi với dữ liệu hỗn hợp.
- *Khám phá ra các cụm với hình dạng bất kỳ*:

  Do hầu hết các CSDL có chứa nhiều cụm dữ liệu với các hình thù khác nhau như: Hình lõm, hình cầu, hình que,…Vì vậy, để khám phá được các cụm có tính tự nhiên thì các thuật toán phân cụm cần phải có khả năng khám phá ra các cụm dữ liệu có hình thù bất kỳ.
- *Tối thiểu lượng tri thức cần cho xác định các tham số vào*: Do các giá trị đầu vào ảnh hưởng rất lớn đến thuật toán phân cụm và rất phức tạp để xác định các giá trị vào thích hợp đối với các CSDL lớn.
- *Khả năng thích nghi với dữ liệu nhiễu*:

  Hầu hết các dữ liệu phân cụm trong khai phá dữ liệu đều chứa đựng các dữ liệu lỗi, dữ liệu không đầy đủ, dữ liệu rác. Thuật toán phân cụm không những hiệu quả đối với các dữ liệu nhiễu mà còn tránh dẫn đến chất lượng phân cụm thấp do nhạy cảm với nhiễu.
- *Ít nhạy cảm với các tham số đầu vào*:

  Nghĩa là giá trị của các tham số đầu vào khác nhau ít gây ra các thay đổi lớn đối với kết quả phân cụm.
- *Có khả năng phân cụm với dữ liều có số chiều cao*:

  Thuật toán có khả năng áp dụng hiệu quả cho dữ liệu có số chiều khác nhau.

- *Dễ hiểu, cài đặt và khả thi*:

  Các yêu cầu này đồng thời là các tiêu chí để đánh giá hiệu quả của các phương pháp PCDL, đây là những thách thức cho các nhà nghiên cứu trong lĩnh vực PCDL.

1.3 Các ứng dụng của phân cụm dữ liệu

Phân cụm dữ liệu là một trong những công cụ chính được ứng dụng trong nhiều lĩnh vực. Một số ứng dụng của phân cụm như: [3]

Xử lý dữ liệu lớn: việc khám phá tri thức trong các cơ sở dữ liệu thường phải xử lý khối lượng dữ liệu rất lớn, nhiều khi ngay cả các thuật toán với độ phức tạp tính toán là đa thức cũng không dùng được. Do đó, việc phân và xử lý theo các cụm là một giải pháp hữu hiệu.

Tạo giả thuyết: phân tích cụm được sử dụng để suy ra một số giả thuyết liên quan đến dữ liệu. Ví dụ: dựa trên tuổi tác và thời điểm mua hàng, chúng ta có thể tìm thấy trong một cơ sở dữ liệu bán lẻ có hai nhóm khách hàng quan trọng. Sau đó, chúng ta có thể suy ra một số giả thuyết cho dữ liệu là: "*những người trẻ tuổi đi mua sắm vào buổi tối*", "*người già đi mua sắm vào buổi sáng*".

Kiểm định giả thuyết: Trong trường hợp này, phân tích cụm được sử dụng cho việc xác minh tính hợp lệ của một giả thuyết cụ thể. Ví dụ, chúng ta xem xét giả thuyết như sau: "*Những người trẻ tuổi đi mua sắm vào buổi tối*". Một cách để xác minh điều này là áp dụng phân tích cụm cho một tập đại diện các cửa hàng. Giả sử rằng mỗi cửa hàng được đặc trưng bởi các chi tiết của khách hàng (tuổi tác, công việc, …) và thời điểm giao dịch. Nếu, sau khi áp dụng phân tích cụm, một cụm tương ứng với "*những người trẻ mua sắm vào buổi tối*" được tạo thành, thì giả thuyết ban đầu đã được chứng minh là hợp lệ.

Cụ thể, các kỹ thuật phân cụm dữ liệu đã được áp dụng cho một số ứng dụng điển hình trong các lĩnh vực sau:

Thương mại: Trong thương mại, phân cụm dữ liệu có thể giúp các thương nhân khám phá ra các nhóm khách hàng quan trọng có các đặc trưng tương đồng nhau và đặc tả họ từ các mẫu mua bán trong cơ sở dữ liệu khách hàng.

Sinh học: Phân cụm dữ liệu được sử dụng để xác định các loài sinh vật, phân loại các Gen với chức năng tương đồng và thu được những hiểu biết bên trong những cấu trúc của quần thể.

Phân tích dữ liệu không gian: Do một lượng lớn dữ liệu không gian có thể thu được từ các hình ảnh vệ tinh, thiết bị y tế, hệ thống thông tin địa lý (GIS), cơ sở dữ

liệu hình ảnh thăm dò,… làm cho người dùng tốn kém và khó khăn để kiểm tra các dữ liệu không gian một cách cụ thể. Phân cụm dữ liệu có thể giúp người dùng tự động phân tích và xử lý các dữ liệu không gian. Nó được sử dụng để nhận dạng, trích xuất các đặc tính hoặc các mẫu dữ liệu quan tâm có thể tồn tại trong cơ sở dữ liệu không gian lớn.

Khai phá Web (Web mining): phân cụm dữ liệu có thể khám phá các nhóm tài liệu quan trọng, có nhiều ý nghĩa trong môi trường web. Các lớp tài liệu này hỗ trợ trong việc phát hiện ra thông tin. Trong tìm kiếm tương tự (similar search), nếu trước đó các trang web đã phân cụm, thì khi lọc các kết quả, ta chỉ tập trung vào các trang Web nằm trong cụm có liên quan nhiều đến câu truy vấn. Như vậy, chất lượng của kết quả tìm kiếm sẽ tốt hơn. Trong phân cụm phân cấp, có thể tạo ra một hệ thống cây phân cấp các chủ đề của các trang Web, làm cho người đọc có thể tìm các trang Web theo chủ đề người đó quan tâm một cách nhanh chóng. Phân cụm cũng có thể ứng dụng vào việc nhóm các kết quả trả về của một máy tình kiếm thành các nhóm có chủ đề, và như vậy người dùng có thể tìm đến các trang Web thuộc chủ đề quan tâm một cách nhanh chóng mà không phải duyệt qua toàn bộ danh sách kết quả trả về của máy tìm kiếm. [2]

1.4 Các kiểu dữ liệu và độ đo tương tự

Trong phần này ta phân tích các kiểu dữ liệu thường được sử dụng trong PCDL. Trong PCDL, các đối tượng dữ liệu cần phân tích có thẻ là con người, nhà cửa, tiền lương, các thực thể,… Các đối tượng này thường được diễn tả dưới các dạng thuộc tính của nó. Các thuộc tính này là các tham số cần cho giải quyết vấn đề PCDL và sự lựa chọn chúng có tác động đáng kể đến các kết quả của phân cụm. Phân loại các kiểu thuộc tính khác nhau của các phần tử dữ liệu.

1.4.1 Cấu trúc dữ liệu

Các thuật toán gom cụm hầu hết sử dụng hai cấu trúc dữ liệu điển hình sau:[3]

Ma trận dữ liệu (hay cấu trúc đối tượng theo biến):Biểu diễn n đối tượng và p biến (hay còn được gọi là các phép đo hoặc các thuộc tính) của đối tượng, có dạng ma trận n hàng và p cột. Trong đó, mỗi hàng biểu diễn một đối tượng, các phần tử trong mỗi hàng chỉ giá trị thuộc tính tương ứng của đối tượng đó.

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix} \qquad (1.1)$$

Ma trận phi tương tự (cấu trúc đối tượng theo đối tượng): Lưu trữ khoảng cách của tất cả các cặp đối tượng. Biểu thị bằng ma trận n hàng và n cột. Trong đó, d(i,j) là khoảng cách hay độ khác biệt giữa các đối tượng i và đối tượng j. d(i,j) là một số không âm, d(i,j) gần tới 0 khi hai đối tượng i và j có độ tương đồng cao hay chúng "gần" nhau, d(i,j) càng lớn nghĩa là hai đối tượng i và j có độ tương đồng càng thấp hay chúng càng "xa" nhau. Do d(i,j) = d(j,i) và d(i,i)=0 nên ta có thể biểu diễn ma trận phi tương tự như sau:

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix} \quad (1.2)$$

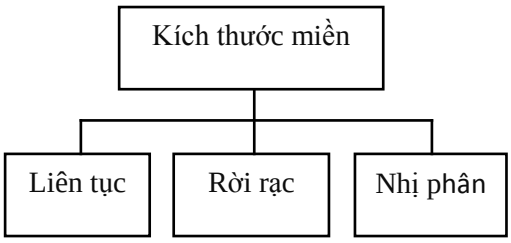
Ma trận dữ liệu thường được gọi là ma trận 2 kiểu (two-mode matrix), trong khi đó ma trận phi tương tự được gọi là ma trận 1 kiểu (one-mode matrix). Phần lớn các thuật toán phân cụm thường sử dụng cấu trúc ma trận phi tương tự. Do đó, nếu dữ liệu cần phân cụm được tổ chức dưới dạng ma trận dữ liệu thì cần biến đổi về dạng ma trận phi tương tự trước khi tiến hành phân cụm.

1.4.2 Các kiểu dữ liệu

Cho một cơ sở dữ liệu D chứa n đối tượng trong không gian k chiều; x, y, z là các đối tượng thuộc D: x = $(x_1, x_2, \dots, x_k)$; y = $(y_1, y_2, \dots, y_k)$; z = $(z_1, z_2, \dots, z_k)$. Trong đó: x_i, y_i, z_i (i = 1..k) là các đặc trưng hoặc thuộc tính tương ứng của các đối tượng x, y, z. Do đó, khái niệm "các kiểu dữ liệu" và "các kiểu thuộc tính dữ liệu" được xem là tương đương nhau.

Có hai đặc trưng để phân loại kiểu dữ liệu là kích thước miền và hệ đo.[2]

1.4.2.1 Phân loại kiểu dữ liệu dựa trên kích thước miền



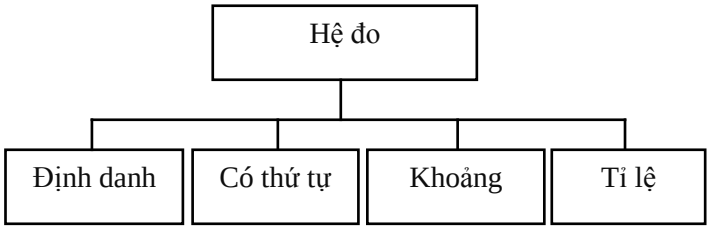
Hình 1.4 Phân loại kiểu dữ liệu dựa trên kích thước miền.

Thuộc tính liên tục (Continuous Attribute): Nếu miền giá trị của nó là vô hạn không đếm được, nghĩa là giữa hai giá trị tồn tại vô số giá trị khác. Thí dụ như các thuộc tính về màu, nhiệt độ hoặc cường độ âm thanh,...

Thuộc tính rời rạc (Discrette Attribute): Nếu miền giá trị của nó là tập hữu hạn hoặc đếm được. Thí dụ: *loại ô tô* là một thuộc tính rời rạc với tập giá trị là: {xe tải, xe khách, xe con, taxi} hay số serial của một cuốn sách, số thành viên trong một lớp,…

Thuộc tính nhị phân (Binary Attribute): Là trường hợp đặc biệt của thuộc tính rời rạc mà miền giá trị của nó chỉ có hai phần tử được diễn tả như: Yes/ No hoặc Nam/ Nữ,...

1.4.2.2 Phân loại kiểu dữ liệu dựa trên hệ đo



Hình 1.5 Phân loại kiểu dữ liệu dựa trên hệ đo.

Giả sử ta có hai đối tượng x, y và các thuộc tính của x_i, y_i tương ứng với thuộc tính thứ i của chúng. Chúng ta có các lớp kiểu dữ liệu như sau:

Thuộc tính định danh(Nominal): đây là dạng thuộc tính khái quát hoá của thuộc tính nhị phân, trong đó miền giá trị là rời rạc không phân biệt thứ tự và có nhiều hơn hai phần tử. Nếu x và y là hai đối tượng thuộc tính thì chỉ có thể xác định là $x \neq y$ hoặc $x = y$. Thí dụ như thuộc tính về nơi sinh.

Thuộc tính có thứ tự (Ordinal): là thuộc tính định danh có thêm tính thứ tự, nhưng chúng không được định lượng. Nếu x và y là hai thuộc tính thứ tự thì ta có thể xác định là $x \neq y$ hoặc $x = y$ hoặc $x > y$ hoặc $x < y$. Thí dụ như thuộc tính huy chương của vận động viên thể thao.

Thuộc tính khoảng (Interval): Dùng để đo các giá trị theo xấp xỉ tuyến tính. Với thuộc tính khoảng, chúng ta có thể xác định một thuộc tính là đứng trước hoặc đứng sau thuộc tính khác với một khoảng là bao nhiêu. Nếu $x_i > y_i$ thì ta nói x cách y một khoảng $|x_i - y_i|$ tương ứng với thuộc tính thứ i. Một thí dụ về thuộc tính khoảng như thuộc tính số serialcủa một đầu sách trong thư viện hoặc thuộc tính số kênh trên truyền hình.

Thuộc tính tỉ lệ (Ratio): là thuộc tính khoảng nhưng được xác định một cách tương đối so với điểm mốc, thí dụ như thuộc tính chiều cao hoặc cân nặng lấy điểm 0 làm mốc.

Trong các thuộc tính dữ liệu trình bày ở trên, thuộc tính định danh và thuộc tính có thứ tự gọi chung là thuộc tính hạng mục, thuộc tính tỉ lệ và thuộc tính khoảng cách được gọi là thuộc tính tham số.

1.4.3 Độ đo tương tự

Sự khác biệt hay tương tự giữa hai đối tượng được xác định qua một hàm khoảng cách giữa chúng, khoảng cách $d(x, y)$ giữa x và y cho bởi mêtric thỏa mãn các tính chất sau:[3,2]

Tính xác định dương:

$$d(x, y) \geq 0, \forall x;\ y, \tag{1.3a}$$

$$d(x, y) = 0\ khi\ v\grave{a}\ ch\hat{i}\ khi\ x = y. \tag{1.3b}$$

Tính giao hoán:

$$d(x, y) = d(y, x), \forall\ x;\ y \tag{1.3c}$$

Bất đẳng thức tam giác:

$$d(x, y) \leq d(x, z) + d(z, y), \forall\ x;\ y;\ z. \tag{1.3d}$$

Nếu không gian đặc trưng là không gian số học d-chiều và mêtric có tính chất:

$$d(ax, y) = |a|d(x, y) \tag{1.3e}$$

Sau đây là các phép đo độ tương tự áp dụng đối với các kiểu dữ liệu khác nhau:[3,2]

1.4.3.1 Thuộc tính nhị phân

Để tìm độ đo, trước hết người ta xây dựng bảng sau :

Bảng 1.1 Bảng giá trị tham số

Đối tượng y

Đối tượng x	y:1	y:0	Tổng
x:1	α	β	$\alpha + \beta$
x:0	γ	δ	$\gamma + \delta$
Tổng	$\alpha + \gamma$	$\beta + \delta$	τ

Trong đó : $\tau = \alpha + \gamma + \beta + \delta$, các đối tượng x, y mà tất cả các thuộc tính tính của nó đều là nhị phân biểu thị bằng 0 và 1. Bảng trên cho ta các thông tin sau :

- α là tổng số các thuộc tính có giá trị là 1 trong cả hai đối tượng x, y;
- β là tổng số các giá trị thuộc tính có giá trị là 1 trong x và 0 trong y;
- γ là tổng số các giá trị thuộc tính có giá trị là 0 trong x và 1 trong y;
- δ là tổng số các giá trị thuộc tính có giá trị là 0 trong x và y.

Khi đó độ đo tương tự được đo như sau:

Hệ số đối sánh đơn giản: $d(x, y) = \frac{\alpha + \delta}{\tau}$, ở đây cả hai đối tượng x và y có vai trò như nhau, nghĩa là chúng đối xứng và có cùng trọng số.

Hệ số Jacard: $d(x, y) = \frac{\alpha}{\alpha + \beta + \gamma}$, chú ý rằng tham số này bỏ qua số các đối sánh giữa 0 – 0. Công thức tính này được sử dụng trong trường hợp mà trọng số của các thuộc tính có giá trị 1 của đối tượng dữ liệu có cao hơn nhiều so với các thuộc tính có giá trị 0, như vậy các thuộc tính nhị phân ở đây là không đối xứng.

1.4.3.2 Thuộc tính định danh

Độ đo phi tương tự giữa hai đối tượng x và y được định nghĩa như sau:

$$d(x, y) = \frac{p - m}{p} \tag{1.4}$$

Trong đó: m là số thuộc tính đối sánh tương ứng trùng nhau và p là tổng số các thuộc tính.

1.4.3.3 Thuộc tính có thứ tự

Phép đo độ phi tương tự giữa các đối tượng dữ liệu với thuộc tính thứ tự được thực hiện như sau, ở đây ta giả sử i là thuộc tính thứ tự có M_i giá trị (M_i là kích thước miền giá trị):

Các trạng thái M_i được sắp thứ tự như sau: [1…M_i], chúng ta có thể thay thế mỗi giá trị của thuộc tính bằng giá trị cùng loại r_i, với $r_i \in \{1.. M_i\}$.

Mỗi một thuộc tính có thứ tự có các miền giá trị khác nhau, vì vậy chúng ta chuyển đổi chúng về cùng miền giá trị [0,1] bằng cách thực hiện phép biến đổi sau cho mỗi thuộc tính :

$$z_i^{(j)} = \frac{r_i^{(j)} - 1}{M_i - 1} \quad (1.5)$$

Sử dụng công thức tính độ phi tương tự của *thuộc tính khoảng* đối với các giá trị $z_i^{(j)}$, đây cũng chính là độ phi tương tự của thuộc tính có thứ tự.

1.4.3.4 Thuộc tính khoảng

Sau khi chuẩn hoá, độ đo phi tương tự của hai đối tượng dữ liệu x, y được xác định bằng các metric khoảng cách:

Khoảng cáchMinskowski :$d(x,y) = (\sum_{i=1}^{n}|x_i - y_i|^r)^{1/r}$, q ≥ 1. (1.7a)

Có ba khoảng cách phổ biến sử dụng khoảng cách Minskowski định nghĩa như sau:

- *Khoảng cáchEuclide* :$d(x,y) = (\sum_{i=1}^{n}|x_i - y_i|^2)^{1/2}$, (q = 2) (1.7b)
- *Khoảng cáchManhattan* :$d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$, (q = 1) (1.7c)
- *Khoảng cáchcực đại* : $d(x,y) = Max_{i=1}^{n}|x_i - y_i|$, (q → ∞). (1.7d)

Khoảng cách Euclide là chuẩn khoảng cách được dùng phổ biến nhất trong các chuẩn theo khoảng cách Minshowski.

Ngoài ra, còn có chuẩn *khoảng cách Mahalanobis*:

$$d(x,y) = (x-y)^T A(x-y) \quad (1.7e)$$

Trong đó, A là một ma trận đối xứng xác định dương.

1.4.3.5 Thuộc tính tỉ lệ

Có nhiều cách khác nhau để tính độ tương tự giữa các thuộc tính tỉ lệ. Một trong những số đó là sử dụng công thức tính logarit cho mỗi thuộc tính x_i, thí dụ $q_i = log(x_i)$, lúc này q_i đóng vai trò như thuộc tính khoảng (Interval - Scale). Phép biến đổi logarit này thích hợp trong trường hợp các giá trị của thuộc tính là số mũ.

1.5 Các phương pháp và các thuật toán phân cụm dữ liệu

Có nhiều thuật toán phân cụm dựa trên các cách tiếp cận khác nhau về tính giống nhau của đối tượng (tính tương đồng) trong cụm và có thể phân làm 4 loại chính [2]:

- Phương pháp phân cấp (Hierarchical Data Clustering);
- Phương pháp phân hoạch (Partition Based Data Clustering);

- Phương pháp dựa trên mật độ (Density Based Data Clustering);
- Phương pháp dựa trên lưới (Grid Based Data Clustering).

Trong đó, hai phương pháp phân cấp và phân hoạch là thông dụng hơn.

1.5.1 Phương pháp phân cấp

Quá trình thực hiện phân cụm theo phương pháp này được mô tả bởi một đồ thị có cấu trúc cây, vì vậy nó còn được gọi là phương pháp phân cụm cây. Trong đó, tập dữ liệu được sắp xếp thành một cấu trúc có dạng hình cây gọi là cây phân cụm. Cây này có thể được xây dựng nhờ kỹ thuật đệ quy theo hai phương pháp tổng quát: phương pháp dưới lên (bottom up) và phương pháp trên xuống (top down).

Các thuật toán theo phương pháp dưới lên còn gọi là các thuật toán trộn. Ban đầu, người ta khởi tạo mỗi đối tượng làm một cụm và dùng thủ tục đệ quy để trộn hai cụm gần nhất với nhau trong mỗi bước để có kết quả chia cụm mới. Thủ tục đệ quy kết thúc ta có tập duy nhất là toàn bộ dữ liệu. Các thuật toán phân biệt với nhau ở tiêu chuẩn đánh giá hai cụm nào là gần nhất dựa trên khoảng cách các cụm chọn trước. Quy tắc để chọn các cụm trộn này được gọi là *quy tắc liên kết*. Quá trình thực hiện thuật toán được biểu diễn thành cây và quyết định phân dữ liệu thành bao nhiêu cụm sẽ do người dùng quyết định. Người dùng cũng dựa trên cây này để nhận được kết quả phân cụm.

Cụ thể, với cách tính khoảng cách để chọn cặp cụm trộn với nhau cho trước, các thuật toán trộn bao gồm các bước sau:

1. Khởi tạo mỗi phần tử làm một cụm $c_i = \{x_i\}$, c = n
2. Khi c ≠ 1 thực hiện lặp:
   2.1. Chọn hai cụm gần nhất c_i và c_j theo quy tắc đã chọn
   2.2. Trộn c_i và c_j thành $c_{ij} = \{c_i \cup c_j\}$ // còn c-1 cụm
   2.3. c ← c-1

Phương pháp trên xuống còn gọi là phương pháp tách, được thực hiện theo trình tự ngược với phương pháp trộn. Trong mỗi bước người ta chọn một cụm để tách thành cụm con theo quy tắc đánh giá và tách cụm cho trước. Phương pháp này phức tạp và lâu hơn phương pháp dưới lên và thường chỉ được áp dụng khi người ta có thêm thông tin về phân bố cụm để có phương pháp tách phù hợp. Ta không đi sâu vào phương pháp này.

Ví dụ:

Trong ví dụ này, ta giải thiết đã có quy tắc liên kết và không bàn cụ thể tới cách chọn cụm trộn. Quá trình thực hiện phương pháp "dưới lên" phân cụm tập dữ liệu S = {a, b, c, d, e} được mô tả trong hình 1.6 cụ thể như sau:

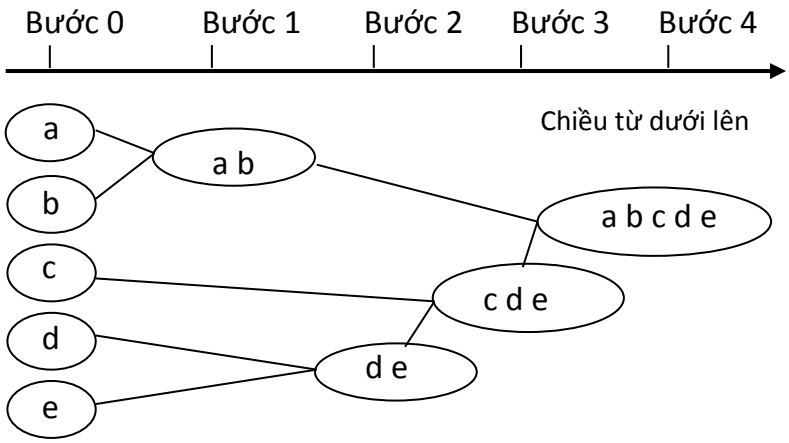
Bước 0: Mỗi đối tượng dữ liệu được gán cho mỗi cụm, như vậy các cụm ban đầu là: {a},{b},{c},{d},{e}.

Bước 1: {a} và {b} là được gộp vào thành một cụm lớn hơn là {a,b} và các cụm thu được là: {a,b},{c},{d},{e}.

Bước 2: Gộp cụm {d},{e} thành {d,e}, các cụm thu được là {a,b},{c},{d,e}.

Bước 3: Gộp cụm {c} với {d,e} thành {c,d,e}, các cụm thu được là {a,b}, {c,d,e}.

Bước 4: Gộp cụm hai cụm {c,d,e} với {a,b} thành {a,b,c,d,e}.



Hình 1.6 Phân cụm tập S = {a, b, c, d, e} theo phương pháp "dưới lên".

Các quy tắc liên kết:

Kết quả phân cụm của một thuật toán phụ thuộc vào mêtric được dùng để tính khoảng cách của các đối tượng. Kết quả phân cụm phân cấp cũng phụ thuộc quy tắc liên kết hay cách tính khoảng cách (hoặc giả khoảng cách) giữa hai cụm c_i và c_j để tìm và trộn hai cụm có khoảng cách nhỏ nhất trong mỗi bước.

Với metric trong không gian đặc trưng xác định bởi một chuẩn $\|.\|$ đã có, sau đây là một số quy tắc liên kết thông dụng.

a) Liên kết đơn

Ký hiệu là NN (Nearest Neighbour). Trong quy tắc này, khoảng cách giữa hai cụm được xác định nhờ khoảng cách nhỏ nhất giữa hai mẫu (đối tượng) tương ứng với hai cụm:

$$d(c_i, c_j) = min\{\|x - y\| : x \in c_i, x \in c_i\} \quad (1.8a)$$

b) Liên kết đầy

Ký hiệu là FN (Furthest Neighbour). Trong quy tắc này, khoảng cách giữa hai cụm được xác định nhờ khoảng cách lớn nhất giữa hai mẫu tương ứng với hai cụm:

$$d(c_i, c_j) = max\{\|x - y\| : x \in c_i, y \in c_j\} \quad (1.8b)$$

c) Liên kết trung bình giữa các nhóm

Ký hiệu là UPGMA (Un-Weighted Pair-Group Method using Arithmetic averages). Như tên gọi của nó, khoảng cách $d(c_i, c_j)$ là trung bình của khoảng cách giữa các cặp đối tượng thuộc hai cụm tương ứng:

$$d(c_i, c_j) = \frac{1}{n_i n_j} \sum_{x \in c_i} \sum_{x \in c_j} \|x - y\| \quad (1.8c)$$

Trong đó:n_i và n_j là số phần tử của các cụm c_i, c_j tương ứng.

d) Liên kết trung bình trong phạm vi nhóm

Ký hiệu là UWGMA (un-weighted within-group method using arithmetic averages). Trong quy tắc này, khoảng cách $d(c_i, c_j)$ là trung bình của khoảng cách giữa các đối tượng trong nhóm mới sau khi đã trộn hai nhóm:

$$d(c_i, c_j) = \frac{1}{C(n_{i+}n_j, 2)} \sum_{x,y \in c_i \cup c_j} \|x - y\| \quad (1.8d)$$

e) Phương pháp Ward

Trong phương pháp này, khoảng cách giữa hai cụm là trung bình của bình phương khoảng cách tới tâm trong phạm vi cụm:

$$d(c_i, c_j) = \frac{1}{n_{i+}n_j} \sum_{x,y \in c_i \cup c_j} \|x - m\|^2 \quad (1.8e)$$

Trong đó: m là tâm của cụm trộn.

1.5.2 Phương pháp phân hoạch

Trong các phương pháp phân hoạch, với số lượng cụm đã định, người ta lần lượt phân các đối tượng dữ liệu vào các cụm, sau đó thực hiện lặp quá trình điều chỉnh để cực tiểu hàm mục tiêu được chọn. Thông dụng nhất là thuật toán k-mean

và các biến thể của nó. Trong các thuật toán này, số lượng cụm k thường được xác định trước hoặc đặt dưới dạng tham số. Với tập dữ liệu D gồm n đối tượng trong không gian d chiều, các đối tượng được phân thành k cụm sao cho tổng bình phương độ lệch của mỗi mẫu tới tâm của nó là nhỏ nhất. Sau đây là thuật toán k-means, thuật toán điển hình của phương pháp này.

Thuật toán k-means

Thuật toán k-means (MacQueue, 1967) chia tập dữ liệu D cho trước thành k cụm $\{c_1, c_2, \dots, c_k\}$, sao cho tổng bình phương khoảng cách của mỗi đối tượng dữ liệu tới tâm cụm chứa nó đạt cực tiểu. Như vậy, hàm mục tiêu của thuật toán này là:

$$E = \sum_{i=1}^{k} \sum_{x \in c_i} \|x - v_i\|^2 \quad (1.9)$$

Trong đó: v_i là tâm của cụm c_i tương ứng.

Thuật toán này thực hiện như sau:

Bước 0: Xác định trước số lượng cụm k và điều kiện dừng;

Bước 1: Khởi tạo ngẫu nhiên k điểm $\{v_i\}_{i=1}^{k}$ làm các tâm cụm;

Bước 2: Lặp khi điều kiện dừng chưa thỏa mãn:

2.1. Phân hoạch D thành k cụm bằng cách gán mỗi đối tượng vào cụm mà nó gần tâm nhất;

2.2. Tính lại các tâm theo các đối tượng đã được phân hoạch ở bước 2.1.

Điều kiện dừng của thuật toán thường chọn từ các điều kiện sau:

- Số lần lặp t = t_{max} , trong đó t_{max} là số cho trước;
- Giá trị của hàm E nhỏ hơn một ngưỡng nào đó (đảm bảo chất lượng của các cụm đủ tốt, hay nó đã chạy được đủ số vòng lặp cần thiết);
- Tới khi các cụm không đổi.

Khi tập dữ liệu không quá lớn thì người ta dùng điều kiện dừng 3.

Nếu tập dữ liệu D gồm n mẫu và số lần lặp ở bước 2 là t thì độ phức tạp của thuật toán chỉ là O(tnk) nên rất thích hợp khi tập D gồm lượng dữ liệu lớn.

1.5.3 Phương pháp dựa trên mật độ

Hầu hết các phương pháp phân hoạch truyền thống đều phân cụm chỉ dựa trên khoảng cách giữa các đối tượng. Chúng chủ yếu tìm ra các giới hạn cụm có dạng hình cầu và rất khó để tìm ra các cụm có hình dạng ngẫu nhiên. Phương pháp phân

cụm dựa vào mật độ xem các cụm như là các vùng có mật độ các đối tượng lớn trong không gian dữ liệu. Các phương pháp dựa vào mật độ có thể sử dụng để loại bỏ nhiễu và phát hiện ra các cụm có hình dạng tự nhiên.

Thuật toán dựa vào mật độ đầu tiên là thuật toán DBSCAN (Ester et al, 1996), thuật toán này xem xét mật độ theo lân cận của mỗi đối tượng, nếu số lượng các đối tượng trong khoảng cách ε của một đối tượng lớn hơn ngưỡng MinPts thì đối tượng đó được xem là nằm trong một cụm. Bởi vì các cụm tìm được phụ thuộc vào tham số ε và MinPts, nên thuật toán DBSCAN cần dựa vào người sử dụng để lựa chọn tập tham số tốt. Để tránh được vấn đề này, năm 1999 Ankerst đề xuất phương pháp sắp xếp các cụm gọi là OPTICS (Ordering Point To Identify the Clustering Structure). OPTICS tính toán việc sắp xếp các cụm có tham số để phân cụm tự động. Nhược điểm của các thuật toán theo hướng này là có độ phức tạp lớn nên không dùng được cho khối lượng dữ liệu lớn. Thuật toán DBSCAN giúp ta hiểu được cách tiếp cận này.

Thuật toán DBSCAN (Density – Based Spatial Clustering of Applications with Noise)

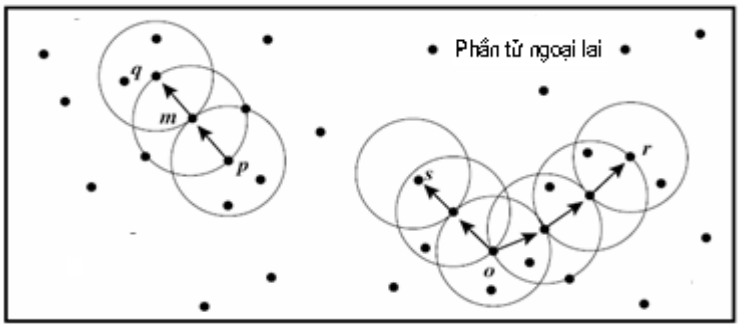
Thuật toan DBSCAN nhóm các vùng có mật độ đủ cao vào trong một cụm và thác triển dựa trên các đối tượng lõi để có các cụm với hình dạng tự nhiên trong các tập không gian đặc trưng. Thuật toán yêu cầu xác định trước hai tham số đầu vào là ε và Minpts. Phân cụm dữ liệu theo thuật toán DBSCAN áp dụng các luật sau đây:

- Các đối tượng nằm trong hình cầu bán kính ε (ε–lân cận) của một đối tượng được gọi là ε–láng giềng của đối tượng đó. Đối tượng có ít nhất là Minpts đối tượng khác là ε–láng giềng thì được gọi là đối tượng nhân.
- Một đối tượng có thể nằm trong một cụm khi và chỉ khi nó nằm trong ε–lân cận của một đối tượng nhân thuộc cụm đó.
- Một đối tượng lõi o là ε–láng giềng của một đối tượng nhân p thì o thuộc cùng cụm với p.
- Hai cụm có giao khác rỗng thì nhập thành một cụm
- Một đối tượng không là nhân r và không là ε–láng giềng của một đối tượng nhân nào thì được xem là phần tử ngoại lai hay là đối tượng nhiễu.

Để lập nên các cụm, DBSCAN kiểm tra ε–láng giềng của mỗi đối tượng trong cơ sở dữ liệu. Nếu ε–láng giềng của một điểm p chứa nhiều hơn Minpts, một cụm mới với p là đối tượng nhân được tạo ra. Các cụm này được mở rộng nhờ liên kết các cụm con tạo nên cụm chứa nó. Những phần tử ngoại lai không được phân cụm, nếu cần thiết thì sau khi phân cụm cụm con hình thành bởi các đối tượng nhân, ta phát triển được thành các cụm có hình dạng phong phú.

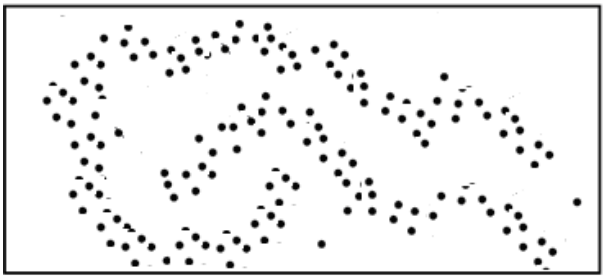
Ví dụ:

Hình 1.7 minh họa một trường hợp với ε là bán kính của hình tròn và Minpts = 3, tập dữ liệu gồm hai cụm và các phần tử ngoại lai rải rác. Các đối tượng {o, p, q, r} là nhân còn s không là đối tượng nhân nhưng nó thuộc cụm vì là ε–láng giềng của một đối tượng là nhân.



Hình 1.7 Hai cụm được tìm bởi thuật toán DBSCAN.

Hình 1.8 minh họa một ví dụ về tập dữ liệu gồm hai cụm được nhận biết nhờ phương pháp này mà không dùng phương pháp phân hoạch được.



Hình 1.8 Hai cụm dữ liệu có thể tìm được nhờ DBSCAN.

1.5.4 Phương pháp dựa trên lưới

Khi dữ liệu thuộc không gian có số chiều lớn, không trực quan hóa được thì việc xác định các tham số ε và Minpts cho các phương pháp phân cụm dựa vào mật độ rất khó khăn, hơn nữa với số lượng dữ liệu lớn thì mất nhiều thời gian chạy. Để nâng cao hiệu quả của phân cụm, một cách tiếp cận là phân chia miền không gian đặc trưng chứa dữ liệu thành một số hữu hạn các ô tạo nên dạng hình lưới và sử dụng các đặc trưng thống kê để phân tích các dữ liệu trong mỗi ô và quyết định tách hay nhập chúng. Bỏ qua nội dung thống kê, ta làm quen với thuật toán STING để hiểu các tiếp cận này.

Thuật toán STING (A STatistical Information Grid approach)

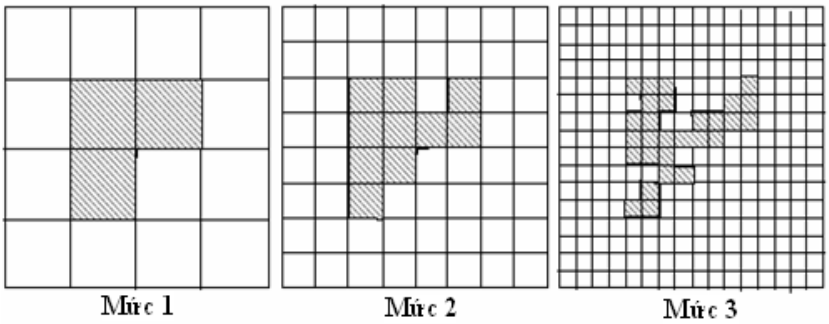
STING do W. Wang và các cộng sự (1997) đề xuất, phương pháp này tổ chức miền không gian chứa dữ liệu thành lưới hình hộp đa mức để phân tích cụm theo

thống kê phân cấp trên từng ô. Ban đầu ta chia miền dữ liệu thành các ô hình chữ nhật (hoặc hình hộp khi không gian có số chiều cao) với chiều dài các cạnh ở mức 1. Việc phân tích thông tin dựa trên các đặc điểm thống kê của tập dữ liệu trong mỗi ô như:

- Count: số đối tượng trong ô;
- M: vectơ trung bình của dữ liệu trong ô;
- S: độ lệch chuẩn của mọi giá trị thuộc tính trong ô;
- Min: giá trị cực tiểu của các thuộc tính trong ô;
- Max: giá trị cực đại của các thuộc tính trong ô;
- Distribution: kiểu phân phối của các giá trị thuộc tính trong ô.

Việc phân tích này giúp ta quyết định có chia ô đang xét ở mức mịn hơn không hay là đã đủ để phân cụm trong từng ô hoặc kết hợp với các cụm ở ô liền kề. Cách phân chia ô như vậy tạo ra một cấu trúc phân cấp: mỗi ô ở mức cao được phân chia thành một số ô ở mức thấp hơn trong bước tiếp theo.

Hình 1.9 mô tả 3 mức lưới liên tiếp nhau trong cấu trúc STING, mỗi ô ở mức trên được phân thành bốn ô ở mức tiếp theo. Các tham số thống kê ở mức cao khi chưa xác định được sẽ được tính toán từ các tham số trong các ô ở mức thấp hơn. Kiểu phân bố ở ô mức cao được tính toán dựa trên các kiểu phân bố ở các ô tương ứng ở mức thấp. Nếu các phân bố ở mức thấp không cho biết phân bố mức cao thì phân bố ở ô mức cao sẽ là không xác định (được đặt là none).



Hình 1.9 Ba tầng liên tiếp nhau của cấu trúc STING.

Việc phân tích thống kê thực hiện phân cấp theo các ô từ tầng trên. Tầng này bao gồm một số lượng nhỏ các ô. Với mỗi ô trong tầng, tính khoảng chắc chắn mà các ô trong đó sẽ trở thành một cụm để quyết định. Các ô không chắc chắn sẽ phân chia tiếp hoặc loại bỏ. Tiến trình này được lặp lại cho đến khi tính chất cụm của dữ liệu trong mỗi ô xác định rõ. Việc phân cụm sẽ hoàn tất khi xác định được quan hệ cụm giữa dữ liệu trong các ô.

Chương II
LÝ THUYẾT TẬP THÔ

2.1 Giới Thiệu

Ngay từ khi xuất hiện, lý thuyết tập thô do Zdzisaw Pawlak khởi xướng vào những năm đầu thập niên tám mươi của thế kỷ hai mươi đã ngay lập tức thu hút sự quan tâm của nhiều nhà nghiên cứu và thực nghiệm trên toàn thế giới. Khả năng ứng dụng trong nhiều lĩnh vực khác nhau cho thấy vai trò quan trọng của lý thuyết này trong việc nghiên cứu và ứng dụng công nghệ thông tin trong thời đại mới.

Lý thuyết tập thô có thể được xem xét theo hai phương diện là mô hình và thực hành. Theo phương diện mô hình, lý thuyết tập thô cho một cách tiếp cận mới cho tính mơ hồ. Các khái niệm mơ hồ được đặc trưng bởi một "miền biên" chứa tất cả các phần tử mà không thể gộp vào miền các đối tượng quan sát hoặc phần bù của miền này. Lý thuyết tập thô được nghiên cứu và phát triển nhằm hiểu tốt hơn ý tưởng của tính mơ hồ. Nó cũng xét đến một vài ý tưởng của Gottfried Leibniz (tính không phân biệt được), George Boole (các phương pháp suy luận), Jan Lukasiewicz (các logic đa trị) và Thomas Bayes (suy luận quy nạp). Về phương diện thực hành, lý thuyết tập thô là ý tưởng nền tảng cho trí tuệ nhân tạo và khoa học nhận thức, đặc biệt cho học máy, phát hiện tri thức, phân tích quyết định, suy luận quy nạp và nhận dạng mẫu. Nó là rất quan trọng cho các nghiên cứu về hệ trợ giúp quyết định và khai phá dữ liệu. Thực tế tiếp cận lý thuyết tập thô là một cách tiếp cận mới cho việc phân tích dữ liệu.

Mục đích chính của sự phân tích tập thô là đưa ra các tập xấp xỉ để biểu diễn các đối tượng không thể được phân lớp một cách chắc chắn bằng cách dùng tri thức có sẵn. Theo cách tiếp cận của lý thuyết tập thô, mọi tập thô được liên kết với hai tập "rõ" là xấp xỉ dưới và xấp xỉ trên của nó. Xấp xỉ dưới bao gồm các đối tượng chắc chắn thuộc, còn xấp xỉ trên chứa tất cả các đối tượng có khả năng thuộc về tập đó. Các tập xấp xỉ là cơ sở để đưa ra các kết luận từ dữ liệu.

2.2 Các khái niệm cơ bản

2.2.1 Hệ thống thông tin

Một tập dữ liệu có thể biểu diễn dưới dạng một bảng, trên đó mỗi dòng biểu diễn thông tin ứng với một đối tượng, mỗi cột biểu diễn một thuộc tính có thể đo được của đối tượng. Bảng này được gọi là một hệ thống thông tin.

Hệ thống thông tin là một cặp $IS = (U, A)$, với U là tập hữu hạn, khác rỗng, được gọi là tập vũ trụ các đối tượng và A là tập hữu hạn khác rỗng các thuộc tính. Với $mỗi\ u \in U$ và $a \in A$, ta ký hiệu u(a) là giá trị của đối tượng u tại thuộc tính a. Nếu gọi Va là tập tất cả các gía trị của thuộc tính a, thì $u(a) \in Va$ với mọi $u \in U$. Bây giờ, nếu $B = \{b1, b2, \cdots, bk\} \subseteq A$ là một tập con các thuộc tính thì ta sẽ ký hiệu bộ các giá trị u(b_i) bởi u(B). Như vậy, nếu u và v là hai đối tượng, thì ta sẽ viết $u(B) = v(B)$ nếu $u(bi) = v(bi)$, với mọi $i = 1, \cdots, k$.

Ví dụ 2.2.1: Một hệ thống thông tin bao gồm 8 đối tượng U={u1, u2, u3, u4, u5, u6, u7, u8}, tập thuộc tính A={Color, Size}, và miền giá trị cho từng thuộc tính là IColor = {Green, Yellow, Red}, ISize = {Small, Medium, Big}.

Bảng 2.1 Hệ Thống Thông Tin

	Color	Size
u1	Green	Big
u2	Green	Small
u3	Yellow	Medium
u4	Red	Medium
u5	Yellow	Medium
u6	Green	Big
u7	Red	Small
u8	Red	Small

2.2.2 Bảng quyết định (Decision Table)

Để có thể biểu diễn một dữ liệu thực tế, trong đó có những thuộc tính quyết định, chúng ta xét một trường hợp đặc biệt của hệ thông tin được gọi là bảng quyết định được định nghĩa như sau

Định nghĩa 1.2[4]: Bảng quyết định là một hệ thống thông tin có dạng $DT = (U, A \cup \{d\})$ Trong đó: $d \notin A$ là thuộc tính phân biệt, được gọi gọi là thuộc tính quyết định. Các thành phần của A được gọi là các thuộc tính điều kiện.

Ví dụ 2.2.2: Bảng sau đây là một bảng quyết định, Bảng này có 8 đối tượng như trong bảng 1, nhưng có thêm thuộc tính quyết định (Shape). Trong bài toán phân lớp thì thuộc tính quyết định chính là lớp của đối tượng cần xếp lớp. Trong ví dụ này thuộc tính quyết định Shape có 3 giá trị là Circle, square và Triangle.

Bảng 2.2 Ví dụ một bảng quyết định

	Color	Size	Shape[D]
u1	Green	Big	Circle
u2	Green	Small	Circle
u3	Yellow	Medium	Square
u4	Red	Medium	Square
u5	Yellow	Medium	Triangle
u6	Green	Big	Circle
u7	Red	Small	Triangle
u8	Red	Small	Triangle

Chúng ta giả sử rằng tập các giá trị của giá trị quyết định d tương đương với tập {1, . . ., r(d)} là các số nguyên dương từ 1 đến r(d), tập này được gọi là phạm vi của thuộc tính quyết định d.

Lớp quyết định thứ k (ký hiệu là Ck) là một tâp các đối tượng thoả mãn: Ck ={u $\in U$: d(u)=k}. Trong đó 1≤ k ≤r(d).

Khi đó giá trị quyết định d sẽ chia tập các đối tượng thành r(d) lớp quyết định:{C1,..., Cr(d)}.

Trong trường hợp tổng quát thì có thể có nhiều thuộc tính quyết định, khi dó bảng quyết định có dạng $DT = (U, C \cup D)$, trong đó:

$A = C \cup D$

C: gọi là tập thuộc tính điều kiện.

D: được gọi là tập thuộc tính quyết định.

2.2.3 Quan hệ không phân biệt được

Một trong những đặc điểm cơ bản của lý thuyết tập thô là dùng để lưu giữ và xử lý các dữ liệu không phân biệt được. Trong một hệ thông tin theo định nghĩa trên cũng có thể có những đối tượng không phân biệt được. Trước tiên ta nhắc lại định nghĩa quan hệ tương đương như sau:

Định nghĩa 1.5[4] Một quan hệ hai ngôi (quan hệ nhị phân) $R \subseteq U \times U$ trên U là một quan hệ tương đương khi nó có 3 tính chất:

- Phản xạ: Mọi đối tượng đều quan hệ với chính nó.
- Đối xứng: Nếu xRy thì yRx.
- Bắc cầu: Nếu xRy và yRz thì xRz.

Quan hệ tương đương R sẽ chia tập các đối tượng U thành các lớp tương đương. Lớp tương đương của phần tử $x \in U$, ký hiệu là $[x]R$ chứa tất cả các đối tượng y mà xRy.

Bây giờ bắt đầu định nghĩa một quan hệ tương đương trên hệ thống thông tin. Quan hệ này sau này được sử dụng để biểu diễn những thông tin không phân biệt được.

Định nghĩa 1.6 [4] cho tập con các thuộc tính B ⊂ A trong hệ thống thông tin (U,A). Quan hệ B – không phân biệt được (Ký hiệu $IND_A(B)$), được định nghĩa như sau:

$$IND_A(B) = \{(x,x') \in U^2 \mid \forall a \in B, a(x)=a(x')\}$$

Khi đó $IND_A(B)$ là một quan hệ không phân biệt được trên B được ký hiệu là $[x]_B$. Hai đối tượng x, x' mà $(x,x') \in IND_A(B)$ được gọi là không phân biệt được bởi các thuộc tính trong B. Khi xét trên một hệ thống thông tin xác định ta sẽ viết IND(B) thay cho $IND_A(B)$.

Ví dụ 2.2.3:Tập thuộc tính B = {Color, Size} trong bảng 2 phân hoạch 8 đối tượng thành các lớp tương đương như sau:

IND(B) = {(u1,u6),(u2),(u3,u5),(u4),(u7,u8)}

Nhận xét: Ta thấy, các đối tượng u1 và u6 cùng một lớp tương đương nên chúng không thể phân biệt với nhau trên tập thuộc tính {Color, Size}.

2.2.4 Các khái niệm xấp xỉ trong tập thô

2.2.4.1 Xấp xỉ dưới, xấp xỉ trên

Định nghĩa 1.7 [4] cho bảng quyết định $DT = (U, C \cup D)$ và tập thuộc tính $B \subset C, X \subseteq U$. Xấp xỉ dưới của tập X tương ứng với B, Ký hiệu theo thứ tự $\underline{B}X$ và $\overline{B}X$ được định nghĩa như sau:

$$\underline{B}X = \{x \in U: [x]B \subset X\},$$

$$\overline{B}X = \{x \in U: [x]B \cap X \neq \emptyset\}.$$

Tập hợp $\underline{B}X$ là tập các đối tượng trong U mà sử dụng các thuộc tính trong B ta có thể biết chắc chắn chúng là phần tử của X.

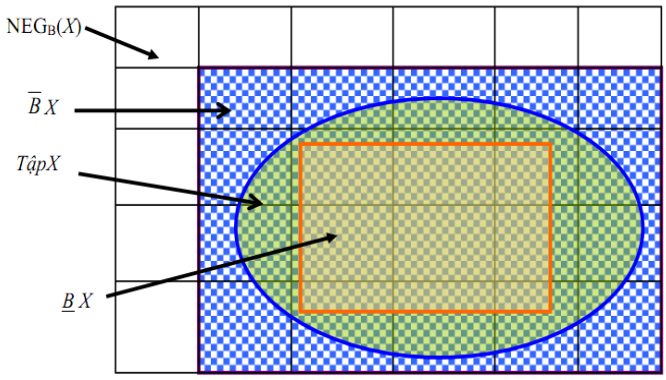
Tập hợp $\overline{B}X$ là tập các đối tượng trong U mà sử dụng các thuộc tính trong B ta chỉ có thể nói rằng chúng có thể là các phần tử của X.

2.2.4.2 Miền biên, miền ngoài

B – biên của tập X, ký hiệu $BNB(X)$, được định nghĩa $BNB(X) = \overline{B}X \setminus \underline{B}X$. $BNB(X)$ chứa những đối tượng mà sử dụng các thuộc tính trong B ta không thể xác định được chúng có thuộc X hay không.

B – ngoài của tập X, ký hiệu $NEG_B(X)$ được định nghĩa $NEG_B(X) = U \setminus \overline{B}X$. $NEG_B(X)$ chứa những đối thượng mà sử dụng các thuộc tính trong B ta biết chắc chắn không thuộc X.

Hình sau trình bày sự mô tả về tập xấp xỉ và miền.



Hình 2.1 Mô tả về tập xấp xỉ và miền

2.2.4.3 Một số tính chất của tập hợp xấp xỉ[1]

1. $\underline{B}(X) \subseteq X \subseteq \overline{B}(X)$

2. $\underline{B}(\emptyset) = \overline{B}(\emptyset) = \emptyset, \underline{B}(U) = \overline{B}(U) = U$

3. $\overline{B}(X \cup Y) = \overline{B}(X) \cup \overline{B}(Y)$

4. $\underline{B}(X \cap Y) = \underline{B}(X) \cap \underline{B}(Y)$

5. Nếu $X \subseteq Y$ thì $\underline{B}(X) \subseteq \underline{B}(Y), \overline{B}(X) \subseteq \overline{B}(Y)$

6. $\underline{B}(X \cup Y) \supseteq$ thì $\underline{B}(X) \cup \underline{B}(Y)$

7. $\overline{B}(X \cap Y) \subseteq \overline{B}(X) \cap \overline{B}(Y)$

8. $\underline{B}(U \backslash X) = U \backslash \overline{B}(X)$

9. $\overline{B}(U \backslash X) = U \backslash \underline{B}(X)$

10. $\underline{B}(\underline{B}(X)) = \overline{B}(\underline{B}(X)) = \underline{B}(X))$

11. $\overline{B}(\overline{B}(X)) = \underline{B}(\overline{B}(X)) = \overline{B}(X))$

Người ta phân tập thô thành 4 loại[4]:

- X là xác định thô thực sự theo B nếu $\underline{B}(X) \neq \emptyset$ và $\overline{B}(X) \neq U$.
- X là không xác định bên trong theo B nếu $\underline{B}(\mathrm{X}) = \emptyset$ và $\overline{B}(X) \neq U$.
- X là không xác định bên ngoài theo B nếu $\underline{B}(X) \neq \emptyset$ và $\overline{B}(X) = U$.

- X là không xác định thực sự theo B nếu $\underline{B}(X) = \emptyset$ và $\overline{B}(X) = U$.

Các khái niệm trên có thể diễn tả như sau:

- Nếu X xác định thô thực sự theo B nghĩa là sử dụng thuộc tính B chúng ta có thể quyết định rằng một số đối tượng của U thuộc tập X và một số đối tượng của U thuộc $U \setminus X$.
- Nếu X là không xác định nội tại bên trong theo B có nghĩa là sử dụng thuộc tính B chúng ta có thể quyết định rằng một số phần tử của U thuộc $U \setminus X$ nhưng không thể chỉ ra được các đối tượng thuộc X.
- Nếu X là không xác định bên ngoài theo B có nghĩa là sử dụng tập thuộc tính B chúng ta có thể quyết định rằng một số phần tử của U thuộc X nhưng không chỉ ra được các đối tượng thuộc $U \setminus X$.
- Nếu X là không xác định thực sự theo B có nghĩa là sử dụng tập thuộc tính B chúng ta không thể chỉ ra bất kỳ đối tượng nào của U có thuộc X hay $U \setminus X$.

2.2.4.4 Độ đo liên quan biên xấp xỉ [1,8]

Tập thô được chỉ số hóa như sau:

$$\alpha_B(X) = \frac{|\underline{B}(X)|}{|\overline{B}(X)|},$$

$\alpha_B(X)$ được gọi là độ đo liên quan biên xấp xỉ của X, với $|X|$ biểu diễn lực lượng của $X \neq \emptyset$. Có hể thấy được $0 \leq \alpha_B(X) \leq 1$. Nếu $\alpha_B(X) = 1$ thì X đúng hoàn toàn đối với B, ngược lại nếu $\alpha_B(X) < 1$ thì X là *thô* đối với B.

2.3. Rút gọn các thuộc tính trong hệ thống thông tin.

Thông tin trong các hệ thống có thể dư thừa, các dư thừa có thể xảy ra[4]:

Trường hợp 1: Các đối tượng giống nhau theo một tập thuộc tính đang quan tâm được lặp lại nhiều lần.

Trường hợp 2: Một số thuộc tính có thể bỏ đi mà thông tin chúng ta đang quan tâm do bảng quyết định cung cấp vẫn không bị mất mát.

Với trường hợp 1: khái niệm lớp tương đương cho ta tiếp cận tinh giảm thông tin cần lưu trữ trong một hệ thông tin. Ta chỉ cần sử dụng một đối tượng để đại diện cho mỗi lớp tương đương.

Với trường hợp 2: Chỉ giữ lại những thuộc tính bảo toàn quan hệ bất khả phân biệt, do đó bảo toàn khả năng xấp xỉ tập hợp trong một hệ thông tin. Quá trình rút gọn một hệ thống thông tin mà tập các thuộc tính của hệ thống thông tin đã được rút gọn là độc lập và không còn thuộc tính nào có thể bị loại bỏ hơn nữa mà không làm mất thông tin từ hệ thống, kết quả được biết đến như là tập rút gọn. Nếu một thuộc tính từ tập con $B \subseteq A$ duy trì mối quan hệ không phân biệt được $IND(A)$ thì các thuộc tính $A \backslash B$ là không cần thiết. Các tập rút gọn cũng là tập con tối thiểu, nghĩa là không chứa các thuộc tính không cần thiết. Do đó việc rút gọn có khả năng phân loại các đối tượng mà không làm thay đổi hình thức của việc diễn tả tri thức.

Thuộc tính cần thiết và không cần thiết.

Xét bảng quyết định $DT = (U, C \cup D)$.

Thuộc tính $c \in C$ được gọi là không cần thiết trong DT nếu $POS_{C}(D) = POS_{(C-\{c\})}(D)$. Ngược lại ta nói c là cần thiết trong DT với *Tập* $POS_{C}(D)$ *được gọi là* C*- miền khẳng định của* D.

Rõ ràng thuộc tính không cần thiết không làm tăng hay giảm khả năng phân loại khi có hoặc không có mặt thuộc tính đó trong C.

Khi loại khỏi C một số thuộc tính có thể bỏ được thì ta được một tập rút gọn của C.

Ta nói bảng quyết định $DT = (U, C \cup D)$ là độc lập nếu tất cả các thuộc tính $c \in C$ đều cần thiết trong DT.

Rút gọn và lõi:[4]Tập thuộc tính $R \subseteq C$ được gọi là một rút gọn của C nếu $DT' = (U, R \cup D)$ là độc lập và $POS_{R}(D) = POS_{C}(D)$.

Một tập rút gọn là một tập con các thuộc tính duy trì các đặc tính cơ bản của tập dữ liệu gốc, do đó các thuộc tính không thuộc về một tập rút gọn là không cần thiết đối với sự phân loại các phần tử của tập vũ trụ.

Tập tất cả các thuộc tính cần thiết trong DT kí hiệu: $CORE(C)$. Khi đó,

$CORE(C) = \cap RED(C)$ với $RED(C)$: Là tập tất cả các rút gọn của C.

2.4 Ma trận phân biệt và hàm phân biệt

Phần trên cung cấp các khái niệm về rút gọn thuộc tính trong hệ thông tin, tuy nhiên chúng chưa thực sự rõ nét và trực quan. Trong phần này chúng ta sẽ thấy bản chất của một rút gọn của tập thuộc tính và đây là cơ sở để hiểu được các thuật toán rút gọn trong một hệ thông tin.[1]

Xét hệ thống thông tin $\boldsymbol{A} = (U, A)$

Ma trận phân biệt của $\boldsymbol{A}$ ký hiệu là $M(\boldsymbol{A})$ là một ma trận đối xứng n x n với phần tử c_{ij} cho như sau:

$$c_{ij} = \begin{cases} \{a \in A: a(x_i) \neq a(x_j)\} & \text{nếu } \exists d \in D\,[d(x_i) \neq d(x_j)] \\ \lambda & \text{nếu } \forall d \in D\,[d(x_i) = d(x_j)] \end{cases}$$

Với $1 \leq j \leq i \leq n$ thì x_i, y_j thuộc A – vùng khẳng định của D. c_{ij} là tập tất cả các thuộc tính điều kiện mà phân loại x_i, x_j thành các lớp khác nhau. Hàm phân biệt được f_A cho một hệ thống thông tin **A** là một hàm kiểu Boolean của m biến logic $a_1^*, \ldots, a_m^*$ (tương ứng với các thuộc tính $a_1, \ldots, a_m$) được xác định như sau:

với $c_{ij} = \{a^* | a \in c_{ij}\}$ $f_A(a_1^*, \ldots, a_m^*) = \wedge\{\vee c^*_{ij} | 1 \leq j \leq i \leq n, c_{ij} \neq \emptyset\}$

với $\vee c_{ij} = \perp$(false) nếu $c_{ij} \neq \emptyset$; $\vee c_{ij}$= T(true) nếu $c_{ij} = \lambda$

Bảng 2.3 Ví dụ cho bảng thông tin[1]:

U	ĐTB	PTTH	Quận huyện	Trường thi	Trúng tuyển
X_1	6.7	Hai Bà Trưng	Hai Bà Trưng	Kinh tế	Trượt
X_2	7.8	Chu Văn An	Ba Đình	HVKTQS	Đỗ
X_3	6.5	Đoàn Thị Điểm	Cầu Giấy	Bách Khoa	Đỗ
X_4	6.5	Đoàn Thị Điểm	Cầu Giấy	HVKTQS	Trượt
X_5	7.5	Chuyên Ngữ	Cầu Giấy	HVKTQS	Xem xét

Bảng 2.4 Ma trận phân biệt được biểu diễn như sau:

	X_1	X_2	X_3	X_4	X_5
X_1					
X_2	ĐTB,PTTH,Quận huyện,Trường thi				
X_3	ĐTB,PTTH,Quận huyện,Trường thi				
X_4		ĐTB, PTTH,Quậnhuyện	Trường thi		
X_5	ĐTB,PTTH,Quận huyện,Trường thi	ĐTB,PTTH,Quận huyện	ĐTB,PTTH, Trườngthi	ĐTB,PTTH	

2.5 Hàm Thành Viên Thô

Trong lý thuyết tập hợp cổ điển, mỗi thành viên thuộc một tập hợp hoặc không. Hàm thành viên (hàm thuộc) là hàm đặc trưng của tập hợp nhận một trong hai giá trị 0 và 1. Trong tập thô, ý tưởng của hàm thành viên thì khác, hàm thành viên thô xác định mức độ giao nhau liên quan giữa tập X và lớp tương đương $[x]_B$ chứa x, nó được định nghĩa như sau:

μ_X^B:U → [0,1] và được xác định $\mu_X^B(x) = \frac{|[x]B \cap X|}{|[x]B|}$

Một số tính chất của hàm thành viên thô[4]:

1. $\mu_X^B(x) = 1 \Leftrightarrow x \in \underline{B}(X)$
2. $\mu_X^B(x) = 0 \Leftrightarrow x \in U - \overline{B}(X)$
3. $0 < \mu_X^B(x) < 1$ $x \in BN_B(X)$
4. $\mu_X^B(x) = \mu_X^B(y)$ nếu $(x, y) \in IND(B)$
5. $\mu_{U-X}^B(x) = 1 - \mu_X^B(x), \forall x \in U$
6. $\mu_{X \cup Y}^B(x) = \max(\mu_X^B(x), \mu_Y^B(x)) \forall x \in U$
7. $\mu_{X \cap Y}^B(x) = \min(\mu_X^B(x), \mu_Y^B(x)) \forall x \in U$

Kết luận chương:

Lý thuyết tập thô đang được nhiều nhà khoa học nghiên cứu và sử dụng trong quá trình khám phá tri thức từ dữ liệu. Các khái niệm nền tảng trong lý thuyết tập thô là hệ thông tin, bảng quyết định, quan hệ không phân biệt được, tập xấp xỉ và sự phụ thuộc thuộc tính. Xấp xỉ trên và dưới trong lý thuyết tập thô mở ra một hướng nghiên cứu mới trong khai phá dữ liệu.

Chương III
ÁP DỤNG THUẬT TOÁN PHÂN CỤM THÔ VÀO BÀI TOÁNPHÂN CỤM NGƯỜI DÙNG TRÊN WEB

3.1 Giới Thiệu

Phân cụm là bước khởi đầu và cơ bản trong phân tích dữ liệu. Phân cụm đã được nghiên cứu trong lĩnh vực học máy và nhận dạng mẫu và đóng một vai trò quan trọng trong các ứng dụng khai thác dữ liệu như thăm dò dữ liệu khoa học, thông tin và khai thác văn bản.

Nó cũng đóng một vai trò quan trọng trong các ứng dụng cơ sở dữ liệu về không gian, phân tích web, quản lý quan hệ khách hàng, tiếp thị, Sinh học, điện toán và nhiều lĩnh vực khác có liên quan.

Các thuật toán Phân cụm đã được phân loại sử dụng nguyên tắc phân loại khác nhau dựa trên các vấn đề quan trọng như cấu trúc thuật toán, bản chất của cụm hình thành, sử dụng bộ tính năng,...

Nói chung, các thuật toán phân nhóm có thể được chia thành hai loại – Partitional(phân vùng) và phân cấp. Các thuật toán Partitional xây dựng một phân vùng của một cơ sở dữ liệu D của n đối tượng vào một tập hợp các cụm k, với k là một tham số đầu vào cho các thuật toán. Để thiết lập giá trị của k, một số kiến thức miền được yêu cầu mà không may không có sẵn cho nhiều ứng dụng.

Các nhóm này được liên tục kết hợp dựa trên một độ đo khoảng cách, cho đến khi chỉ có một nhóm còn lại hoặc kết thúc. Trong phân chia phân nhóm theo cấp bậc, chúng ta bắt đầu với việc tất cả các dữ liệu trong một cụm lớn và dần dần chia chúng thành các cụm nhỏ hơn dựa trên các độ đo khoảng cách.

Một cụm thô được định nghĩa một cách tương tự như một tập thô. Xấp xỉ dưới của một cụm thô chứa các đối tượng mà nó thuộc về nhóm đó. Xấp xỉ trên của một cụm thô chứa các đối tượng trong nhóm này cũng là thành viên của Các cụm khác. Lợi thế của việc sử dụng bộ thô là không giống như các kỹ thuật khác, lý thuyết tập thô không yêu cầu bất kỳ thông tin trước về các dữ liệu như khả năng về thống kê và một chức năng thành viên trong lý thuyết tập mờ.

Trong chương này, tôi trình bày một thuật toán phân cụm phân cấp sử dụng xấp xỉ trên dựa trên lý thuyết tập thô. Kết quả phương pháp trả về các cụm thô trong đó một đối tượng là thành viên của nhiều hơn một cụm.[7]

3.2 Bài Toán

Áp dụng thuật toán phân cụm thô vào bài phân cụm người dùng trên web(chuyển hướng người dùng web). Với mỗi người dùng cho ta một đối tượng dữ liệu tuần tự bao gồm tập hợp thứ tự những lần duyệt web của người dùng. Trong luận văn trích trọn n trình tự(n đối tượng người dùng) ngẫu nhiên từ bộ dữ liệu duy nhất được mô tả trong bảng 3.1[7] với n lần lượt : 100, 200, 300, 400, 500, 1000, 2000, 3000, 4000, 5000. Kết quả thực nghiệm được trình bày trong phần **3.6**

Bảng 3.1 Mô tả bảng dữ liệu MSNBC

Tổng số bộ dữ liệu	
Số người sử dụng	989.818
Số lần thăm tối thiểu	1
Số lần thăm tối đa	500
Số lượng thăm trung bình của mỗi người dùng	5,7 lần

Dữ liệu từ kho lưu trữ dữ liệu UCI [http://kdd.ics.uci.edu/] Internet Information Server (IIS) bao gồm các bản ghi cho msnbc.com và tin tức liên quan đến các phần của msn.com. Mỗi chuỗi trong bộ dữ liệu tương ứng với lượt xem của người dùng trong khoảng thời hai mươi bốn giờ. Mỗi sự kiện trong chuỗi tương ứng với yêu cầu của người sử dụng cho một trang. Yêu cầu không được ghi lại ở mức độ tốt nhất của chi tiết nhưng ở cấp độ của loại trang được xác định bởi người quản trị trang web. Đây là 17 trang cụ thể: ‘frontpage’, ‘news’, ‘tech’, ‘local’, ‘opinion’, ‘on-air’, ‘misc’, ‘weather’, ‘health’, ‘living’, ‘business’, ‘sports’, ‘summary’, ‘bbs’ (bulletin board service), ‘travel’, ‘msn-news’ and ‘msn-sports’. Bảng 3.1 cho thấy các đặc tính của dữ liệu. Mỗi loại trang được đại diện bởi một số nguyên nhãn. Ví dụ, ‘frontpage’ được mã hoá là 1, ‘news’ là 2, ‘tech’ như 3, vv Mỗi hàng mô tả các số truy cập của một người dùng duy nhất.

3.3 Dữ liệu tuần tự

Phân nhóm đáng tin cậy của các phiên người dùng web có thể đạt được nếu cả hai nội dung cũng như thứ tự các lượt ghé thăm trang được xem xét. Bằng cách này, cả hai chuyến thăm trang của người sử dụng thực tế cũng như sở thích và yêu cầu người sử dụng được nắm bắt. Hầu hết các phương pháp tiếp cận trong khai thác web không sử dụng tính chất tuần tự của phiên người dùng. Thường được mô hình hóa các phiên trong một chiều không gian vector của các trang web. Các n - không gian vector có thể được nhị phân, cho biết một trang web cụ thể được truy cập hay không trong một phiên. Các vector có thể mang theo các thông tin liên quan đến việc đếm tần số của lượt ghé thăm trang web trong một phiên. Vì vậy, tùy thuộc vào bản chất của các giá trị liên kết với các không gian n, phân tích hạn chế người dùng đang được thực hiện.

Nói chung, các thuật toán phân nhóm sử dụng một trong hai các hàm khoảng cách hay chức năng tương tự để so sánh cặp trình tự. Nhiều người trong số các số liệu cho các trình tự không hoàn toàn đủ điều kiện như là số liệu do một hoặc nhiều lý do. Trong phần sau, giới thiệu ngắn gọn về độ đo tương tự S^3M [7]. Độ đo này xem xét cả các thiết lập cũng như trình tương tự trên hai chuỗi.

Trong chương này trình bày một kỹ thuật phân nhóm mới cho các trình tự sử dụng khái niệm về hạn chế - tương tự xấp xỉ trên. Ý tưởng chính là tìm một tập hợp các tính năng mà nắm bắt được thông tin tuần tự của các chuỗi dữ liệu cũng như nội dung thông tin. Những bộ tính năng được dự báo vào một không gian xấp xỉ trên. Hạn chế - tương tự kỹ thuật xấp xỉ trên được áp dụng để có được xấp xỉ trên của cụm thô trong đó một yếu tố có thể thuộc về nhiều hơn một cụm.

3.4 Độ đo tương tự của các trình tự (S^3M)[7]

Một chuỗi được tạo thành từ một tập hợp các mục có thể xảy ra trong thời gian hay xảy ra cái khác, đó là, ở vị trí nhưng không nhất thiết phải liên quan với thời gian. Có thể nói rằng một chuỗi là một tập có thứ tự của các tập tin. Thông thường, một chuỗi được ký hiệu là S= (a_1,a_2,...,a_n), với a_1,a_2,...,a_n là những tập hợp mục đặt trong chuỗi S.

Chiều dài của chuỗi được định nghĩa là số lượng các tập mục có trong trình tự, ký hiệu là |S|. Để tìm ra các mẫu trong trình tự, nó là cần thiết để không chỉ nhìn vào các mục có trong trình tự mà còn là thứ tự xuất hiện của chúng. Một biện pháp

mới, được gọi là trình tự và thiết lập độ đo tương tự (S^3M) đã được giới thiệu cho các lĩnh vực an ninh mạng. Độ đo S^3M bao gồm hai phần: Một là định lượng các thành phần của chuỗi (bộ tương tự) và một định lượng tính chất tuần tự. Trình tự giống nhau định lượng số lượng tương tự theo thứ tự xuất hiện của các tập mục trong hai chuỗi. Chiều dài của dãy con chung dài nhất (LLCS) đối với chiều dài của chuỗi dài nhất với quyết định các khía cạnh tương tự trên hai chuỗi. Ví dụ, với hai chuỗi A và B, tương tự được đo như sau:

$$SeqSim(A,B) = \frac{LLCS(A,B)}{\max(|A|,|B|)}$$

Bộ tương tự (độ đo tương tự Jaccard) được định nghĩa là tỷ lệ với số tập mục phổ biến và số lượng các tập mục chung trong hai chuỗi. Như vậy, cho hai chuỗi A và B, tập tương tự được đo như sau:

$$SetSim(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Xem xét hai trình tự A và B, trong đó A = (a, b, c, d) và B = (d, c, b, a). Bây giờ, các bi độ đo tập tương tự cho hai trình tự này là 1, chỉ ra rằng thành phần của chúng là như nhau.

Nhưng chúng ta có thể thấy rằng chúng không ở tất cả các tương tự khi xem xét thứ tự xuất hiện của các tập mục. Khía cạnh này được xác bởi các thành phần trình tự giống nhau. Nơi các thành phần tương tự là 0,25 cho những trình tự này. $LLCS$ theo dõi những vị trí xuất hiện của tập mục trong trình tự. Cho hai trình tự, C = (a, b, c, d) và D = (b, a, k, c, t, p, d), $LLCS(C,D)$ và sau khi chuẩn hóa, các thành phần trình tự tương đồng lượt ra được 0,43. Sự tương tự thiết lập cho hai trình tự này là 0,57. Hai ví dụ trên minh họa cho sự cần thiết phải kết hợp bộ tương đồng và trình tự các thành phần tương đồng vào một chức năng. Do đó, S^3M biện pháp cho hai trình tự A và B được cho bởi:

$$S^3M(A,B) = \text{p}*\frac{LLCS(A,B)}{\max(|A|,|B|)} + \text{q}*\frac{|A \cap B|}{|A \cup B|}$$

Với $p + q = 1$ và $\text{p}, q \geq 0$, p và q xác định trọng lượng tương đối được đưa ra cho trật tự xảy ra (trình tự tương đồng) và nội dung (tập tương tự), tương ứng. Trong ứng dụng thực tế, có thể chỉ định các thông số này. Các LLCS giữa hai chuỗi có thể được tìm thấy bằng cách tiếp cận năng động lập trình. Ở đây, $p + q = 1$ và $p, q \geq 0$. p và q xác định trọng lượng tương đối được đưa ra cho trật tự

xảy ra (tương tự) và nội dung (thiết lập tương tự), tương ứng. Trong ứng dụng thực tế, người sử dụng có thể chỉ định các thông số này[7].

S là một tập hợp các chuỗi hữu hạn được tạo ra từ một tập hợp các biểu tượng, $\sum$. Cho R là tập các số thực sau đó $Sim(s_i, s_j)$: S x $S = R$ được gọi là chỉ số tương đồng giữa các trình tự $s_i, s_j \in S$ nếu có đủ các đặc tính sau đây:

(1) *Non negativity*(không âm): $Sim(s_i, s_j) \geq 0$ với $\forall s_i, s_j \in S$.

(2) *Symmetry*(tính đối xứng): $Sim(s_i, s_j) = Sim(s_j, s_i) \geq 0$ với $\forall s_i, s_j \in S$.

(3) *Normalization*(tiêu chuẩn hóa): $Sim(s_i, s_j) \leq 1$ với $\forall s_i, s_j \in S$.

3.5 Thuật toán phân cụm thô

Trong nhiều ứng dụng khai thác dữ liệu, các thuộc tính lớp của hầu hết các đối tượng là không khác biệt nhưng không rõ ràng. Mơ hồ trong dữ liệu đã thu hút các nhà toán học, triết học, lý luận học và gần đây các nhà khoa học máy tính. Lý thuyết tập thô là một phương pháp để giải quyết sự mơ hồ. Khái niệm cốt lõi của lý thuyết tập thô là mối quan hệ không phân biệt được có các tính chất phản xạ, đối xứng và bắc cầu. Tính không phân biệt được phân vùng không gian vào các lớp tương đương, tạo thành các hạt cơ bản.

Cho $X \subseteq U$ là một mối quan hệ $\tau \subseteq X \; x \; U$ là một mối quan hệ dung sai trên U, nếu

1, τ là phản xạ, có nghĩa là đối với bất kỳ $x \in U, x \; \tau \; x$.

2, τ là đối xứng, nghĩa là cho bất kỳ cặp $x, y \in U, x \; \tau \; y = y \; \tau \; x$.

Định nghĩa của xấp xỉ trên và dưới của một tập bây giờ có thể dễ dàng xây dựng sử dụng các lớp khoan dung. Để làm điều này, chúng ta thay thế các lớp dung sai cho các lớp indiscernibility trong định nghĩa cơ bản của xấp xỉ trên và dưới của bộ này. Như vậy, xấp xỉ dung sai một tập hợp con X được của vũ trụ U được định nghĩa như trong định nghĩa 1 như sau:

Định nghĩa 1[7]:

Cho $X \subset U$ và một mối quan hệ dung sai nhị phân R được xác định trên U. Xấp xỉ dưới của X, ký hiệu $\underline{R}(X)$ và xấp xỉ trên của X, ký hiệu $\overline{R}(X)$ tương ứng được quy định như sau:

$\underline{R}(X) = \{x \in X, \mathrm{R}(x) \subseteq X\}$ và

$\overline{R}(X) = \bigcup_{x \in X} R(x)$

Đề xuất một thuật toán phân sử dụng tập thô cho phân nhóm các giao dịch sử dụng web. Cho $x_i \in U$ là một giao dịch người dùng bao gồm chuỗi các lượt ghé thăm trang web. Đối với phân nhóm các giao dịch sử dụng, ban đầu mỗi giao dịch được thực hiện như là một cụm duy nhất. Để cho các cụm thứ i là $C_i = \{x_i\}$. Rõ ràng, C_i là một tập hợp con của U. Xấp xỉ trên của C_i, ký hiệu là $\overline{R}(X)$, là một tập hợp các giao dịch tương tự như x_i, đó là, một sử dụng truy cập các trang web trong x_i cũng có thể truy cập các trang web khác có mặt trong các giao dịch thuộc $\overline{R}(X)$.

Đối với bất kỳ giá trị ngưỡng không âm $\delta \in (0, 1]$ và đối với bất kỳ hai đối tượng $x, y \in U$, một mối quan hệ nhị phân τ trên U được kí hiệu là $x \tau y$ được xác định bởi $x \tau y$ khi và chỉ khi $Sim(x, y) \geq \delta$. Mối quan R này là một quan hệ dung sai và R có cả phản xạ và đối xứng nhưng có thể không bắc cầu. Xấp xỉ trên $\overline{R}(X)$ đầu tiên có một tập hợp các đối tượng giống nhau nhất x_i. Vì vậy, xấp xỉ trên đầu tiên của một đối tượng x_i có thể được định nghĩa như sau:

Định nghĩa 2 [7]:

Đối với một giá trị ngưỡng không âm cho $\delta \in (0, 1]$ và một bộ $X = \{x_1, x_{2, \ldots,} x_n\}$, $X \subseteq U$ xấp xỉ trên đầu tiên là:

$\overline{R}(\{x_i\}) = \{x_j | Sim(x_i, x_j) \geq \delta\}$

Một số bộ trong tập từ xấp xỉ trên đầu tiên có thể chia sẻ các yếu tố (còn gọi là phần tử ranh giới). Các yếu tố ranh giới có thể hướng đến quá trình phân nhóm. Các yếu tố được chia sẻ, được tạo ra sau khi xấp xỉ trên đầu, có thể là thành viên tiềm năng của các tập mới hình thành trong xấp xỉ trên thứ hai hoặc cao hơn. Điều này có thể được quyết định bằng cách tính toán cường độ của yếu tố chia sẻ cho tất cả các cụm nó thuộc về. Điều này được đo bằng cách sử dụng một tham số được gọi là giống nhau tương đối. Giá trị của thứ hai và sự giống nhau xấp xỉ trên cao được tính toán trong điều kiện tương tự tương đối. Đối với hai bộ giao nhau $X, Y \in U$. Sự giống nhau tương đối của X đối với Y với được cho bởi :

$$RelSim(x_i, x_j) = \frac{|\overline{R}(xi) \cap \overline{R}(xj)|}{|\overline{R}(xi) - \overline{R}(xj)|} \quad \text{Khi } \overline{R}(X) \not\subseteq \overline{R}(Y)$$

Bây giờ chúng ta xác định được đề xuất hạn chế tương tự -xấp xỉ trên trong định nghĩa sau đây:

Định nghĩa 3.[7] Cho $X = \{x_1, x_2, \ldots, x_n\}$, $X \subseteq U$. Cho một giá trị không âm cố định σ ∈ (0, 1], hạn chế tương tự-xấp xỉ trên của x_i được cho bởi:
$\overline{RR}(\{x_i\}) = \{ x_j \in \bigcup_{xl \in \overline{R}(xi)} \overline{R}(xl) | RelSim(x_i, x_j) \geq \sigma \}$ Khi $\overline{R}(x_i) \nsubseteq \overline{R}(x_j)$

Nói cách khác, tất cả các trình tự x_j thuộc sự giống nhau xấp xỉ trên của các yếu tố của $\overline{R}(x_i)$ là tương đối tương tự như x_i bị hạn chế (hoặc sáp nhập) vào sự giống nhau xấp xỉ trên tiếp theo của x_i.

Lặp lại quá trình tính toán hạn chế tương tự-xấp xỉ trên tiếp cho một σ cho đến khi hai hạn chế tương tự-xấp xỉ trên liên tiếp vẫn như cũ. Ở đây, δ là một tham số người dùng định nghĩa được gọi là tương tự tương đối, được sử dụng để hợp nhất hai lần xấp xỉ trên cho sự hình thành của thứ hai và cao hơn xấp xỉ trên. σ là người dùng xác định ngưỡng tham số sử dụng để xác định sự giống nhau giữa hai đối tượng và được sử dụng để tìm xấp xỉ trên đầu tiên. Các hạn chế tương tự-xấp xỉ trên được tính cho tất cả các giao dịch của U. Thuật toán đầy đủ cho các tính toán của tập thô dựa trên phân nhóm được đưa ra trong thuật toán 1.

Không giống như các thuật toán truyền thống khác, trong cách tiếp cận này nhiều hơn hai giao dịch có thể kết hợp để tạo thành một cụm. Ngoài ra, số lượng tính toán xấp xỉ trên cho bộ tương tự như giảm đi số lần lặp lại tăng lên. Vì vậy, các phân nhóm thô đề xuất hội tụ nhanh hơn.

Thuật toán

Phân cụm dựa trên tập thô

Input:

T: Một tập hợp các n trình tự $\in U$

Threshold(ngưỡng) $\delta \in (0, 1]$

Tương tự tương đối $\sigma \in (0, 1]$

Output:

Số cụmC

Begin

Step 1: Xây dựng ma trận tương tự sử dụng độ đo S^3M.

Step 2: Đối với mỗi $x_i \in U$, Tính $S_i = \overline{R}(x_i)$ sử dụng định nghĩa 2 cho cho ngưỡng δ.

Step 3: Cho $US = \cup_i S_i$, $C = \emptyset$

Step 4: Với mọi $S_i \in US$ Tính ràng buộc tương tự-xấp xỉ trên tiếp theo S' sử dụng định nghĩa 3 cho tương đối σ

if $S_i = S_i$'

$C = C \cup S_i$'

$US = US \setminus \{S_i\}$

endif

Step 5: Lặp lại bước 4 đến khi $US = \emptyset$

Step 6: Trả về C

End

- Độ phức tạp thuật toán: $O(N^2 log_2 L) + O(N/|R|) + O(Nlogk)$.

Ví dụ: Ta có 10 người dùng với mỗi người dùng là một trình tự chuyển hướng web được cho trong hình sau:

T1: on-air misc misc misc on-air misc
T2: news sports tech local sports sports
T3: bbs bbs bbs bbs bbs bbs
T4: frontpage frontpage sports news news local
T5: on-air weather weather weather weather sports
T6: on-air on-air on-air on-air tech bbs
T7: frontpage bbs bbs frontpage frontpage news
T8: frontpage frontpage frontpage frontpage frontpage bbs
T9: news news travel opinion opinion msn-news
T10: frontpage business frontpage news news bbs

Hình 3.1 Ví dụ dữ liệu chuyển hướng Web

	$T1$	$T2$	$T3$	$T4$	$T5$	$T6$	$T7$	$T8$	$T9$	$T10$
$T1$	1	0	0	0	0.21	0.29	0	0	0	0
$T2$	0	1	0	0.47	0.17	0.17	0.17	0	0.15	0.15
$T3$	0	0	1	0	0	0.25	0.33	0.33	0	0.21
$T4$	0	0.47	0	1	0.17	0	0.45	0.27	0.24	0.5
$T5$	0.21	0.17	0	0.17	1	0.18	0	0	0	0
$T6$	0.29	0.17	0.25	0	0.18	1	0.18	0.21	0	0.17
$T7$	0	0.17	0.33	0.45	0	0.18	1	0.58	0.17	0.62
$T8$	0	0	0.33	0.27	0	0.21	0.58	1	0	0.5
$T9$	0	0.15	0	0.24	0	0	0.17	0	1	0.24
$T10$	0	0.15	0.21	0.5	0	0.17	0.62	0.5	0.24	1

Hình 3.2 Ma trận tương tự bằng cách sử dụng số liệu đề xuất với p = 0,5

Xét 10 chuỗi dữ liệu như hình.3.1. Bảng tương tự đã được tính toán bằng cách sử dụng ma trận tương tự S^3M với p = 0,5 (Hình 3.2). Sự giống nhau xấp xỉ trên đầu tiên tại ngưỡng giá trị δ = 0.2 được cho bởi $\overline{R}(T_i)$ với i = 1, 2, …,10. như dưới đây:

$$\overline{R}(T1) = \{T1, T5, T6\}$$
$$\overline{R}(T2) = \{T2, T4\}$$
$$\overline{R}(T3) = \{T3, T6, T7, T8, T10\}$$
$$\overline{R}(T4) = \{T2, T4, T7, T8, T9, T10\}$$
$$\overline{R}(T5) = \{T1, T5\}$$
$$\overline{R}(T6) = \{T1, T3, T6, T8\}$$
$$\overline{R}(T7) = \{T3, T4, T7, T8, T10\}$$
$$\overline{R}(T8) = \{T3, T4, T6, T7, T8, T10\}$$
$$\overline{R}(T9) = \{T4, T9, T10\}$$
$$\overline{R}(T10) = \{T3, T4, T7, T8, T9, T10\}$$

Hình 3.3 Kết quả $\overline{R}(T_i)$

Trong bước đầu tiên, sự giống nhau xấp xỉ trên thứ hai của xấp xỉ trên của $T1$ được cho bởi

$\overline{R\overline{R}'}(T_1) = \{T_1, T_3, T_5, T_6, T_8\}$

Bây giờ, hạn chế tương tự-xấp xỉ trên được áp dụng trên $\overline{R\overline{R}'}$sử dụng Định nghĩa 3 với σ = 1. Có thể thấy rằng chỉ có các yếu tố $T1$, $T5$ và $T6$ đủ điều kiện để được trong $\overline{R\overline{R}'}(T1)$.

Ví dụ, hãy xem xét yếu tố $T3$, $\overline{R}(T1) \cap \overline{R}(T3) = \{T6\}$ và $\overline{R}(T1) - \overline{R}(T3) = \{T1,T5\}$. Như vậy, sự giống nhau quan hệ cực giữa $T1$ và $T3$ là:

$RelSim(x_i,x_j) = \frac{|\overline{R}(T1)\cap\overline{R}(T3)|}{|\overline{R}(T1)-\overline{R}(T3)|} = \frac{1}{2} < \sigma$ do đó $T3$ sẽ không sáp nhập vào $\overline{R}(T_1)$

Như vậy, Tập các xấp xỉ hạn chế-tương tự được đưa ra trong hình sau:

$$\overline{RR}(T1) = \{\mathbf{T1}, \mathbf{T5}, \mathbf{T6}\}$$
$$\overline{RR}(T2) = \{\mathbf{T2}, \mathbf{T4}\}$$
$$\overline{RR}(T3) = \{\mathbf{T3}, \mathbf{T6}, \mathbf{T7}, \mathbf{T8}, \mathbf{T10}\}$$
$$\overline{RR}(T4) = \{\mathbf{T2}, \mathbf{T4}, \mathbf{T7}, \mathbf{T8}, \mathbf{T9}, \mathbf{T10}\}$$
$$\overline{RR}(T5) = \{\mathbf{T1}, \mathbf{T5}\}$$
$$\overline{RR}(T6) = \{T3, T6, T8\}$$
$$\overline{RR}(T7) = \{\mathbf{T3}, \mathbf{T4}, \mathbf{T7}, \mathbf{T8}, \mathbf{T10}\}$$
$$\overline{RR}(T8) = \{\mathbf{T3}, \mathbf{T4}, \mathbf{T6}, \mathbf{T7}, \mathbf{T8}, \mathbf{T10}\}$$
$$\overline{RR}(T9) = \{\mathbf{T4}, \mathbf{T9}, \mathbf{T10}\}$$
$$\overline{RR}(T10) = \{\mathbf{T3}, \mathbf{T4}, \mathbf{T7}, \mathbf{T8}, \mathbf{T9}, \mathbf{T10}\}$$

Hình 3.4 Tập các xấp xỉ hạn chế-tương tự

Trong các tập trên các tập được in đậm ở trên xấp xỉ liên tiếp đều giống nhau.

Ví dụ: $\overline{R}(T1) = \overline{RR}(T1) = \{T1, T5, T6\}$

Như vậy, sự giống nhau xấp xỉ trên thứ ba sẽ được tính cho chỉ những yếu tố có tương tự liên tiếp xấp xỉ trên là không giống nhau. Như vậy, chỉ T6 cần được xem xét cho sự giống nhau xấp xỉ trên thứ ba.

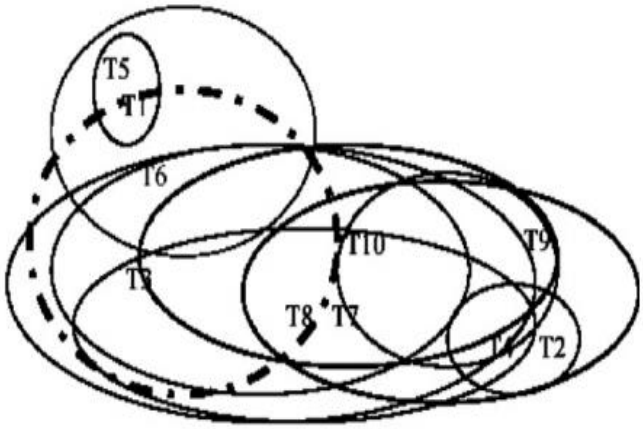
$\overline{RRR}(T6) = \{\boldsymbol{T3}, \boldsymbol{T6}, \boldsymbol{T8}\}$

Do không có sự thay đổi trong hạn chế-tương xấp xỉ trên cho tất cả các yếu tố,thuật toán đã hội tụ. Họ cụm cuối cùng được đưa ra trong hình sau:

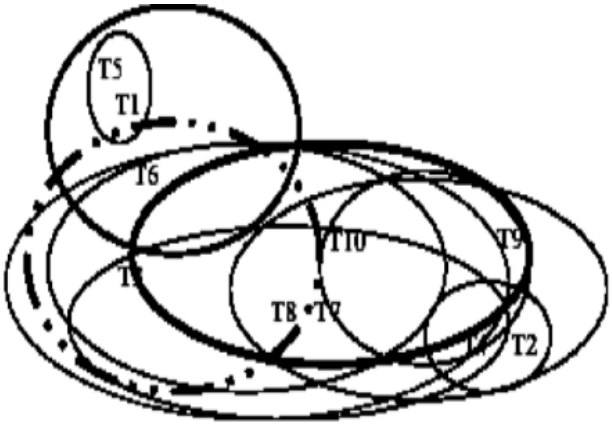
$\{T1, T5, T6\},$
$\{T2, T4\},$
$\{T3, T6, T7, T8, T10\},$
$\{T2, T4, T7, T8, T9, T10\},$
$\{T1, T5\},$
$\{T3, T4, T7, T8, T10\},$
$\{T3, T4, T6, T7, T8, T10\},$
$\{T4, T9, T10\}$ and
$\{T3, T4, T7, T8, T9, T10\}$

Hình 3.5 Họ cụm cuối được đưa ra

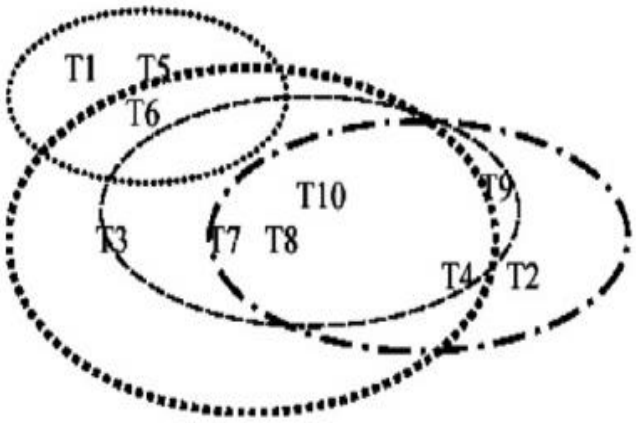
Kết quả thử nghiệm với ví dụ trên:



Hình 3.6 Kết quả xấp xỉ trên đầu tiên



Hình 3.7 Kết quả xấp xỉ trên thứ hai



Hình 3.8 Kết quả xấp xỉ trên thứ ba

3.6 Kết quả thử nghiệm với $\delta = 0.8$ và $\sigma = 1$.

Với trích trọn số lượng mẫu n ngẫu nhiên từ bộ dữ liệu được mô tả trong bảng 3.1 thu được kết quả trong bảng sau.

Bảng 3.2 Kết quả thực nghiệm với $\delta = 0.8$ và $\sigma = 1$.

Số lượng mẫu(n)	Kết quả số cụm trả về sau hạn chế xấp xỉ trên
100	38 Cụm
200	80 Cụm
300	120 Cụm
400	149 Cụm
500	174 Cụm
1000	287 Cụm
2000	467 Cụm
3000	653 Cụm
4000	824 Cụm
5000	965 Cụm

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết Luận

Lý thuyết tập thô, ban đầu được đề xuất bởi Pawlak vào năm 1982, đã thu hút được nhiều nhà nghiên cứu từ các miền khác nhau và dẫn đến các ứng dụng thành công trong các lĩnh vực khác nhau và một trong những bài toán quan trọng trong lĩnh vực khai phá dữ liệu là bài toán phân cụm dữ liệu. Phân cụm dữ liệu, nói một cách khái quát là việc tự động sinh ra các cụm dựa vào sự tương tự của các đối tượng dữ liệu. Trong các kỹ thuật phân cụm dữ liệu, kỹ thuật phân cụm dữ liệu dựa trên lý thuyết tập thô là một lĩnh vực nghiên cứu rộng lớn và đầy triển vọng. Chính vì vậy, với đề tài "**Phân cụm thô của dữ liệu tuần tự**", luận văn đã tập trung tìm hiểu, nghiên cứu và đạt được một số kết quả sau đây:

Tìm hiểu tổng quan về phân cụm dữ liệu, giới thiệu một số khái niệm liên quan trong phân cụm dữ liệu.

Tổng quan về lý thuyết tập thô bao gồm hệ thông tin, bảng quyết định, tính không phân biệt được và xấp xỉ tập hợp.

Dựa trên thuyết tập thô và thuật toán phân cụm thô áp dụng vào bài toán phân cụm người dùng trên web (chuyển hướng người dùng web).

Mặc dù đã cố gắng và nỗ lực hết mình, nhưng do thời gian nghiên cứu và trình độ của bản thân có hạn nên luận văn không thể tránh khỏi những thiếu sót và hạn chế, tôi rất mong nhận được những ý kiến đóng góp để luận văn đạt được kết quả tốt hơn.

Hướng Phát Triển

Trong thời gian tới, tôi sẽ cố gắng tìm hiểu nhiều hơn nữa về các phương pháp phân cụm dữ liệu, đặc biệt là phương pháp phân cụm dựa trên lý thuyết tập thô và cố gắng mở rộng ứng dụng của thuật toán phân cụm thô vào nhiều bài toán thực tế.

Xây dựng và cải tiến thuật toán phân cụm thô áp dụng vào các bài toán với dữ liệu lớn hơn hay mang tính thực tiễn như cảnh báo tắc đường…

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Đỗ Mai Hường (2007),*Một số vấn đề liên quan đến lý thuyết tập thô*. Luận văn thạc sĩ.

[2] Hoàng Văn Dũng (2007),*Khai phá dữ liệu web bằng kỹ thuật phân cụm*. Luận văn thạc sĩ.

[3] Nguyễn Trung Đức (2013), *Tiếp cận mờ trong phân cụm dữ liệu*. Luận văn thạc sĩ.

[4] Phạm Văn Long (2012), *Khai phá dữ liệu theo tiếp cận tập thô và cây quyết định - ứng dụng trong phân lớp năng khiếu học sinh*. Luận văn thạc sĩ.

Tiếng anh

[5] Jianhua Yang (2002), *Algorithmic engineering of clustering and cluster validity with applications to web usage mining*, School of Electrical Engineering and Computer Science, Australia.

[6] Jiawei Han, Micheline Kamber (2001), *Data Mining: Concepts and Techniques - Second Edition*, Hacours Science and Technology Company, USA.

[7] Pradeep Kumar, P. Radha Krishna,, Raju. S. Bapi, Supriya Kumar De(2007): *Rough clustering of sequential data*.

[8] Ivo Düntsch & Günther Gediga (2000), *Rough set data analysis: A road to non-invasive knowledge discovery*.

[9] Zdzislaw Pawlak (1991), *ROUGH SETS Theoretical Aspects of Reasoning about Data*, Institute of Computer Science, Warsaw University of Technology.

Một số trang web

[10] http://documents.tips/documents/ly-thuyet-tap-tho-va-cac-khai-niem.html

[11] http://www.stat.columbia.edu/~madigan/W2025/notes/clustering.pdf.